Available online at www.sciencedirect.com
SCIENCE DIRECT®
Comput. Methods Appl. Mech. Engrg. 195 (2006) 4917–4952
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Conservative single-step time-integration schemes with higher-order accuracy for multi-particle dynamics with local two-point potentials

E. Graham [a], G. Jelenić [b,*]

[a] UGS, Parker's House, 46 Regent Street, Cambridge CB2 1DP, United Kingdom
[b] Department of Civil Engineering, University of Rijeka, Viktora Cara Emina 5, 51000 Rijeka-Fiume, Croatia

Received 4 March 2005; received in revised form 7 July 2005; accepted 8 July 2005

Abstract

A general framework for algorithms that conserve linear and angular momenta for problems of multi-particle mechanics is presented. Conditions for energy conservation are derived, and different manners in which this may be achieved are discussed. A detailed examination of the relative equilibrium states is carried out, and conditions under which algorithms preserve these states are given; in particular, algorithms can be designed to capture the *exact* solutions of relative equilibrium problems, although these algorithms are unlikely to be energy-conserving. Following on from the approach proposed by Argyris et al. [J.H. Argyris, P.C. Dunne, T. Angelopoulos, Dynamic response by large step integration, Earthquake Engrg. Struct. Dynam. 2 (1973) 185–203], the local accuracy characteristics of algorithms are investigated thoroughly, and it is shown that there is no limit to the order of accuracy that can be achieved by algorithms in this framework, even for problems with time-dependent forces. No extra stages of calculation or additional degrees of freedom are required to be present, although the sparsity of the resulting system of equations is compromised. A few examples of new algorithms are given, and their properties verified on some model problems.

Keywords: Higher-order accuracy; Conservation; Relative equilibria; Time reversibility

1. Introduction

In this work, we explore the possibilities for higher-order accuracy in the design of algorithms to solve non-linear, dynamic problems involving large displacements. The exact solutions to these problems are generally unavailable in closed form, existing only as power series in Δt. Our sphere of interest concerns undamped systems for which the forces are derived from a single scalar function, known as *monogenic* systems [2]. We are particularly interested in *Hamiltonian* systems, for which the external forces are *conservative*; in these cases, the Hamiltonian function H defining the total energy of the system is *constant* throughout the motion.

Fundamental to the accuracy of an algorithm is the concept of *stability*. There are many different definitions of algorithmic stability that exist; e.g., [3–6] and many more besides. Loosely speaking, each relates to whether or not an algorithm produces a solution that is bounded as the total time $T \to \infty$ (for some fixed Δt), assuming the exact solution to be bounded also. In any quest for accuracy, stability is essential; therefore we seek to ensure stability first, and subsequently concentrate on improving the accuracy of an algorithm. As a consequence, we will deal exclusively with *implicit* schemes,

* Corresponding author. Tel.: +385 51 352 114; fax: +385 51 332 816.
E-mail address: gordan@gradri.hr (G. Jelenić).

0045-7825/$ - see front matter
doi:10.1016/j.cma.2005.07.025

since explicit algorithms are known to be only conditionally stable [3,7]. For monogenic systems, we can formulate the definition of stability in terms of the change in energy within a single time-step, as was first done by Belytschko and Schoeberle [5]. Specifically, the internal energy of the system must not be allowed to increase beyond the amount of work done by the external forces during the time-step. It is possible to design algorithms that satisfy this condition for all time-step sizes; such algorithms are thus *unconditionally stable* for general non-linear problems.

For Hamiltonian systems, it is also possible to design algorithms that preserve the symplectic nature of the system: a detailed discussion of the construction of symplectic algorithms is given by Simo and Tarnow in Appendix II of [8], and also by Marsden and West [9]. It has been shown by Zhong and Marsden, however, that an algorithm cannot be symplectic and simultaneously conserve energy for a general non-integrable system, assuming the time-step size to be constant [10]. Also, algorithms which are spectrally stable and dissipate energy for linear problems cannot be symplectic, as demonstrated by Simo and Tarnow [8]. A choice must therefore be made, between the properties of symplecticity and energy conservation or dissipation, from the outset. Given the fact that a definition of stability is readily available in terms of energy growth, we elect to design algorithms based on energy criteria. In support of this decision, Simo and co-workers have shown that for stiff problems of non-linear elasticity, energy-conserving algorithms tend to perform better than symplectic algorithms [8,11,12].

Unconditional stability alone is not enough to ensure the overall accuracy of an algorithm. An extreme example of this was given by Ortiz [13], where a convergent, energy-conserving algorithm is shown to give completely inaccurate results after only a small number of time-steps. In the case of *systems with symmetries*, however, we have more information. These systems furnish two other constants of motion, namely the total linear momentum $\mathscr{L}$ and the total angular momentum $\mathscr{J}$, each of which gives information about the qualitative (as well as quantitative) nature of the solution. Algorithms can also be designed to conserve these momenta for such systems, in conjunction with energy conservation, with a view to achieving better accuracy. The importance of angular momentum conservation in regard to accuracy was noted by Betsch and Steinmann [14], and classic examples of such algorithms are given by Simo and co-workers [8,15].

As a consequence of having constant linear and angular momenta, an additional property of systems with symmetries is the existence of families of fully integrable solutions, each induced by a particular combination of initial conditions. These are known as *relative equilibrium states* (or *steady states*), and give further information about the stability of the system: see [16,17] for a detailed account. Algorithms that conserve momenta can be designed to preserve these relative equilibrium states (when the initial conditions arise): those that do give solutions to a steady-state problem that physically resemble the exact solution, and thus may have enhanced stability and accuracy properties for problems of *approximately* steady-state motion. An analysis of two popular time-integration schemes in this regard is given by Gonzalez and Simo [12] and further discussion on the importance of preservation of relative equilibrium orbits is provided by Armero and Romero [18]. Examples of algorithms designed to preserve relative equilibria include the energy–momentum algorithm of Simo and Tarnow [8], and subsequent algorithms that dissipate energy by Armero and co-workers [18–20].

One further property of dynamical systems in physics that we touch upon briefly is that of time-reversibility [21], which is closely related to the uniqueness of a dynamic response. In the discrete case, however, it is not certain that, at any given point on the solution, negating the time-step would recover the solution given at the previous point in time. Algorithms that guarantee this are described as *time-reversible*. An early citation of the importance of this property in the engineering context is due to Argyris et al. [1], and further examples of such algorithms relevant to our work are the energy-conserving algorithm proposed independently by Simo and Gonzalez [11] and Reich [22], as well as the symplectic mid-point rule [8,15,23,24] and the so-called assumed distance method [25].

It is widely accepted that an algorithm should be at least second-order accurate (e.g. [4]), and the energy–momentum algorithms of Simo et al. mentioned earlier all satisfy this requirement. Various ways to increase the order of accuracy have been proposed. These include *composition methods*, whereby greater accuracy is achieved by computing intermediate results at additional points within a single time-step; example algorithms include those given independently by Yoshida [21], Forest [26] and Tarnow and Simo [27]. A disadvantage is that the procedure involves stepping backwards in time, using a larger time-step size than the original algorithm; this makes the principle less attractive for algorithms that are not time-reversible, and increases the risk of instability or divergence during the non-linear iteration process. Another approach to enhancing accuracy can be taken by discretising the equations of motion using *finite elements in time*, where the accuracy can be prescribed by the degree of the polynomial basis functions chosen; example algorithms include those of Betsch and Steinmann [28] for non-linear dynamics. These schemes bear close resemblance to Gauss Runge–Kutta methods, as described in the Appendix of [28].

Both of these strategies to improve accuracy entail additional computation cost, due to the calculation of intermediate results or the presence of extra degrees of freedom in the temporal domain of the problem. A third approach, aimed at avoiding this additional cost, is based on *Taylor series expansions* of the state variables; for linear dynamics, this method is equivalent to using Padé approximations to the exact solution. Early work was done along these lines for non-linear analysis by Argyris et al. [1,29], who presented *arbitrarily* accurate algorithms that are time-reversible, although not conservative. They were followed by LaBudde and Greenspan, who produced arbitrarily accurate schemes that also conserve

energy and angular momentum for a central-force problem [30], and similar energy-conserving schemes for the N-body problem [31]. These algorithms are not time-reversible, however, and do not preserve the orbits of relative equilibria when higher than second-order accurate. In both cases, small time-step sizes were necessary to ensure convergence of the non-linear solution procedure.

To achieve our goal of designing algorithms with desirable accuracy characteristics, we will follow this approach of using Taylor series expansions. Firstly, however, we aim to equip the algorithms with the properties of energy and momentum conservation, preservation of relative equilibrium states and time reversibility, when the relevant physical principles apply. Subject to retaining these properties of the system, we seek to maximise the order of accuracy of the algorithms we design, and present criteria by which it may be achieved.

In [32], we developed a framework for designing higher-order accurate, conservative algorithms for the central-force problem that did not entail additional computational effort in the manner described above. This work extends [32] to the realm of multi-body problems, and discusses the extra complexity involved. Specifically, we deal with non-linear elasticity of multi-element truss structures, with the view that any progress made in this area will highlight avenues of research for time integration involving problems of two- and three-dimensional continua. Also, specific to elastic structures are certain types of strain-energy potential, which in principle may be substituted with a different potential and thus make the present framework entirely applicable to the problems of multi-particle dynamics. The background theory is presented in Section 2, and in Section 3 we establish a general framework in which higher-order algorithms with the required properties exist, that does not entail intermediate calculations or additional degrees of freedom. The conservation and accuracy criteria are derived in Sections 4 and 5, and in Section 6 we take a brief look at some new and existing algorithms that fit into the framework, which are tested on a couple of model problems in Section 7.

2. Equations of motion

Let $\boldsymbol{r}(\boldsymbol{X},t),\dot{\boldsymbol{r}}(\boldsymbol{X},t)\in\mathbb{R}^3$ be the position and velocity at time t of a point $x\in\mathscr{B}$, where $\mathscr{B}_0\subset\mathbb{R}^3$ is the initial volume of the continuum; let $\rho(\boldsymbol{X},t)$ represent the current density of the material, with $\rho_0(\boldsymbol{X})\equiv\rho(\boldsymbol{X},0)$ the initial density; and let $\boldsymbol{X}$ be the position vector of x in the reference configuration. Using the conservation of mass, we have the following definitions (e.g. [33]):

$$M := \int_{\mathscr{B}_0}\rho_0\,\mathrm{d}V, \tag{2.1}$$

$$\mathscr{L} := \int_{\mathscr{B}_0}\rho_0\dot{\boldsymbol{r}}\,\mathrm{d}V, \tag{2.2}$$

$$\mathscr{J} := \int_{\mathscr{B}_0}\boldsymbol{r}\times\rho_0\dot{\boldsymbol{r}}\,\mathrm{d}V \quad \text{and} \tag{2.3}$$

$$T := \frac{1}{2}\int_{\mathscr{B}_0}\rho_0\dot{\boldsymbol{r}}\cdot\dot{\boldsymbol{r}}\,\mathrm{d}V, \tag{2.4}$$

for the mass M, linear momentum $\mathscr{L}$, angular momentum $\mathscr{J}$ and the kinetic energy T of the continuum. We also define Φ to be the total potential energy function from which the forces are derived, which is independent of velocities. Furthermore, we will require knowledge of the *centre of mass* of the continuum, which is defined as

$$\boldsymbol{r}^{\mathrm{c}} := \frac{1}{M}\int_{\mathscr{B}_0}\rho_0\boldsymbol{r}\,\mathrm{d}V \tag{2.5}$$

(using the conservation of mass).

Introducing *spatial* discretisation of the position vector, we have

$$\boldsymbol{r}(\boldsymbol{X},t)=\boldsymbol{N}(\boldsymbol{X})\boldsymbol{R}(t), \tag{2.6}$$

where $\boldsymbol{N}(\boldsymbol{X})\in\mathbb{R}^{3\times 3N}$ is a matrix of shape functions and

$$\boldsymbol{R}(t)=\langle\,\boldsymbol{r}^1(t)\quad\cdots\quad\boldsymbol{r}^N(t)\,\rangle\in\mathbb{R}^{3N}$$

a vector of nodal positions, with N the number of spatial nodes used in the discretisation. (Here and throughout the paper we use the notation $\langle\cdot\rangle$ to describe a column vector thereby enabling such expressions to be contained within a line of text.) Thus $\ddot{\boldsymbol{r}}(\boldsymbol{X},t)=\boldsymbol{N}(\boldsymbol{X})\ddot{\boldsymbol{R}}(t)$ and, by defining the mass matrix $\boldsymbol{M}\in\mathbb{R}^{3N\times 3N}$ to be

$$\boldsymbol{M} := \int_{\mathscr{B}_0}\rho_0\boldsymbol{N}^{\mathrm{T}}\boldsymbol{N}\,\mathrm{d}V, \tag{2.7}$$

we can express the general semi-discrete equation for undamped motion of a continuum as

$$\boldsymbol{M}\ddot{\boldsymbol{R}}+\nabla_{\boldsymbol{R}}\Phi=\boldsymbol{0}. \tag{2.8}$$

For the problems we consider in this work, the potential function $\Phi(\boldsymbol{R}, t)$ can be written as

$$\Phi(\boldsymbol{R}, t) = \phi(\boldsymbol{R}) - U(\boldsymbol{R}, t), \tag{2.9}$$

where ϕ denotes the strain energy of the system, and U is the function from which the externally applied force is derived. Thus we have

$$\nabla_{\boldsymbol{R}} U(\boldsymbol{R}, t) = \boldsymbol{F}(\boldsymbol{R}, t)$$

for an external force $\boldsymbol{F}$, and hence we may express (2.8) as

$$\boldsymbol{M}\ddot{\boldsymbol{R}} + \nabla\phi = \boldsymbol{F}, \tag{2.10}$$

which describes the dynamic equilibrium of the continuum.

For a single linear (two-noded) bar element of initial length l_0 and cross-sectional area A_0 (assumed constant along the length of the bar), we have the position vector $\boldsymbol{R} = \langle \boldsymbol{r}^1 \ \boldsymbol{r}^2 \rangle \in \mathbb{R}^6$ and shape function matrix

$$\boldsymbol{N}(s) = [\, N_1(s)\boldsymbol{I}_3 \quad N_2(s)\boldsymbol{I}_3 \,] \in \mathbb{R}^{3\times 6}; \quad N_1(s) = \frac{l_0 - s}{l_0}, \quad N_2(s) = \frac{s}{l_0} \tag{2.11}$$

for $s \in [0, l_0]$, with $\boldsymbol{I}_3$ the three-dimensional identity matrix. We also have the mass and the stiffness of the bar as

$$m := A_0\rho_0 \int_0^{l_0} \mathrm{d}s \equiv A_0\rho_0 l_0 \quad \text{and} \quad k := \frac{E\overline{A}}{\overline{l}}$$

for constant density ρ_0, where $\overline{l}$ and $\overline{A}$ represent the natural (undeformed) length and cross-sectional area, respectively, and E is Young's modulus of the material. For an elastic, homogeneous bar with assumed linear interpolation given by (2.6) and (2.11), the strain energy function $\phi(\cdot)$ is dependent only on the length $l := \|\boldsymbol{r}^2 - \boldsymbol{r}^1\|$ of the bar, as it is in the equivalent central-force problem [32]. Thus we have

$$\nabla\phi(l) = \phi'(l)\nabla\left\{\sqrt{(\boldsymbol{r}^2 - \boldsymbol{r}^1)\cdot(\boldsymbol{r}^2 - \boldsymbol{r}^1)}\right\} = \frac{\phi'(l)}{l}\widetilde{\boldsymbol{I}}\boldsymbol{R}, \quad \text{where } \widetilde{\boldsymbol{I}} := \begin{pmatrix} \boldsymbol{I}_3 & -\boldsymbol{I}_3 \\ -\boldsymbol{I}_3 & \boldsymbol{I}_3 \end{pmatrix}.$$

Using the abbreviation

$$f(l) := \frac{\phi'(l)}{l}$$

the equation of motion becomes, for a single bar element,

$$\boldsymbol{M}\ddot{\boldsymbol{R}} + f(l)\widetilde{\boldsymbol{I}}\boldsymbol{R} = \boldsymbol{F}. \tag{2.12}$$

In general, we consider systems comprising many bar elements. Applying Hamilton's principle to the whole structure, the equilibrium equation (2.10) becomes an assembly of elemental force contributions in the form of (2.12) (e.g. [4], Section 7.2). Therefore we have

$$\boldsymbol{M}\ddot{\boldsymbol{R}} := \sum_{\substack{i=1,\\ j>i}}^{N} \boldsymbol{M}_{ij}\ddot{\boldsymbol{R}}_{ij}, \quad \nabla\phi := \sum_{\substack{i=1,\\ j>i}}^{N} \nabla\phi_{ij} = \sum_{\substack{i=1,\\ j>i}}^{N} f_{ij}(l_{ij})\widetilde{\boldsymbol{I}}\boldsymbol{R}_{ij} \quad \text{and} \quad \boldsymbol{F} := \sum_{\substack{i=1,\\ j>i}}^{N} \boldsymbol{F}_{ij}$$

for $\boldsymbol{M} \in \mathbb{R}^{3N\times 3N}$; $\boldsymbol{R}, \ddot{\boldsymbol{R}}, \boldsymbol{F}, \nabla\phi \in \mathbb{R}^{3N}$; $\boldsymbol{M}_{ij} \in \mathbb{R}^{6\times 6}$; and $\boldsymbol{R}_{ij}, \ddot{\boldsymbol{R}}_{ij}, \boldsymbol{F}_{ij}, \nabla\phi_{ij} \in \mathbb{R}^6$, where $(\cdot)_{ij}$ denotes a quantity pertaining to the element connecting nodes i and j, with

$$\boldsymbol{R}_{ij} := \langle\, \boldsymbol{r}^i \quad \boldsymbol{r}^j \,\rangle. \tag{2.13}$$

(Note that the summation symbol $\sum$ here conveys an assembly of elemental contributions.) Thus

$$m_{ij} := \begin{cases} A_{ij,0}\rho_{ij,0}l_{ij,0} & : \ \exists \text{ an element connecting nodes } i \text{ and } j, \text{ and} \\ 0 & : \ \text{otherwise} \end{cases}$$

denotes an elemental mass; similarly, k_{ij} and ϕ_{ij} denote the stiffness and strain energy of the element connecting nodes i and j if it exists, respectively, with $k_{ij} := 0$ and $\phi_{ij} := 0$ otherwise. We also have the total mass of the whole structure as

$$M = \sum_{\substack{i=1,\\ j>i}}^{N} m_{ij}. \tag{2.14}$$

In addition, we have the centre of mass for the whole system as

$$\boldsymbol{r}^{\mathrm{c}} = \frac{1}{M}\sum_{\substack{i=1,\\ j>i}}^{N} \frac{m_{ij}}{2}(\boldsymbol{r}^i + \boldsymbol{r}^j) = \frac{1}{M}\boldsymbol{\mathcal{I}}_N \boldsymbol{M}\boldsymbol{R}, \tag{2.15}$$

where the matrix $\boldsymbol{\mathcal{I}}_N \in \mathbb{R}^{3\times 3N}$ is defined as

$$\boldsymbol{\mathcal{I}}_N := [\underbrace{\boldsymbol{I}_3 \quad \cdots \quad \boldsymbol{I}_3}_{N\,\text{times}}]. \tag{2.16}$$

We define the symmetric global matrix $\boldsymbol{\mathcal{F}}$ with submatrices $\boldsymbol{\mathcal{F}}^{ij} \in \mathbb{R}^{3\times 3}$ such that

$$\boldsymbol{\mathcal{F}}^{ij} := \begin{cases} \left(\sum_{k=1}^{N} f_{ik}(l_{ik})\right)\boldsymbol{I}_3 & : \quad i = j, \\ -f_{ij}(l_{ij})\boldsymbol{I}_3 & : \quad i \neq j, \end{cases} \tag{2.17}$$

where $f_{ij} = \frac{\phi^{ij'}(l_{ij})}{l_{ij}}$. (Recall the notational difference between an *elemental matrix* $\boldsymbol{M}_{ij} \in \mathbb{R}^{6\times 6}$ and a *submatrix* $\boldsymbol{M}^{ij} \in \mathbb{R}^{3\times 3}$.) Thus we have

$$\nabla\phi = \boldsymbol{\mathcal{F}}\boldsymbol{R}, \tag{2.18}$$

and so we can now express (2.10) globally as

$$\boldsymbol{M}\ddot{\boldsymbol{R}} + \boldsymbol{\mathcal{F}}\boldsymbol{R} = \boldsymbol{F}, \tag{2.19}$$

where the global mass matrix is symmetric and positive-definite. For linear systems, $\boldsymbol{\mathcal{F}}$ is constant, and we have $\boldsymbol{\mathcal{F}}\boldsymbol{R} = \boldsymbol{K}\boldsymbol{U}$, where $\boldsymbol{K}$ is the *stiffness matrix* and $\boldsymbol{U}$ is a vector of nodal displacements. Here, we emphasise that $\boldsymbol{\mathcal{F}}$ is a function of $\boldsymbol{R}$. In view of our forthcoming time-integration schemes, we split (2.19) into a coupled first-order system with momenta

$$\boldsymbol{P} = \boldsymbol{M}\dot{\boldsymbol{R}} = \langle\, \boldsymbol{p}^1 \quad \cdots \quad \boldsymbol{p}^N \,\rangle \tag{2.20}$$

and positions $\boldsymbol{R}$ as the primary variables. Thus we arrive at

$$\begin{aligned} &\dot{\boldsymbol{P}} + \boldsymbol{\mathcal{F}}\boldsymbol{R} = \boldsymbol{F}, \\ &\dot{\boldsymbol{R}} = \boldsymbol{M}^{-1}\boldsymbol{P} \end{aligned} \tag{2.21}$$

which gives the equations of motion for an assembly of bar elements.

Turning our attention to the conservation properties of the semi-discrete system, for the linear momentum we have, from (2.2), (2.5) and (2.15),

$$\mathscr{L} = M\dot{\boldsymbol{r}}^{\mathrm{c}} = \sum_{\substack{i=1,\\ j>i}}^{N} \frac{m_{ij}}{2}(\dot{\boldsymbol{r}}^i + \dot{\boldsymbol{r}}^j) = \sum_{i=1}^{N} \boldsymbol{p}^i = \boldsymbol{\mathcal{I}}_N \boldsymbol{P} \tag{2.22}$$

and from (2.17) we see that $\sum_{i=1}^{N} \boldsymbol{\mathcal{F}}^{ij} = \boldsymbol{0}_3 \ \forall 1 \leqslant j \leqslant N$, where $\boldsymbol{0}_3$ is the three-dimensional zero matrix. Hence

$$\dot{\mathscr{L}} = \sum_{i=1}^{N} \boldsymbol{F}^i \tag{2.23}$$

and *for systems where the external forces and reactions (due to supports) sum to zero, the total linear momentum is conserved.* With regard to the angular momentum, from (2.3), (2.6) and (2.11), it can be shown that

$$\mathscr{J} = \sum_{i=1}^{N} \boldsymbol{r}^i \times \boldsymbol{p}^i \tag{2.24}$$

and hence

$$\dot{\mathscr{J}} = \sum_{i=1}^{N} (\dot{\boldsymbol{r}}^i \times \boldsymbol{p}^i + \boldsymbol{r}^i \times \dot{\boldsymbol{p}}^i) = \sum_{i=1}^{N} \boldsymbol{r}^i \times \boldsymbol{F}^i, \tag{2.25}$$

since both $\boldsymbol{M}$ and $\boldsymbol{\mathcal{F}}$ are symmetric. Hence *for systems where the moments of the external forces and reactions sum to zero, the total angular momentum is conserved.*

The total energy H of the (undamped) system is the sum of the kinetic and potential energies, i.e.

$$H := T + \Phi. \tag{2.26}$$

We can write the kinetic energy T as

$$T = \frac{1}{2}\boldsymbol{P}\cdot\boldsymbol{M}^{-1}\boldsymbol{P} \tag{2.27}$$

and from (2.9) we have

$$\Phi(\boldsymbol{R},t) = \phi(\boldsymbol{R}) - U(\boldsymbol{R},t).$$

The potential function U can be split into conservative and non-conservative components, i.e.

$$U(\boldsymbol{R},t) = U^{\mathrm{C}}(\boldsymbol{R}) + U^{\mathrm{NC}}(\boldsymbol{R},t). \tag{2.28}$$

Differentiating H with respect to time then gives, from (2.8) and (2.9),

$$\dot{H} = -\frac{\partial U^{\mathrm{NC}}}{\partial t}. \tag{2.29}$$

Thus *for (monogenic) systems where U is not explicitly dependent on time, the total energy is conserved.*

The conservation of momenta gives rise to a set of *relative equilibrium states* as solutions to system (2.21) when $\boldsymbol{F} = \boldsymbol{0}$. These are described in detail in [12,16–18] and references therein; here, we describe those characteristics of relative equilibria that are relevant to truss structures. We also derive the initial conditions under which these states exist. Corresponding results for 3D continua are given in Chapter 3 of [17].

For each type of momentum conserved by the system, there exists an associated *group motion*: thus from linear momentum conservation we have the group of translations, and from angular momentum conservation the group of rotations. Physically speaking, relative equilibria are solutions to system (2.21) when $\boldsymbol{F} = \boldsymbol{0}$ that are *group orbits*; that is, solutions that differ only by a group motion. Therefore relative equilibrium motion can consist of either uniform translation, uniform rotation or, more generally, a combination of the two. This motion is a consequence of the invariance of the states of stresses and strains with respect to rigid body motion.

Mathematically speaking, relative equilibria are solutions to system (2.21) (with $\boldsymbol{F} = \boldsymbol{0}$) that make the total energy H stationary for prescribed values of the linear momentum $\mathscr{L}$ and angular momentum $\mathscr{J}$ [17]. In other words, they are solutions $\langle\widetilde{\boldsymbol{P}}(t)\widetilde{\boldsymbol{R}}(t)\rangle$ such that

$$\nabla H(\widetilde{\boldsymbol{P}},\widetilde{\boldsymbol{R}}) = \boldsymbol{0} \quad \text{with constraints} \quad \mathscr{L}(\widetilde{\boldsymbol{P}}) = \widetilde{\mathscr{L}} \quad \text{and} \quad \mathscr{J}(\widetilde{\boldsymbol{P}},\widetilde{\boldsymbol{R}}) = \widetilde{\mathscr{J}}. \tag{2.30}$$

Accordingly, we need to solve

$$\begin{aligned} &\nabla_{\boldsymbol{R}} H - (\nabla_{\boldsymbol{R}} \otimes \mathscr{J})\boldsymbol{\omega} = \boldsymbol{0},\\ &\nabla_{\boldsymbol{P}} H - (\nabla_{\boldsymbol{P}} \otimes \mathscr{L})\boldsymbol{\eta} - (\nabla_{\boldsymbol{P}} \otimes \mathscr{J})\boldsymbol{\omega} = \boldsymbol{0}, \end{aligned} \tag{2.31}$$

where $\boldsymbol{\eta},\boldsymbol{\omega} \in \mathbb{R}^3$ are the Lagrange multipliers (e.g. [34, Section 4.9]), as shown in [17]. We define the skew-symmetric operator $\widehat{(\cdot)} : \mathbb{R}^3 \to \mathbb{R}^{3\times 3}$ such that

$$\widehat{\boldsymbol{u}} := \begin{pmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{pmatrix} \iff \widehat{\boldsymbol{u}}\boldsymbol{v} \equiv \boldsymbol{u}\times\boldsymbol{v}\ \forall \boldsymbol{u},\boldsymbol{v} \in \mathbb{R}^3, \tag{2.32}$$

and also the global matrix

$$\widehat{\boldsymbol{\Omega}} := \begin{pmatrix} \widehat{\boldsymbol{\omega}} & \boldsymbol{0}_3 & \cdots & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \widehat{\boldsymbol{\omega}} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \cdots & \boldsymbol{0}_3 & \widehat{\boldsymbol{\omega}} \end{pmatrix} \in \mathbb{R}^{3N\times 3N}. \tag{2.33}$$

From (2.22) and (2.24), (2.31) becomes, after some algebra,

$$\begin{aligned} &\mathscr{F}\boldsymbol{R} + \widehat{\boldsymbol{\Omega}}\boldsymbol{P} = \boldsymbol{0},\\ &\boldsymbol{M}^{-1}\boldsymbol{P} - \mathscr{J}_N^{\mathrm{T}}\boldsymbol{\eta} - \widehat{\boldsymbol{\Omega}}\boldsymbol{R} = \boldsymbol{0}. \end{aligned} \tag{2.34}$$

System (2.34), along with the constraints $\mathscr{L}(\boldsymbol{P}) = \widetilde{\mathscr{L}}$ and $\mathscr{J}(\boldsymbol{P},\boldsymbol{R}) = \widetilde{\mathscr{J}}$, can now be solved for $\boldsymbol{R}(t)$, $\boldsymbol{P}(t)$, $\boldsymbol{\eta}(t)$ and $\boldsymbol{\omega}(t)$, thus furnishing the relative equilibrium solution $\langle\widetilde{\boldsymbol{P}}(t),\widetilde{\boldsymbol{R}}(t)\rangle$.

More information concerning the physical nature of relative equilibrium solutions is yet available, however, when we consider that $\boldsymbol{R}(t)$ and $\boldsymbol{P}(t)$ also satisfy (2.21) for $\boldsymbol{F} = \boldsymbol{0}$. Summarising the relevant findings of [17], we have the following important facts regarding the Lagrange multipliers $\boldsymbol{\eta}(t)$ and $\boldsymbol{\omega}(t)$:

$$\begin{aligned} &\text{(i)}\ \ \boldsymbol{\omega} \times \dot{\boldsymbol{r}}^{\mathrm{c}} = \boldsymbol{0}, \\ &\text{(ii)}\ \ \boldsymbol{\eta} = \dot{\boldsymbol{r}}^{\mathrm{c}} - \boldsymbol{\omega} \times \boldsymbol{r}^{\mathrm{c}}, \ \text{and} \\ &\text{(iii)}\ \ \boldsymbol{\omega} = \boldsymbol{\omega}_0 \ \forall t \ \text{i.e. } \boldsymbol{\omega} \text{ is constant.} \end{aligned} \tag{2.35}$$

From (2.2), (2.5), (2.23) and (2.35) we see that $\boldsymbol{\eta}$ is also constant. We now define the *relative position vectors*

$$\bar{\boldsymbol{r}}^i := \boldsymbol{r}^i - \boldsymbol{r}^{\mathrm{c}}; \ 1 \leqslant i \leqslant N \quad \text{and} \quad \overline{\boldsymbol{R}} := \langle\, \bar{\boldsymbol{r}}^1 \quad \cdots \quad \bar{\boldsymbol{r}}^N \,\rangle.$$

By substituting $(2.35)_{2,3}$ into $(2.34)_2$, we get

$$\dot{\overline{\boldsymbol{R}}} - \widehat{\boldsymbol{\Omega}}_0 \overline{\boldsymbol{R}} = \boldsymbol{0}.$$

Since $\sum_{j=1}^{N} \mathscr{F}^{ij} = \boldsymbol{0}_3 \ \forall 1 \leqslant i \leqslant N$, we have $\mathscr{F}\boldsymbol{R} = \mathscr{F}\overline{\boldsymbol{R}}$; furthermore, it can be shown that $\widehat{\boldsymbol{\Omega}}_0$ commutes with $\boldsymbol{M}$, and so by virtue of $(2.35)_{1,3}$ we can write $\widehat{\boldsymbol{\Omega}}\boldsymbol{P} = \widehat{\boldsymbol{\Omega}}_0 \boldsymbol{M} \dot{\overline{\boldsymbol{R}}}$. Therefore (2.34) becomes

$$\begin{aligned} &\mathscr{F}\overline{\boldsymbol{R}} + \widehat{\boldsymbol{\Omega}}_0 \boldsymbol{M} \dot{\overline{\boldsymbol{R}}} = \boldsymbol{0}, \\ &\dot{\overline{\boldsymbol{R}}} - \widehat{\boldsymbol{\Omega}}_0 \overline{\boldsymbol{R}} = \boldsymbol{0}. \end{aligned} \tag{2.36}$$

System (2.36) thus gives necessary and sufficient conditions for the relative equilibrium solution $\boldsymbol{P}(t) = \widetilde{\boldsymbol{P}}(t)$ and $\boldsymbol{R}(t) = \widetilde{\boldsymbol{R}}(t)$ of (2.21) to exist, in terms of the constant vector $\boldsymbol{\omega}$ and the position and velocity of the centre of mass of the structure (which are related). Thus the initial conditions for relative equilibrium states are given by

$$\begin{aligned} &\mathscr{F}_0 \overline{\boldsymbol{R}}_0 + \widehat{\boldsymbol{\Omega}}_0 \boldsymbol{M} \overline{\boldsymbol{V}}_0 = \boldsymbol{0}, \\ &\overline{\boldsymbol{V}}_0 = \widehat{\boldsymbol{\Omega}}_0 \overline{\boldsymbol{R}}_0, \end{aligned} \tag{2.37}$$

where $\dot{\boldsymbol{R}}(0) = \boldsymbol{V}_0$, $\dot{\boldsymbol{r}}^{\mathrm{c}}(0) = \boldsymbol{v}_0^{\mathrm{c}}$, $\overline{\boldsymbol{V}}_0 := \langle\, \boldsymbol{v}_0^1 - \boldsymbol{v}_0^{\mathrm{c}} \quad \cdots \quad \boldsymbol{v}_0^N - \boldsymbol{v}_0^{\mathrm{c}} \,\rangle$ and $\mathscr{F}_0 \equiv \mathscr{F}(\boldsymbol{R}_0)$. Finally, we can obtain the explicit form of $\widetilde{\boldsymbol{P}}(t)$ and $\widetilde{\boldsymbol{R}}(t)$ from $(2.36)_2$. Introducing the matrix exponential $\exp(t\boldsymbol{A})$ defined by

$$\exp(t\boldsymbol{A}) := \sum_{s=0}^{\infty} \frac{t^s}{s!} \boldsymbol{A}^s, \tag{2.38}$$

which is convergent for all $\boldsymbol{A} \in \mathbb{R}^{3N \times 3N}$ (e.g. [35, Chapter 5]), we have

$$\begin{aligned} &\dot{\overline{\boldsymbol{R}}} = \widehat{\boldsymbol{\Omega}}_0 \overline{\boldsymbol{R}} \iff \overline{\boldsymbol{R}}(t) = \exp(t\widehat{\boldsymbol{\Omega}}_0)\overline{\boldsymbol{R}}_0 \iff \bar{\boldsymbol{r}}^i = \exp(t\widehat{\boldsymbol{\omega}_0})\bar{\boldsymbol{r}}_0^i, \\ &\text{and} \quad \dot{\overline{\boldsymbol{R}}}(t) = \exp(t\widehat{\boldsymbol{\Omega}}_0)\overline{\boldsymbol{V}}_0 \iff \dot{\bar{\boldsymbol{r}}}^i = \exp(t\widehat{\boldsymbol{\omega}_0})\bar{\boldsymbol{v}}_0^i; 1 \leqslant i \leqslant N \\ &\text{with} \quad \boldsymbol{r}^{\mathrm{c}}(t) = \boldsymbol{r}_0^{\mathrm{c}} + t\boldsymbol{v}_0^{\mathrm{c}}. \end{aligned} \tag{2.39}$$

It can be shown (e.g. [36]) that the transformation matrix $\exp(t\widehat{\boldsymbol{\omega}_0})$ represents a rotation of angle $\|\boldsymbol{\omega}_0\|t$ about an axis parallel to $\boldsymbol{\omega}_0$. Thus all of the nodes remain fixed in relation to one another, so that *the structure rotates as a rigid body, with (constant) angular velocity* $\boldsymbol{\omega}_0$. Recalling (2.35), we also note that *the velocity of the centre of mass is aligned with the axis of rotation*. Therefore the overall motion is *a superposition of uniform rotation and uniform translation along the axis of rotation*. The two special cases are then pure rotation (when $\boldsymbol{v}_0^{\mathrm{c}} = \boldsymbol{0}$) and pure translation (when $\boldsymbol{\omega}_0 = \boldsymbol{0}$).

3. Algorithm derivation

Following the approach given in [32], we now describe a family of single-step time-integration schemes to solve system (2.21) approximately, that can be specialised to conserve various constants of motion. These algorithms are given in terms of global position and momentum vectors, which implies that these quantities are defined to be continuous across element boundaries. The general form for such a family is

$$\begin{aligned} &\boldsymbol{R}_{n+1} = \boldsymbol{A}\boldsymbol{R}_n + \boldsymbol{B}\boldsymbol{P}_n + \boldsymbol{R}_F, \\ &\boldsymbol{P}_{n+1} = \boldsymbol{C}\boldsymbol{R}_n + \boldsymbol{D}\boldsymbol{P}_n + \boldsymbol{P}_F, \end{aligned} \tag{3.1}$$

where $\boldsymbol{R}_k, \boldsymbol{P}_k \in \mathbb{R}^{3N}$ are the discrete approximations to the positions $\boldsymbol{R}(t_k)$ and momenta $\boldsymbol{P}(t_k)$ at time $t_k \geqslant 0$; $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$, $\boldsymbol{D} \in \mathbb{R}^{3N\times 3N}$ are matrices of parameters that depend upon the configuration and the time-step Δt; and $\boldsymbol{R}_F, \boldsymbol{P}_F \in \mathbb{R}^{3N}$ are vectors that depend on the external force $\boldsymbol{F}$. Given the form of the matrices $\mathscr{F}$ and $\boldsymbol{M}$ in (2.21), we now restrict matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ and $\boldsymbol{D}$ to have scaled unit submatrices also; therefore

$$\boldsymbol{A}^{ij} = a^{ij}\boldsymbol{I}_3, \quad \boldsymbol{B}^{ij} = b^{ij}\boldsymbol{I}_3, \quad \boldsymbol{C}^{ij} = c^{ij}\boldsymbol{I}_3, \quad \text{and} \quad \boldsymbol{D}^{ij} = d^{ij}\boldsymbol{I}_3; \qquad 1 \leqslant i, j \leqslant N.$$

Thus in general there are $4N^2$ unspecified parameters in (3.1), excluding those related to $\boldsymbol{R}_F$ and $\boldsymbol{P}_F$.

We now define

$$\boldsymbol{Z}_n := \left\{ \begin{matrix} \boldsymbol{R}_n \\ \boldsymbol{P}_n \end{matrix} \right\} \quad \text{and} \quad \boldsymbol{Z}_F(\boldsymbol{Z}_{n+1}, \boldsymbol{Z}_n, \boldsymbol{F}, \Delta t) := \left\{ \begin{matrix} \boldsymbol{R}_F \\ \boldsymbol{P}_F \end{matrix} \right\}$$

and can thus express algorithm (3.1) in matrix form as

$$\boldsymbol{Z}_{n+1} = \mathscr{B}_{n+1}\boldsymbol{Z}_n + \boldsymbol{Z}_F, \quad \text{where } \mathscr{B}_{n+1} \equiv \mathscr{B}(\boldsymbol{Z}_{n+1}, \boldsymbol{Z}_n, \Delta t) := \begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{pmatrix} \tag{3.2}$$

(with $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{D}$, $\boldsymbol{P}_F$ and $\boldsymbol{R}_F$ dependent on $\boldsymbol{Z}_{n+1}$, $\boldsymbol{Z}_n$ and Δt, and $\boldsymbol{P}_F$ and $\boldsymbol{R}_F$ also dependent on $\boldsymbol{F}$). We require that $\mathscr{B}_{n+1}$ be non-singular, to prevent the possible occurrence of the solution $\boldsymbol{Z}_{n+1} = \boldsymbol{0}$ when $\boldsymbol{Z}_F = \boldsymbol{0}$ and $\boldsymbol{Z}_n \neq \boldsymbol{0}$. Thus $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ and $\boldsymbol{D}$ are constrained by the condition $\det(\mathscr{B}_{n+1}) \neq 0$. An expression for the determinant of a 2×2 block matrix in the form of $(3.2)_2$ can be given in terms of its component matrices as

$$\det(\mathscr{B}_{n+1}) = \det(\boldsymbol{A})[\det(\boldsymbol{D} - \boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B})] \tag{3.3}$$

provided that the upper left matrix $\boldsymbol{A}$ is non-singular (e.g. Chapter 0 of [35]).

3.1. Inherent momentum conservation

We now look at the conditions under which algorithm (3.1) conserves both linear and angular momenta in the absence of external forces. From (2.22), the total discrete linear momentum $\mathscr{L}_n$ at time-step n is computed as

$$\mathscr{L}_n = \sum_{i=1}^{N} \boldsymbol{p}_n^i = \mathscr{I}_N \boldsymbol{P}_n.$$

From (3.1), we have

$$\mathscr{L}_\Delta = \mathscr{I}_N \boldsymbol{P}_\Delta = \mathscr{I}_N[\boldsymbol{C}\boldsymbol{R}_n + (\boldsymbol{D} - \boldsymbol{I}_{3N})\boldsymbol{P}_n + \boldsymbol{P}_F],$$

where here and throughout the paper we define $(\cdot)_\Delta := (\cdot)_{n+1} - (\cdot)_n$ for any given physical quantity $(\cdot)$, and we note that $\boldsymbol{R}_n$ and $\boldsymbol{P}_n$ are arbitrary. Therefore algorithm (3.1) will conserve linear momentum in general if and only if

$$\mathscr{I}_N \boldsymbol{C} = \mathscr{O}_N, \quad \mathscr{I}_N \boldsymbol{D} = \mathscr{I}_N \quad \text{and} \quad \mathscr{I}_N \boldsymbol{P}_F = \boldsymbol{0}, \tag{3.4}$$

where the matrix $\mathscr{O}_N \in \mathbb{R}^{3\times 3N}$ is defined as

$$\mathscr{O}_N := [\underbrace{\boldsymbol{0}_3 \quad \cdots \quad \boldsymbol{0}_3}_{N \text{ times}}]. \tag{3.5}$$

Thus the columns of $\boldsymbol{C}$ must sum to zero (meaning that $\boldsymbol{C}$ is singular), and those of $\boldsymbol{D}$ to one, whilst the force-related momentum components $\boldsymbol{p}_F^i$ must also sum to zero. This amounts to $2N$ conditions involving $\boldsymbol{C}$ and $\boldsymbol{D}$ that algorithm (3.1) must satisfy.

From (2.24), the total discrete angular momentum $\mathscr{J}_n$ at time-step n is given as

$$\mathscr{J}_n = \sum_{i=1}^{N} \boldsymbol{r}_n^i \times \boldsymbol{p}_n^i.$$

From (3.1), we have

$$\begin{aligned}\mathscr{J}_\Delta &= \sum_{i=1}^{N}\left(\boldsymbol{r}^i_{n+1}\times\boldsymbol{p}^i_{n+1}-\boldsymbol{r}^i_n\times\boldsymbol{p}^i_n\right)=\sum_{i=1}^{N}\left(\left[\sum_{j=1}^{N}\left(a^{ij}\boldsymbol{r}^j_n+b^{ij}\boldsymbol{p}^j_n\right)+\boldsymbol{r}^i_F\right]\times\left[\sum_{k=1}^{N}\left(c^{ik}\boldsymbol{r}^k_n+d^{ik}\boldsymbol{p}^k_n\right)+\boldsymbol{p}^i_F\right]-\boldsymbol{r}^i_n\times\boldsymbol{p}^i_n\right)\\ &=\sum_{i=1}^{N}\left[\sum_{j,k=1}^{N}\left(a^{ij}c^{ik}\boldsymbol{r}^j_n\times\boldsymbol{r}^k_n+b^{ij}d^{ik}\boldsymbol{p}^j_n\times\boldsymbol{p}^k_n+a^{ij}d^{ik}\boldsymbol{r}^j_n\times\boldsymbol{p}^k_n+b^{ij}c^{ik}\boldsymbol{p}^j_n\times\boldsymbol{r}^k_n\right)+\boldsymbol{r}^i_{n+1}\times\boldsymbol{p}^i_F+\boldsymbol{r}^i_F\times\boldsymbol{p}^i_{n+1}-\boldsymbol{r}^i_F\times\boldsymbol{p}^i_F-\boldsymbol{r}^i_n\times\boldsymbol{p}^i_n\right]\\ &=\sum_{i=1}^{N}\left(\sum_{\substack{j=1,\\ k>j}}^{N}\left[(a^{ij}c^{ik}-a^{ik}c^{ij})\boldsymbol{r}^j_n\times\boldsymbol{r}^k_n+(b^{ij}d^{ik}-b^{ik}d^{ij})\boldsymbol{p}^j_n\times\boldsymbol{p}^k_n\right]\right)\\ &\quad+\sum_{i=1}^{N}\left(\sum_{j,k=1}^{N}(a^{ij}d^{ik}-b^{ik}c^{ij})\boldsymbol{r}^j_n\times\boldsymbol{p}^k_n-\boldsymbol{r}^i_n\times\boldsymbol{p}^i_n+\boldsymbol{r}^i_{n+1}\times\boldsymbol{p}^i_F+\boldsymbol{r}^i_F\times\boldsymbol{p}^i_{n+1}-\boldsymbol{r}^i_F\times\boldsymbol{p}^i_F\right),\end{aligned}$$

where the vectors $\boldsymbol{r}^j_n\times\boldsymbol{r}^k_n$, $\boldsymbol{p}^j_n\times\boldsymbol{p}^k_n$ and $\boldsymbol{r}^j_n\times\boldsymbol{p}^k_n$ are unrelated for all $j,k>j$; $1\leqslant j,k\leqslant N$. Therefore algorithm (3.1) will conserve angular momentum in general if and only if

$$\left.\begin{aligned}&\sum_{i=1}^{N}(a^{ij}c^{ik}-a^{ik}c^{ij})=0,\\ &\sum_{i=1}^{N}(b^{ij}d^{ik}-b^{ik}d^{ij})=0\end{aligned}\right\}\ \forall j,k>j;\quad \sum_{i=1}^{N}(a^{ij}d^{ik}-b^{ik}c^{ij})=\delta_{ij}\ \forall j,k$$
$$\text{and}\quad \boldsymbol{r}^i_F=\boldsymbol{p}^i_F=\boldsymbol{0}\ \forall i;\ 1\leqslant i,j,k\leqslant N,$$

where δ_{ij} is the Kronecker delta such that $\delta_{ij}=1$ if $i=j$ and 0 otherwise. This amounts to a further $2N^2-N$ conditions on $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ and $\boldsymbol{D}$ which, when expressed in matrix form, become

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{C}=\boldsymbol{C}^{\mathrm{T}}\boldsymbol{A},\quad \boldsymbol{B}^{\mathrm{T}}\boldsymbol{D}=\boldsymbol{D}^{\mathrm{T}}\boldsymbol{B}\quad\text{and}\quad \boldsymbol{A}^{\mathrm{T}}\boldsymbol{D}-\boldsymbol{C}^{\mathrm{T}}\boldsymbol{B}=\boldsymbol{I}_{3N},\tag{3.6}$$

along with $\boldsymbol{R}_F=\boldsymbol{P}_F=\boldsymbol{0}$. Now, from (3.3) we see that

$$\det(\mathscr{B}_{n+1})=\det(\boldsymbol{A}^{\mathrm{T}})[\det(\boldsymbol{D}-\boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B})]=\det(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{D}-\boldsymbol{A}^{\mathrm{T}}\boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B})$$

and when incorporating conditions (3.6), this becomes

$$\det(\mathscr{B}_{n+1})=\det(\boldsymbol{I}_{3N}+\boldsymbol{C}^{\mathrm{T}}\boldsymbol{B}-\boldsymbol{C}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{A}^{-1}\boldsymbol{B})=\det(\boldsymbol{I}_{3N})=1.\tag{3.7}$$

Hence angular momentum conservation automatically assures that $\mathscr{B}_{n+1}$ is non-singular, as required, with unit determinant. Thus *any family of single-step algorithms that conserves both linear and angular momenta can have at most* $2N^2-N$ *free parameters, excluding those related to the external force*. Note that in linear dynamics, $\mathscr{B}_{n+1}$ is a constant amplification matrix; thus (3.7) implies preservation of the phase space (Liouville's theorem) for any angular momentum-conserving algorithm in linear dynamics.

3.2. Choice of parameters

We wish to express our algorithms in a form that will relate to our previous work on the central-force problem [32]; thus we choose

$$\begin{aligned}&\frac{1}{\Delta t}\left(\boldsymbol{P}_\Delta+\mathscr{G}\boldsymbol{P}_{1/2}\right)=-\mathscr{X}\boldsymbol{R}_{1/2}+\boldsymbol{F}_a,\\ &\frac{1}{\Delta t}\left(\boldsymbol{R}_\Delta-\mathscr{G}^{\mathrm{T}}\boldsymbol{R}_{1/2}\right)=\mathscr{M}^{-1}\boldsymbol{P}_{1/2}-\boldsymbol{V}_a,\end{aligned}\tag{3.8}$$

where $\mathscr{G}$, $\mathscr{X}$ and $\mathscr{M}$ are the parameter matrices (with $\mathscr{M}$ necessarily non-singular), $\boldsymbol{F}_a$ and $\boldsymbol{V}_a$ are algorithmic force and velocity vectors that pertain to the external force; here and throughout the paper we define $(\cdot)_{1/2}:=\frac{1}{2}[(\cdot)_n+(\cdot)_{n+1}]$. As before, each matrix will consist of unit submatrices:

$$\mathscr{G}^{ij}=g^{ij}\boldsymbol{I}_3,\quad \mathscr{X}^{ij}=x^{ij}\boldsymbol{I}_3\quad\text{and}\quad \mathscr{M}^{ij}=\mu^{ij}\boldsymbol{I}_3.\tag{3.9}$$

We will refer to this family of momentum-conserving algorithms collectively as Algorithm MC. Note that for the specific case where $\mathscr{G} := \mathbf{0}_{3N}$, $\mathscr{M} := \boldsymbol{M}$, $\boldsymbol{F}_a := \boldsymbol{F}$ and $\boldsymbol{V}_a := \boldsymbol{0}$, we have

$$\begin{aligned} \frac{1}{\Delta t}\boldsymbol{P}_\Delta &= -\mathscr{X}\boldsymbol{R}_{1/2} + \boldsymbol{F}, \\ \frac{1}{\Delta t}\boldsymbol{R}_\Delta &= \boldsymbol{M}^{-1}\boldsymbol{P}_{1/2}, \end{aligned} \tag{3.10}$$

which is the form of several familiar time-integration schemes (e.g. [8,15,37]), each distinguished by its definition of $\mathscr{X}$.

We can relate this set of parameters to the canonical set $\{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}, \boldsymbol{P}_F, \boldsymbol{R}_F\}$ by forming the equation

$$\boldsymbol{Z}_\Delta = \mathscr{C}_{n+1}\boldsymbol{Z}_{1/2} + \boldsymbol{Z}_a$$

from (3.8), where

$$\mathscr{C}_{n+1} := \begin{pmatrix} \mathscr{G}^{\mathrm{T}} & \Delta t\mathscr{M}^{-1} \\ -\Delta t\mathscr{X} & -\mathscr{G} \end{pmatrix} \quad \text{and} \quad \boldsymbol{Z}_a := \Delta t\begin{Bmatrix} -\boldsymbol{V}_a \\ \boldsymbol{F}_a \end{Bmatrix}. \tag{3.11}$$

This in turn gives us

$$\boldsymbol{Z}_{n+1} = \left(\boldsymbol{I}_{6N} - \frac{1}{2}\mathscr{C}_{n+1}\right)^{-1}\left[\left(\boldsymbol{I}_{6N} + \frac{1}{2}\mathscr{C}_{n+1}\right)\boldsymbol{Z}_n + \boldsymbol{Z}_a\right],$$

and thus from $(3.2)_1$ we have

$$\mathscr{B}_{n+1} = \left[\boldsymbol{I}_{6N} - \frac{1}{2}\mathscr{C}_{n+1}\right]^{-1}\left[\boldsymbol{I}_{6N} + \frac{1}{2}\mathscr{C}_{n+1}\right] \quad \text{and} \quad \boldsymbol{Z}_F = \left[\boldsymbol{I}_{6N} - \frac{1}{2}\mathscr{C}_{n+1}\right]^{-1}\boldsymbol{Z}_a. \tag{3.12}$$

Combining (3.11) and (3.12) leads to $\{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}, \boldsymbol{P}_F, \boldsymbol{R}_F\}$ in terms of $\{\mathscr{G}, \mathscr{X}, \mathscr{M}, \boldsymbol{F}_a, \boldsymbol{V}_a\}$, provided all the relevant inverses exist.

4. Properties of the algorithm

We will now derive conditions for parameters $\mathscr{G}$, $\mathscr{X}$, $\mathscr{M}$, $\boldsymbol{F}_a$ and $\boldsymbol{V}_a$ under which the algorithm will conserve linear and angular momenta, conserve the total energy, preserve relative equilibria and be time-reversible.

4.1. Conservation of linear momentum

Proposition 1. *Algorithm MC gives the discrete linear momentum derivative as*

$$\frac{1}{\Delta t}\mathscr{L}_\Delta = \sum_{i=1}^{N}\boldsymbol{F}_a^i \tag{4.1}$$

provided that

$$\sum_{i=1}^{N}\mathscr{G}^{ij} = \sum_{i=1}^{N}\mathscr{X}^{ij} = \boldsymbol{0}_3 \quad \forall 1 \leqslant j \leqslant N. \tag{4.2}$$

Thus linear momentum is conserved whenever $\sum_{i=1}^{N}\boldsymbol{F}_a^i = \boldsymbol{0}$.

The proof is given in Appendix A.1; note the similarity between (4.1) and (2.23). Eq. (4.2) therefore imposes $2N$ conditions on matrices $\mathscr{G}$ and $\mathscr{X}$.

4.2. Conservation of angular momentum

Proposition 2. *Algorithm MC gives the discrete angular momentum derivative as*

$$\frac{1}{\Delta t}\mathscr{J}_\Delta = \sum_{i=1}^{N}\left(\boldsymbol{r}_{1/2}^i \times \boldsymbol{F}_a^i + \boldsymbol{p}_{1/2}^i \times \boldsymbol{v}_a^i\right) \tag{4.3}$$

provided that

$$\mathscr{X} = \mathscr{X}^{\mathrm{T}} \quad \textit{and} \quad \mathscr{M} = \mathscr{M}^{\mathrm{T}}. \tag{4.4}$$

Thus angular momentum is conserved whenever $\boldsymbol{F}_a = \boldsymbol{V}_a = \boldsymbol{0}$.

The proof is given in Appendix A.2; note the similarity between (4.3) and (2.25). Eq. (4.4) imposes a further $N^2 - N$ conditions on matrices $\mathscr{X}$ and $\mathscr{M}$. The total number of free parameters remaining in $\mathscr{G}$, $\mathscr{X}$ and $\mathscr{M}$ after conservation of momenta has been secured is $2N^2 - N$. This tallies with the amount given in Section 3 for $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ and $\boldsymbol{D}$; hence *Algorithm MC is a fully general form for single-step momentum-conserving algorithms*, provided that (4.2) and (4.4) are satisfied. Note that this means $\mathscr{G}$ and $\mathscr{X}$ are singular, and that $\mathscr{X}$ and $\mathscr{M}$ are symmetric.

4.3. Conservation of energy

From (2.9), (2.26) and (2.27), the discrete total energy at time-step n is given as

$$H_n = \frac{1}{2}\boldsymbol{P}_n \cdot \boldsymbol{M}^{-1}\boldsymbol{P}_n + \phi_n - U_n. \tag{4.5}$$

For Algorithm MC to be energy-conserving, we therefore require

$$H_{n+1} = H_n \tag{4.6}$$

to hold for all n under the appropriate conditions.

4.3.1. A single, global condition for energy conservation

Proposition 3. *Algorithm MC gives the discrete energy derivative as*

$$\frac{1}{\Delta t}H_\Delta = -\frac{U^{\mathrm{NC}}_\Delta}{\Delta t} \tag{4.7}$$

if $\mathscr{X} := \kappa\overline{\mathscr{X}}$ *for an arbitrary matrix* $\overline{\mathscr{X}}$, *where* κ *is defined by*

$$\begin{aligned} &A\kappa + B = 0 \quad \textit{for} \\ &A = -\Delta t\overline{\mathscr{X}}\boldsymbol{R}_{1/2}\cdot\boldsymbol{M}^{-1}\boldsymbol{P}_{1/2} \quad \textit{and} \quad B = \phi_\Delta - U^{\mathrm{C}}_\Delta + (\Delta t\boldsymbol{F}_a - \mathscr{G}\boldsymbol{P}_{1/2})\cdot\boldsymbol{M}^{-1}\boldsymbol{P}_{1/2}, \end{aligned} \tag{4.8}$$

with U^{C} *and* U^{NC} *denoting the conservative and non-conservative parts of the potential function* U *as defined in* (2.28). *Thus energy is conserved whenever* $U^{\mathrm{NC}} = 0$.

The proof is given in Appendix A.3, and we see the similarity between (4.7) and (2.29). Eq. (4.8) therefore fixes one more of the free parameters; thus *families of single-step energy–momentum algorithms can have up to* $2N^2 - N - 1$ *free parameters, excluding those related to the external force*.

Algorithms that do not conserve energy can be converted to energy-conserving algorithms by transforming $\mathscr{X} \mapsto \kappa\mathscr{X}$, with κ as defined in (4.8). This leaves any momentum-conserving attributes of the algorithm intact, since the structure of $\mathscr{X}$ is unaltered. For example, the *symplectic-momentum mid-point algorithm* (e.g. [15,23,24,38]), which is an instance of (3.10) with $\mathscr{X}$ defined such that $\mathscr{X}\boldsymbol{R}_{1/2} := \nabla_{\boldsymbol{R}_{1/2}}\phi(\boldsymbol{R}_{1/2})$, can be made to conserve energy with κ defined by (4.8); this becomes

$$[\nabla_{\boldsymbol{R}_{1/2}}\phi(\boldsymbol{R}_{1/2})\cdot\boldsymbol{R}_\Delta]\kappa = \phi_\Delta \tag{4.9}$$

for constant $\boldsymbol{F}$, which matches Eq. (2.27) of [15], where this algorithm was first proposed. The property of symplecticity is not present in the new algorithm, however. A similar conversion can be performed on another instance of (3.10) known as the *assumed distance method* [37,25], which preserves relative equilibria but does not conserve energy in general; defining κ according to (4.8) generates an energy-conserving algorithm, but the property of preserving relative equilibria is lost, as will be explained in Section 4.5.

There are serious flaws in using a single scalar condition to establish energy conservation, however. Firstly, (4.8) admits a solution for κ only when $A \neq 0$ or $\lim_{A\to 0}\{-B/A\}$ exists. Except for the particular case of a single bar element fixed at one end with no external force (thereby equivalent to a central-force problem), this cannot be guaranteed; hence a drawback with this method of energy conservation is the fact that, for typical $\mathscr{G}$, $\mathscr{M}$ and $\overline{\mathscr{X}}$, it does not guarantee an energy-conserving solution for all possible configurations and time-step sizes, since κ cannot always be defined by (4.8). (This issue was also raised in [15].) Another objection is that it introduces a global coupling between all of the degrees of freedom; this cannot be motivated physically, as mentioned in [22], and it destroys any sparsity in the tangent stiffness matrix used during the non-linear solution procedure. Furthermore, adding energy conservation to a non-conserving algorithm does not necessarily enhance its performance: results have shown that in cases where a non-conserving scheme fails due to a large increase in the total energy, its conserving counterpart invariably fails to provide a solution (i.e., the non-linear solution procedure does not converge) [39,40]. In the case of the assumed distance method, adding energy conservation in this way can actually degrade its performance [40].

In summary, it does not seem as though algorithms should be designed to conserve energy in this way, since there are many undesirable side-effects. An alternative approach to energy conservation [8,11,22,41] will now be presented.

4.3.2. Elemental conditions for energy conservation

From Appendix A.3 we see that

$$H_\Delta = -U^{\mathrm{NC}}_\Delta \iff (\Delta t \boldsymbol{F}_a - \mathscr{G}\boldsymbol{P}_{1/2}) \cdot \boldsymbol{M}^{-1}\boldsymbol{P}_{1/2} + \phi_\Delta - U^{\mathrm{C}}_\Delta = \Delta t \mathscr{X}\boldsymbol{R}_{1/2} \cdot \boldsymbol{M}^{-1}\boldsymbol{P}_{1/2},$$

and thus using $(3.8)_2$ to substitute for $\boldsymbol{P}_{1/2}$ we arrive at the condition

$$\begin{aligned}\mathscr{X}\boldsymbol{R}_{1/2} \cdot \boldsymbol{M}^{-1}\mathscr{M}(\boldsymbol{R}_\Delta - \mathscr{G}^{\mathrm{T}}\boldsymbol{R}_{1/2}) = \phi_\Delta - U^{\mathrm{C}}_\Delta - \Delta t \mathscr{X}\boldsymbol{R}_{1/2} \cdot \boldsymbol{M}^{-1}\mathscr{M}\boldsymbol{V}_a + \left(\Delta t \boldsymbol{F}_a - \mathscr{G}\mathscr{M}\left[\frac{1}{\Delta t}(\boldsymbol{R}_\Delta - \mathscr{G}^{\mathrm{T}}\boldsymbol{R}_{1/2}) + \boldsymbol{V}_a\right]\right) \\ \cdot \boldsymbol{M}^{-1}\mathscr{M}\left[\frac{1}{\Delta t}(\boldsymbol{R}_\Delta - \mathscr{G}^{\mathrm{T}}\boldsymbol{R}_{1/2}) + \boldsymbol{V}_a\right]\end{aligned} \tag{4.10}$$

for energy conservation.

For algorithms of the form (3.10), (4.10) reduces to

$$\mathscr{X}\boldsymbol{R}_{1/2} \cdot \boldsymbol{R}_\Delta = \phi_\Delta - U^{\mathrm{C}}_\Delta + \boldsymbol{F} \cdot \boldsymbol{R}_\Delta = \phi_\Delta, \tag{4.11}$$

since $\boldsymbol{F} \cdot \boldsymbol{R}_\Delta = U^C_\Delta$. Given that these algorithms conserve both momenta, $\mathscr{X}$ must be symmetric with rows and columns that sum to zero. Here and throughout the paper we will use the shorthand notation $\langle \boldsymbol{B}_{ij} \rangle$ to denote symmetric block matrices $\boldsymbol{A} \in \mathbb{R}^{nm \times nm}$ of the form

$$\boldsymbol{A}^{ij} = \begin{cases} \sum_{k=1}^{N} \boldsymbol{B}_{ik} & : \quad i = j, \\ -\boldsymbol{B}_{ij} & : \quad i \neq j, \end{cases}$$

for some $\boldsymbol{B}_{ij} \in \mathbb{R}^{m \times m}$; note that each row (and column) sums to zero. Hence we can write

$$\mathscr{X} \equiv \langle \xi_{ij} \boldsymbol{I}_3 \rangle \tag{4.12}$$

for some ξ_{ij} (to be defined). It can easily be shown that, as a consequence of the form of $\mathscr{X}$,

$$\mathscr{X}\boldsymbol{R}_{1/2} \cdot \boldsymbol{R}_\Delta = \sum_{\substack{i=1,\\ j>i}}^{N} \xi_{ij} \boldsymbol{r}^{ij}_{1/2} \cdot \boldsymbol{r}^{ij}_\Delta \quad \forall \boldsymbol{R}_n, \boldsymbol{R}_{n+1} \text{ and } \xi_{ij},$$

where $\boldsymbol{r}^{ij} := \boldsymbol{r}^j - \boldsymbol{r}^i$. Recalling the definition of ϕ_Δ, (4.11) becomes

$$\sum_{\substack{i=1,\\ j>i}}^{N} \xi_{ij} \boldsymbol{r}^{ij}_{1/2} \cdot \boldsymbol{r}^{ij}_\Delta = \sum_{\substack{i=1,\\ j>i}}^{N} \phi_{ij\Delta}. \tag{4.13}$$

This equation can clearly be satisfied by defining

$$\xi_{ij} := \frac{\phi_{ij\Delta}}{\boldsymbol{r}^{ij}_{1/2} \cdot \boldsymbol{r}^{ij}_\Delta} \quad \forall i, j, \tag{4.14}$$

implying $\xi_{ij} = \xi_{ji}$ and $\xi_{ii} = 0$. The resulting algorithm is known as the *energy–momentum mid-point algorithm* [8,11,22,41], and does not require a global energy condition of the form (4.8). (This equation is satisfied by $\kappa := 1$.) Since the $\xi_{ij} \equiv \xi_{ij}(\boldsymbol{r}^{ij})$ depend only on the element connecting nodes i and j (if it exists), we call these *elemental conditions* for energy conservation. The energy–momentum algorithm suffers none of the drawbacks mentioned in conjunction with global energy conservation, being well-defined when $\boldsymbol{r}^{ij}_{1/2} \cdot \boldsymbol{r}^{ij}_\Delta = 0$, and has been shown to perform well for a wide range of problems [8,11,32,40].

We now look at the conditions for general $\mathscr{G}, \mathscr{X}$ and $\mathscr{M}$ under which *elemental energy conservation*—that is, energy conservation through satisfaction of simultaneous elemental equations—can be achieved alongside conservation of momenta. Elemental energy conservation is a much more stringent requirement than global energy conservation, with a far greater number of conditions to be satisfied. These amount to balancing each individual $\phi_{ij\Delta}$ term in (4.10) with an algorithmic quantity wholly defined by $\boldsymbol{r}^{ij}$ (the relative position vector for the element joining nodes i and j), i.e.

$$\phi_{ij\Delta} = \alpha_{ij}(\boldsymbol{r}^{ij}) \quad \forall i, j,$$

for some α_{ij}. Note that this is tantamount to stipulating that (4.10) consist *entirely* of elemental terms (i.e. those that are functions of some $\boldsymbol{r}^{ij}$ only), except for those pertaining to the external force: if other terms were to appear, a further (non-

elemental) condition would be required to ensure that they sum to zero. Given the form of $\mathscr{X}$, we have the left-hand side of (4.10) as

$$\mathscr{X}\boldsymbol{R}_{1/2}\cdot\boldsymbol{M}^{-1}\mathscr{M}(\boldsymbol{R}_\Delta-\mathscr{G}^{\mathrm{T}}\boldsymbol{R}_{1/2})=\sum_{\substack{i=1,\\ j>i}}^{N}\xi_{ij}\boldsymbol{r}^{ij}_{1/2}\cdot(\check{\boldsymbol{r}}^{ij}_\Delta-\check{\boldsymbol{r}}^{ij}_{1/2}),$$

where $\check{\boldsymbol{R}}=\boldsymbol{M}^{-1}\mathscr{M}\boldsymbol{R}$ and $\check{\boldsymbol{R}}=\boldsymbol{M}^{-1}\mathscr{M}\mathscr{G}^{\mathrm{T}}\boldsymbol{R}$. Elemental energy conservation then requires $\check{\boldsymbol{r}}^{ij}=a_{ij}\boldsymbol{r}^{ij}$ and $\breve{\boldsymbol{r}}^{ij}=b_{ij}\boldsymbol{r}^{ij}$, and hence we must have $\mathscr{M}:=a\boldsymbol{M}$ and $\mathscr{G}:=\frac{b}{a}\boldsymbol{I}_n$ for scalars a and b (not necessarily constant), i.e. $a_{ij}=a$, $b_{ij}=b\ \forall i,j$. Given that $\mathscr{G}$ is singular (from Proposition 1), however, we must have $\mathscr{G}:=\boldsymbol{0}_{3N}$, meaning $b=0$. So the whole of (4.10) now becomes

$$a\mathscr{X}\boldsymbol{R}_{1/2}\cdot\boldsymbol{R}_\Delta=\phi_\Delta-U^{\mathrm{C}}_\Delta-a\Delta t\mathscr{X}\boldsymbol{R}_{1/2}\cdot\boldsymbol{V}_a+a\boldsymbol{F}_a\cdot\boldsymbol{R}_\Delta+a\Delta t\boldsymbol{F}_a\cdot\boldsymbol{V}_a. \tag{4.15}$$

For problems without external forces, we have $\boldsymbol{F}_a=\boldsymbol{V}_a=\boldsymbol{0}$ from Proposition 2, which requires that $\mathscr{X}:=\frac{1}{a}\langle\xi_{ij}\boldsymbol{I}_3\rangle$ for ξ_{ij} defined by (4.14). For the general conservative case, then, (4.15) becomes

$$-U^{\mathrm{C}}_\Delta-\Delta t\langle\xi_{ij}\boldsymbol{I}_3\rangle\boldsymbol{R}_{1/2}\cdot\boldsymbol{V}_a+a\boldsymbol{F}_a\cdot(\boldsymbol{R}_\Delta+a\boldsymbol{V}_a)=0,$$

which is satisfied by $\boldsymbol{F}_a:=\frac{1}{a}\boldsymbol{F}$, $\boldsymbol{V}_a:=\boldsymbol{0}$. Therefore *the only instances of Algorithm MC that give rise to elemental energy conservation are scaled variations of the energy–momentum mid-point algorithm*, i.e.

$$\mathscr{G}:=\boldsymbol{0}_{3N},\quad \mathscr{X}:=\alpha\langle\xi_{ij}\boldsymbol{I}_3\rangle,\quad \mathscr{M}:=\frac{1}{\alpha}\boldsymbol{M},\quad \boldsymbol{F}_a:=\alpha\boldsymbol{F}\quad\text{and}\quad \boldsymbol{V}_a:=\boldsymbol{0} \tag{4.16}$$

for some $\alpha(\boldsymbol{Z}_{n+1},\boldsymbol{Z}_n,\Delta t)\in\mathbb{R}$, with ξ_{ij} defined by (4.14).

4.4. Preservation of relative equilibria

From (2.39), we have the exact solution of a relative equilibrium problem as

$$\overline{\boldsymbol{R}}(t)=\exp(t\widehat{\boldsymbol{\Omega}}_0)\overline{\boldsymbol{R}}_0,\quad \dot{\overline{\boldsymbol{R}}}(t)=\exp(t\widehat{\boldsymbol{\Omega}}_0)\overline{\boldsymbol{V}}_0\quad\text{and}\quad \boldsymbol{r}^{\mathrm{c}}(t)=\boldsymbol{r}^{\mathrm{c}}_0+t\boldsymbol{v}^{\mathrm{c}}_0$$

with $\exp(t\widehat{\boldsymbol{\Omega}}_0)\equiv\mathrm{diag}[\exp(t\widehat{\boldsymbol{\omega}}_0)]$, where $\overline{\boldsymbol{R}}_0$ and $\overline{\boldsymbol{V}}_0$ are given by

$$\overline{\boldsymbol{R}}_0=\boldsymbol{R}_0-\boldsymbol{R}^{\mathrm{c}}_0,\quad \boldsymbol{R}^{\mathrm{c}}_0:=\langle\underbrace{\boldsymbol{r}^{\mathrm{c}}_0\quad\cdots\quad\boldsymbol{r}^{\mathrm{c}}_0}_{N\text{ times}}\rangle,\quad \overline{\boldsymbol{V}}_0=\boldsymbol{V}_0-\boldsymbol{V}^{\mathrm{c}}_0\quad\text{and}\quad \boldsymbol{V}^{\mathrm{c}}_0:=\langle\underbrace{\boldsymbol{v}^{\mathrm{c}}_0\quad\cdots\quad\boldsymbol{v}^{\mathrm{c}}_0}_{N\text{ times}}\rangle.$$

We note from Section 2 that $\boldsymbol{\omega}_0\times\boldsymbol{v}^{\mathrm{c}}_0=\boldsymbol{0}$, and that $\boldsymbol{R}_0$ and $\boldsymbol{V}_0$ are defined by (2.37). We now define a *relative equilibrium path* as the discrete solution

$$\boldsymbol{R}_k=\boldsymbol{R}^{\mathrm{c}}_k+\exp(\lambda_{\mathrm{r}}k\Delta t\widehat{\boldsymbol{\Omega}}_0)\overline{\boldsymbol{R}}_0,\quad \boldsymbol{P}_k=\boldsymbol{M}\Big(\boldsymbol{V}^{\mathrm{c}}_0+\exp(\lambda_{\mathrm{r}}k\Delta t\widehat{\boldsymbol{\Omega}}_0)\overline{\boldsymbol{V}}_0\Big)\quad\text{and}\quad \boldsymbol{R}^{\mathrm{c}}_k=\boldsymbol{R}^{\mathrm{c}}_0+\lambda_{\mathrm{t}}k\Delta t\boldsymbol{V}^{\mathrm{c}}_0\qquad\forall k, \tag{4.17}$$

where λ_{r} and λ_{t} are constants. Solution (4.17) describes a combination of uniform rotation about an axis of inertia and uniform translation along it, as does the exact solution. However, the angular velocity of the discrete solution is $\lambda_{\mathrm{r}}\boldsymbol{\omega}_0$, and the translational velocity is $\lambda_{\mathrm{t}}\boldsymbol{v}^{\mathrm{c}}_0$. The exact solution is evidently captured if $\lambda_{\mathrm{r}}=\lambda_{\mathrm{t}}=1$; for $\lambda_{\mathrm{r}}=\lambda_{\mathrm{t}}\neq1$, the trajectory of the exact solution is recovered (with different velocities). In general, the discrete solution need not necessarily produce points along the orbit of the exact solution, as was the case for the central-force problem in [32].

For Algorithm MC to produce paths of relative equilibria under initial conditions (2.37), we require that the solution given in (4.17) be inserted into the algorithm without conflict for $k=n$ and $k=n+1$, for certain values of λ_{r} and λ_{t}. We now introduce the notation

$$(\cdot)^{\mathrm{RE}}:=(\cdot)\big|\begin{array}{l}\boldsymbol{R}_n,\boldsymbol{P}_n,\boldsymbol{R}_{n+1},\boldsymbol{P}_{n+1}\text{ defined by (4.17)}\\ \boldsymbol{R}_0,\boldsymbol{V}_0\text{ defined by (2.37) and }\boldsymbol{F}=\boldsymbol{0}\end{array}$$

for a given quantity $(\cdot)$, to denote the value taken when a relative equilibrium solution is in effect. Note that this is equivalent to the notation used in [32] in the case of a central-force problem.

Proposition 4. *Under initial conditions (2.37), Algorithm MC produces paths of relative equilibria provided that*

$$\begin{gathered}\left(\mathscr{X}^{\mathrm{RE}}-\lambda_{\mathrm{r}}\frac{\tan\left(\frac{1}{2}\theta^{\mathrm{RE}}\right)}{\frac{1}{2}\theta^{\mathrm{RE}}}\mathscr{F}_0\right)\overline{\boldsymbol{R}}_0=\boldsymbol{0},\quad \left(\lambda_{\mathrm{r}}\frac{\tan\left(\frac{1}{2}\theta^{\mathrm{RE}}\right)}{\frac{1}{2}\theta^{\mathrm{RE}}}\mathscr{M}^{\mathrm{RE}}-\boldsymbol{M}\right)\overline{\boldsymbol{V}}_0=\boldsymbol{0},\\ \mathscr{G}^{\mathrm{RE}}=\boldsymbol{0}_{3N},\quad (\boldsymbol{M}-\lambda_{\mathrm{t}}\mathscr{M}^{\mathrm{RE}})\boldsymbol{V}^{\mathrm{c}}_0=\boldsymbol{0}\quad\textit{and}\quad \boldsymbol{F}^{\mathrm{RE}}_a=\boldsymbol{V}^{\mathrm{RE}}_a=\boldsymbol{0}\end{gathered} \tag{4.18}$$

for some fixed λ_r, λ_t, where $\theta^{\mathrm{RE}} := \lambda_r\|\boldsymbol{\omega}_0\|\Delta t$ and $-\pi < \theta^{\mathrm{RE}} < \pi$, and that for all n it gives a unique solution for $\boldsymbol{R}_{n+1}$ and $\boldsymbol{P}_{n+1}$ given $\boldsymbol{R}_n$, $\boldsymbol{P}_n$ and Δt.

The proof is given in Appendix A.4. Eq. (4.18) therefore gives criteria that $\mathscr{G}$, $\mathscr{X}$, $\mathscr{M}$, $\boldsymbol{F}_a$ and $\boldsymbol{V}_a$ must satisfy under initial conditions (2.37) for the algorithm to produce a relative equilibrium path. In fact, it is possible to design algorithms to capture the relative equilibrium solution *exactly*, with $\lambda_{\mathrm{r}} = \lambda_{\mathrm{t}} = 1$; an example of this shall be given in Section 6.3.

4.5. Conservation of energy and the preservation of relative equilibria

Lemma 1. *Any algorithm that produces paths of relative equilibria also conserves energy along those paths.*

The proof is given in Appendix A.5; thus no inherent conflict arises from having conservation of energy and preservation of relative equilibria within the same algorithm.

Lemma 2. *Any algorithm that conserves energy globally cannot produce paths of relative equilibria in general, in that an algorithm conserving energy via (4.8) cannot always be well-defined under relative equilibrium conditions.*

This can be argued by contradiction as follows: if an algorithm produces paths of relative equilibria, it satisfies (4.18). It can then be seen from (A.1) in Appendix A.3 that A and B from (4.8) are such that $A^{\mathrm{RE}} = B^{\mathrm{RE}} = 0$, with no obvious value for

$$\lim_{\substack{A\to A^{\mathrm{RE}},\\ B\to B^{\mathrm{RE}}}} \left\{\frac{-B}{A}\right\}$$

that is uniquely defined regardless of how $A \to A^{\mathrm{RE}}$ and $B \to B^{\mathrm{RE}}$. This is one of those instances where κ cannot be evaluated via (4.8), as mentioned in Section 4.3.1.

Lemma 3. *Any algorithm that conserves energy elementally produces paths of relative equilibria such that*

$$\lambda_{\mathrm{r}} = \frac{\frac{1}{2}\alpha^{\mathrm{RE}}\theta^{\mathrm{RE}}}{\tan\left(\frac{1}{2}\theta^{\mathrm{RE}}\right)} \quad \textit{and} \quad \lambda_{\mathrm{t}} = \alpha^{\mathrm{RE}}, \tag{4.19}$$

for $\theta^{\mathrm{RE}} := \lambda_r\|\boldsymbol{\omega}_0\|\Delta t$ and α given in (4.16).

It is straightforward to show that $\mathscr{X}$ defined by (4.12) and (4.14) is such that $\mathscr{X}^{\mathrm{RE}} = \mathscr{F}_0$; the result can then be read directly from (4.16) and (4.18). A corollary of Lemma 3 is that *no elementally energy-conserving algorithm can capture the exact trajectory of a general relative equilibrium problem*, given that $\tan^{-1} x \neq x$ for $x \neq 0$. (For the special cases of pure translation and pure rotation, the exact trajectories can be recovered.)

Taken together, these three lemmas imply that instances of Algorithm MC are unlikely to both conserve energy and produce paths of relative equilibria unless they are scaled variations of the energy–momentum mid-point algorithm, as given by (4.16).

4.6. Time reversibility

An algorithm is described as time-reversible if, at any given configuration $\boldsymbol{Z}_{n+1}$, applying a negative time-step of $-\Delta t$ recovers the previous configuration $\boldsymbol{Z}_n$ [1,21]. From (3.2), an algorithm is thus time-reversible if

$$\boldsymbol{Z}_{n+1} = \mathscr{B}(\boldsymbol{Z}_{n+1}, \boldsymbol{Z}_n, \Delta t)\boldsymbol{Z}_n + \boldsymbol{Z}_F(\boldsymbol{Z}_{n+1}, \boldsymbol{Z}_n, \Delta t) \iff \boldsymbol{Z}_n = \mathscr{B}(\boldsymbol{Z}_n, \boldsymbol{Z}_{n+1}, -\Delta t)\boldsymbol{Z}_{n+1} + \boldsymbol{Z}_F(\boldsymbol{Z}_n, \boldsymbol{Z}_{n+1}, -\Delta t). \tag{4.20}$$

We now introduce for any quantity $(\cdot)$ the notation

$$(\cdot)^{\mathrm{TR}} := (\cdot)|_{\boldsymbol{Z}_{n+1}\leftrightarrow\boldsymbol{Z}_n,\Delta t\leftrightarrow-\Delta t}.$$

Proposition 5. *Algorithm MC is time-reversible if*

$$\mathscr{X}^{\mathrm{TR}} = \mathscr{X}, \quad \mathscr{M}^{\mathrm{TR}} = \mathscr{M}, \quad \mathscr{G}^{\mathrm{TR}} = -\mathscr{G}, \quad \boldsymbol{F}_a^{\mathrm{TR}} = \boldsymbol{F}_a \quad \textit{and} \quad \boldsymbol{V}_a^{\mathrm{TR}} = \boldsymbol{V}_a. \tag{4.21}$$

The proof is given in Appendix A.6.

5. Local accuracy analysis

We now analyse the local accuracy characteristics of Algorithm MC, and investigate its capacity for higher-order accuracy when applied to general non-linear problems. We will also derive the series form of the exact solution. This section takes a very similar approach to that of [32].

We define the *local error vector* as

$$\boldsymbol{\epsilon} := \boldsymbol{Z}_{n+1} - \boldsymbol{Z}(t_{n+1}) \quad \text{when } \boldsymbol{Z}_n = \boldsymbol{Z}(t_n), \tag{5.1}$$

with $\boldsymbol{Z} = \langle \boldsymbol{R} \quad \boldsymbol{P} \rangle$ as before: throughout this section, we will assume the solution at time-step n to be exact, i.e. $\boldsymbol{Z}_n = \boldsymbol{Z}(t_n)$. We also define the *residual vector*

$$\boldsymbol{g}(\boldsymbol{X}) := \mathscr{B}(\boldsymbol{X}, \boldsymbol{Z}_n, \Delta t)\boldsymbol{Z}_n + \boldsymbol{Z}_F(\boldsymbol{X}, \boldsymbol{Z}_n, \boldsymbol{F}, \Delta t) - \boldsymbol{X}, \tag{5.2}$$

where $\mathscr{B}$ and $\boldsymbol{Z}_F$ were introduced in Section 3. Consequently we have

$$\boldsymbol{g}(\boldsymbol{Z}_{n+1}) = \boldsymbol{0} \text{ and hence } \boldsymbol{g}[\boldsymbol{Z}(t_{n+1})] = \boldsymbol{g}(\boldsymbol{Z}_{n+1} - \boldsymbol{\epsilon}) = -\nabla \boldsymbol{g}(\boldsymbol{Z}_{n+1})\boldsymbol{\epsilon} + \mathscr{O}(\|\boldsymbol{\epsilon}\|^2),$$

where $\nabla \boldsymbol{g}$ is known as the *Jacobian matrix*. Given that $\boldsymbol{g}(\boldsymbol{X}) \in \mathscr{O}(1)$ (i.e. $\mathscr{O}[\Delta t^0]$) for general $\boldsymbol{X}$, we have $\nabla \boldsymbol{g}(\boldsymbol{Z}_{n+1}) \in \mathscr{O}(1)$ also; thus the dependence of $\boldsymbol{g}[\boldsymbol{Z}(t_{n+1})]$ on Δt reveals the size of the local error $\boldsymbol{\epsilon}$.

We begin by introducing the abbreviations

$$\boldsymbol{\zeta} := \langle \boldsymbol{Z}(t_{n+1})\boldsymbol{Z}(t_n)\Delta t \rangle \quad \text{and} \quad \tilde{\boldsymbol{\zeta}} := \langle \boldsymbol{Z}(t_{n+1})\boldsymbol{Z}(t_n)\boldsymbol{F}\Delta t \rangle,$$

and setting $\boldsymbol{X} = \boldsymbol{Z}(t_{n+1})$ in (5.2) to get

$$\boldsymbol{g}[\boldsymbol{Z}(t_{n+1})] := \mathscr{B}(\boldsymbol{\zeta})\boldsymbol{Z}_n + \boldsymbol{Z}_F(\tilde{\boldsymbol{\zeta}}) - \boldsymbol{Z}(t_{n+1}). \tag{5.3}$$

Assuming that $\boldsymbol{Z}(t)$ is analytic in a neighbourhood of t_n, we have

$$\boldsymbol{Z}(t_{n+1}) = \boldsymbol{Z}(t_n + \Delta t) = \sum_{s=0}^{\infty} \frac{\boldsymbol{Z}^{(s)}(t_n)}{s!}\Delta t^s, \tag{5.4}$$

where $(\cdot)^{(s)} \equiv \frac{\mathrm{d}^s}{\mathrm{d}t^s}\{(\cdot)\}$. We now express (2.21) in matrix form as

$$\dot{\boldsymbol{Z}} = \boldsymbol{\Psi}\boldsymbol{Z} + \widetilde{\boldsymbol{F}}, \quad \text{where } \boldsymbol{\Psi}(t) = \begin{pmatrix} \boldsymbol{0}_{3N} & \boldsymbol{M}^{-1} \\ -\mathscr{F} & \boldsymbol{0}_{3N} \end{pmatrix} \text{ and } \widetilde{\boldsymbol{F}}(t) = \begin{Bmatrix} \boldsymbol{0} \\ \boldsymbol{F} \end{Bmatrix}. \tag{5.5}$$

We now derive the series solution of $(5.5)_1$ for known data at time t_n. By repeated differentiation of $(5.5)_1$, we can express the derivative $\boldsymbol{Z}^{(s+1)}$ in terms of lower-order derivatives of $\boldsymbol{Z}$ and $\boldsymbol{\Psi}$, i.e.

$$\ddot{\boldsymbol{Z}} = \dot{\boldsymbol{\Psi}}\boldsymbol{Z} + \boldsymbol{\Psi}\dot{\boldsymbol{Z}} + \dot{\widetilde{\boldsymbol{F}}}, \quad \boldsymbol{Z}^{(3)} = \ddot{\boldsymbol{\Psi}}\boldsymbol{Z} + 2\dot{\boldsymbol{\Psi}}\dot{\boldsymbol{Z}} + \boldsymbol{\Psi}\ddot{\boldsymbol{Z}} + \ddot{\widetilde{\boldsymbol{F}}},$$
$$\boldsymbol{Z}^{(4)} = \boldsymbol{\Psi}^{(3)}\boldsymbol{Z} + 3\ddot{\boldsymbol{\Psi}}\dot{\boldsymbol{Z}} + 3\dot{\boldsymbol{\Psi}}\ddot{\boldsymbol{Z}} + \boldsymbol{\Psi}\boldsymbol{Z}^{(3)} + \widetilde{\boldsymbol{F}}^{(3)}, \ldots$$

and so on. Summarising this procedure, we have

$$\boldsymbol{Z}^{(s+1)} = \sum_{r=0}^{s} \binom{s}{r} \boldsymbol{\Psi}^{(s-r)}\boldsymbol{Z}^{(r)} + \widetilde{\boldsymbol{F}}^{(s)}, \tag{5.6}$$

where $\binom{s}{r}$ represents the binomial coefficient $\frac{s!}{(s-r)!r!}$. Inserting (5.6) into (5.4) gives

$$\begin{aligned}
\boldsymbol{Z}(t_{n+1}) &= \boldsymbol{Z}_n + \sum_{s=0}^{\infty} \frac{\boldsymbol{Z}_n^{(s+1)}}{(s+1)!}\Delta t^{s+1} = \boldsymbol{Z}_n + \sum_{s=0}^{\infty} \frac{\Delta t^{s+1}}{(s+1)!}\left[\sum_{r=0}^{s} \binom{s}{r} \boldsymbol{\Psi}_n^{(s-r)}\boldsymbol{Z}_n^{(r)} + \widetilde{\boldsymbol{F}}_n^{(s)}\right] \\
&= \boldsymbol{Z}_n + \sum_{s=0}^{\infty} \frac{\Delta t^{s+1}}{(s+1)!}\left(\boldsymbol{\Psi}_n^{(s)}\boldsymbol{Z}_n + \widetilde{\boldsymbol{F}}_n^{(s)}\right) + \sum_{s=0}^{\infty} \frac{\Delta t^{s+1}}{(s+1)!}\sum_{r=1}^{s} \binom{s}{r} \boldsymbol{\Psi}_n^{(s-r)}\boldsymbol{Z}_n^{(r)} \\
&= \boldsymbol{Z}_n + \sum_{s=0}^{\infty} \frac{\Delta t^{s+1}}{(s+1)!}\left(\boldsymbol{\Psi}_n^{(s)}\boldsymbol{Z}_n + \widetilde{\boldsymbol{F}}_n^{(s)}\right) + \sum_{s=0}^{\infty} \frac{\Delta t^{s+2}}{(s+2)!}\sum_{r=0}^{s} \binom{s+1}{r+1} \boldsymbol{\Psi}_n^{(s-r)}\boldsymbol{Z}_n^{(r+1)},
\end{aligned}$$

where $\sum_{r=1}^{s}(\cdot) := 0$ for $s < r$. Continuing in the same manner, we now substitute for $\boldsymbol{Z}_n^{(r+1)}$ using (5.6) to get

$$\begin{aligned}
\boldsymbol{Z}(t_{n+1}) = \boldsymbol{Z}_n &+ \sum_{s=0}^{\infty} \frac{\Delta t^{s+1}}{(s+1)!}\left(\boldsymbol{\Psi}_n^{(s)}\boldsymbol{Z}_n + \widetilde{\boldsymbol{F}}_n^{(s)}\right) + \sum_{s=0}^{\infty} \frac{\Delta t^{s+2}}{(s+2)!}\sum_{r=0}^{s} \binom{s+1}{r+1} \boldsymbol{\Psi}_n^{(s-r)}\left(\boldsymbol{\Psi}^{(r)}\boldsymbol{Z}_n + \widetilde{\boldsymbol{F}}^{(r)}\right) \\
&+ \sum_{s=0}^{\infty} \frac{\Delta t^{s+3}}{(s+3)!}\sum_{r=0}^{s} \binom{s+2}{r+2} \boldsymbol{\Psi}_n^{(s-r)}\sum_{q=0}^{r} \binom{r+1}{q+1} \boldsymbol{\Psi}^{(r-q)}\boldsymbol{Z}_n^{(q+1)}.
\end{aligned}$$

The expression for $\boldsymbol{Z}(t_{n+1})$ can now be seen to consist of a sum of terms of the form

$$\left[\sum_{s_0=0}^{\infty}\frac{\Delta t^{s_0+m+1}}{(s_0+m+1)!}\sum_{s_1=0}^{s_0}\binom{s_0+m}{s_1+m}\boldsymbol{\Psi}_n^{(s_0-s_1)}\sum_{s_2=0}^{s_1}\binom{s_1+m-1}{s_2+m-1}\boldsymbol{\Psi}^{(s_1-s_2)}\cdots\sum_{s_m=0}^{s_{m-1}}\binom{s_{m-1}+1}{s_m+1}\boldsymbol{\Psi}^{(s_{m-1}-s_m)}\right]\boldsymbol{\mathcal{V}} \tag{5.7}$$

for $m=0,1,2\ldots$, where $\boldsymbol{\mathcal{V}}$ stands for either $\boldsymbol{\Psi}^{(s_m)}\boldsymbol{Z}_n$ or $\widetilde{\boldsymbol{F}}^{(s_m)}$. We can write (5.7) more compactly as

$$\sum_{s_0=0}^{\infty}\frac{\Delta t^{s_0+m+1}}{(s_0+m+1)!}\left(\prod_{i=0}^{m-1}\left[\sum_{s_{i+1}=0}^{s_i}\binom{s_i+m-i}{s_{i+1}+m-i}\boldsymbol{\Psi}_n^{(s_i-s_{i+1})}\right]\right)\boldsymbol{\mathcal{V}},$$

with $s_{-1}:=-\infty$ in the case $m=0$. Thus we can express the solution to $(5.5)_1$ at time t_{n+1} in (relatively) compact form as

$$\boldsymbol{Z}(t_{n+1})=\boldsymbol{\mathscr{B}}^{\mathrm{e}}\boldsymbol{Z}_n+\boldsymbol{Z}_F^{\mathrm{e}}, \tag{5.8}$$

where

$$\begin{aligned}\boldsymbol{\mathscr{B}}^{\mathrm{e}}&:=\sum_{m=0}^{\infty}\left[\sum_{s_0=0}^{\infty}\frac{\Delta t^{s_0+m+1}}{(s_0+m+1)!}\left(\prod_{i=0}^{m-1}\left[\sum_{s_{i+1}=0}^{s_i}\binom{s_i+m-i}{s_{i+1}+m-i}\boldsymbol{\Psi}_n^{(s_i-s_{i+1})}\right]\boldsymbol{\Psi}_n^{(s_m)}\right)\right]+\boldsymbol{I}_{6N}\quad\text{and}\\ \boldsymbol{Z}_F^{\mathrm{e}}&:=\sum_{m=0}^{\infty}\left[\sum_{s_0=0}^{\infty}\frac{\Delta t^{s_0+m+1}}{(s_0+m+1)!}\left(\prod_{i=0}^{m-1}\left[\sum_{s_{i+1}=0}^{s_i}\binom{s_i+m-i}{s_{i+1}+m-i}\boldsymbol{\Psi}_n^{(s_i-s_{i+1})}\right]\widetilde{\boldsymbol{F}}_n^{(s_m)}\right)\right].\end{aligned} \tag{5.9}$$

(Note that for $m=0$, we have $\prod_{i=0}^{m-1}(\cdot)_i:=\boldsymbol{I}_{6N}$.) Eqs. (5.8) and (5.9) thus give the exact solution to the semi-discrete equations of motion (2.21), to which we can compare the solutions obtained from our algorithms. From (5.3), (5.8) and (5.9) we can see that the order of $\boldsymbol{g}[\boldsymbol{Z}(t_{n+1})]$ with respect to Δt is governed by how closely $\boldsymbol{\mathscr{B}}(\zeta)$ and $\boldsymbol{Z}_F(\tilde{\zeta})$ match $\boldsymbol{\mathscr{B}}^{\mathrm{e}}$ and $\boldsymbol{Z}_F^{\mathrm{e}}$, respectively. Using (5.3), we see that since $\boldsymbol{Z}_n\in\mathscr{O}(1)$,

$$\boldsymbol{g}[\boldsymbol{Z}(t_{n+1})]\in\mathscr{O}(\Delta t^{p+1})\iff\boldsymbol{\mathscr{B}}(\zeta)-\boldsymbol{\mathscr{B}}^{\mathrm{e}}\in\mathscr{O}(\Delta t^{p+p_1}),\ \boldsymbol{Z}_F(\tilde{\zeta})-\boldsymbol{Z}_F^{\mathrm{e}}\in\mathscr{O}(\Delta t^{p+p_2}),$$

where $p_1,p_2\in\mathbb{Z}^+$ are such that $\mathrm{Min}\{p_1,p_2\}=1$, and thus

$$\boldsymbol{\epsilon}\in\mathscr{O}(\Delta t^{p+1})\iff\boldsymbol{\mathscr{B}}(\zeta)-\boldsymbol{\mathscr{B}}^{\mathrm{e}}\in\mathscr{O}(\Delta t^{p+p_1}),\ \boldsymbol{Z}_F(\tilde{\zeta})-\boldsymbol{Z}_F^{\mathrm{e}}\in\mathscr{O}(\Delta t^{p+p_2}). \tag{5.10}$$

Eq. (5.10) thus contains the criteria for Algorithm MC to be pth-order accurate. To express this in terms of parameters $\mathscr{G}$, $\mathscr{X}$, $\mathscr{M}$, $\boldsymbol{F}_a$ and $\boldsymbol{V}_a$, we note from (3.12) that

$$\left[\boldsymbol{I}_{6N}-\frac{1}{2}\mathscr{C}(\zeta)\right]\boldsymbol{\mathscr{B}}(\zeta)=\left[\boldsymbol{I}_{6N}+\frac{1}{2}\mathscr{C}(\zeta)\right]\quad\text{and}\quad\left[\boldsymbol{I}_{6N}-\frac{1}{2}\mathscr{C}(\zeta)\right]\boldsymbol{Z}_F(\tilde{\zeta})=\boldsymbol{Z}_a(\tilde{\zeta}),$$

with $\mathscr{C}$ and $\boldsymbol{Z}_a$ given in (3.11) (and repeated here for convenience) as

$$\mathscr{C}=\begin{pmatrix}\mathscr{G}^{\mathrm{T}}&\Delta t\mathscr{M}^{-1}\\-\Delta t\mathscr{X}&-\mathscr{G}\end{pmatrix}\quad\text{and}\quad\boldsymbol{Z}_a=\Delta t\begin{Bmatrix}-\boldsymbol{V}_a\\\boldsymbol{F}_a\end{Bmatrix}. \tag{5.11}$$

Therefore pre-multiplying (5.10) by $-\left[\boldsymbol{I}_{6N}-\frac{1}{2}\mathscr{C}(\zeta)\right]$ gives us

$$\boldsymbol{\epsilon}\in\mathscr{O}(\Delta t^{p+1})\iff\left[\boldsymbol{I}_{6N}-\frac{1}{2}\mathscr{C}(\zeta)\right]\boldsymbol{\mathscr{B}}^{\mathrm{e}}-\left[\boldsymbol{I}_{6N}+\frac{1}{2}\mathscr{C}(\zeta)\right]\in\mathscr{O}(\Delta t^{p+p_1}),\left[\boldsymbol{I}_{6N}-\frac{1}{2}\mathscr{C}(\zeta)\right]\boldsymbol{Z}_F^{\mathrm{e}}-\boldsymbol{Z}_a(\tilde{\zeta})\in\mathscr{O}(\Delta t^{p+p_2}), \tag{5.12}$$

since the left-hand side is unaffected provided that $\mathscr{C}(\zeta)$ is bounded as $\Delta t\to 0$. We assume that these parameter matrices are also analytic functions of t within the same neighbourhood of t_n as $\boldsymbol{Z}(t)$, and thus we can write

$$\mathscr{C}(\zeta)=\sum_{s=0}^{\infty}\mathscr{C}_s(t_n)\Delta t^s\quad\text{and}\quad\boldsymbol{Z}_a(\tilde{\zeta})=\sum_{s=0}^{\infty}\boldsymbol{Z}_{a,s}(t_n)\Delta t^s, \tag{5.13}$$

where the coefficients are wholly determined at time t_n. This implies similar series expansions for $\mathscr{G}$, $\mathscr{X}$, $\mathscr{M}^{-1}$, $\boldsymbol{F}_a$ and $\boldsymbol{V}_a$, and we can relate the expansion of $\mathscr{M}^{-1}$ to that of $\mathscr{M}$ via the formula

$$\begin{aligned}\mathscr{M}(\zeta)=\sum_{s=0}^{\infty}\mathscr{M}_s\Delta t^s=\mathscr{M}_0\left(\boldsymbol{I}_{3N}+\sum_{s=0}^{\infty}\mathscr{M}_0^{-1}\mathscr{M}_{s+1}\Delta t^{s+1}\right)\ &\Rightarrow\ \mathscr{M}^{-1}(\zeta)=\left(\boldsymbol{I}_{3N}+\sum_{s=0}^{\infty}\mathscr{M}_0^{-1}\mathscr{M}_{s+1}\Delta t^{s+1}\right)^{-1}\mathscr{M}_0^{-1}\\&=\sum_{r=0}^{\infty}\left(-\sum_{s=0}^{\infty}\mathscr{M}_0^{-1}\mathscr{M}_{s+1}\Delta t^{s+1}\right)^r\mathscr{M}_0^{-1},\end{aligned} \tag{5.14}$$

which is derived using the binomial formula (e.g. [34]) and valid for sufficiently small Δt. From (5.9), we also have

$$\mathscr{B}^{\mathrm{e}} = \sum_{s=0}^{\infty} \mathscr{B}^{\mathrm{e}}_s(t_n)\Delta t^s \quad \text{and} \quad \boldsymbol{Z}^{\mathrm{e}}_F = \sum_{s=0}^{\infty} \boldsymbol{Z}^{\mathrm{e}}_{F,s}(t_n)\Delta t^s. \tag{5.15}$$

Therefore (5.12), (5.13) and (5.15) combine to give

$$\begin{aligned} \boldsymbol{\epsilon} \in \mathscr{O}(\Delta t^{p+1}) \iff \mathscr{B}^{\mathrm{e}}_s - \frac{1}{2}\sum_{r=0}^{s} \mathscr{C}_r \mathscr{B}^{\mathrm{e}}_{s-r} - \frac{1}{2}\mathscr{C}_s = \begin{cases} \boldsymbol{I}_{6N} : & s = 0, \\ \boldsymbol{0}_{6N} : & s = 1, \ldots, p \end{cases} \\ \text{and } \boldsymbol{Z}^{\mathrm{e}}_{F,s} - \frac{1}{2}\sum_{r=0}^{s} \mathscr{C}_r \boldsymbol{Z}^{\mathrm{e}}_{F,s-r} = \boldsymbol{Z}_{a,s} \text{ for } s = 0, \ldots, p. \end{aligned} \tag{5.16}$$

We will now derive cumulative criteria for pth-order accuracy when $p = 0, 1, 2$.

5.1. Zeroth-order accuracy

For $p = 0$, we have

$$\mathscr{B}^{\mathrm{e}}_0 - \frac{1}{2}\mathscr{C}_0 \mathscr{B}^{\mathrm{e}}_0 - \frac{1}{2}\mathscr{C}_0 = \boldsymbol{I}_{6N} \quad \text{and} \quad \boldsymbol{Z}^{\mathrm{e}}_{F,0} - \frac{1}{2}\mathscr{C}_0 \boldsymbol{Z}^{\mathrm{e}}_{F,0} = \boldsymbol{Z}_{a,0}.$$

From (5.9) we have

$$\mathscr{B}^{\mathrm{e}}_0 = \boldsymbol{I}_{6N} \quad \text{and} \quad \boldsymbol{Z}^{\mathrm{e}}_{F,0} = \boldsymbol{0}, \tag{5.17}$$

and so we have zeroth-order accuracy if and only if

$$\mathscr{C}_0 = \boldsymbol{0}_{6N} \quad \text{and} \quad \boldsymbol{Z}_{a,0} = \boldsymbol{0}. \tag{5.18}$$

From (5.11), this equates to

$$\mathscr{G}_0 = \boldsymbol{0}_{3N}. \tag{5.19}$$

Note that this property is *not* sufficient to assure convergence to the true solution as $\Delta t \to 0$. An algorithm that is zeroth-order accurate will in fact give $\boldsymbol{Z}_n = \boldsymbol{Z}_0 \ \forall n$ using an infinitesimal time-step size, which can be considered the most basic approximation to the true solution.

5.2. First-order accuracy

For $p = 1$, the conditions include those for $p = 0$ and also

$$\mathscr{B}^{\mathrm{e}}_1 - \frac{1}{2}\mathscr{C}_0 \mathscr{B}^{\mathrm{e}}_1 - \frac{1}{2}\mathscr{C}_1 \mathscr{B}^{\mathrm{e}}_0 - \frac{1}{2}\mathscr{C}_1 = \boldsymbol{0}_{6N} \quad \text{and} \quad \boldsymbol{Z}^{\mathrm{e}}_{F,1} - \frac{1}{2}\mathscr{C}_0 \boldsymbol{Z}^{\mathrm{e}}_{F,1} - \frac{1}{2}\mathscr{C}_1 \boldsymbol{Z}^{\mathrm{e}}_{F,0} = \boldsymbol{Z}_{a,1}$$

from (5.16). After incorporating (5.17) and (5.18), these become

$$\mathscr{C}_1 = \mathscr{B}^{\mathrm{e}}_1 \quad \text{and} \quad \boldsymbol{Z}_{a,1} = \boldsymbol{Z}^{\mathrm{e}}_{F,1}.$$

From (5.9) we see that

$$\mathscr{B}^{\mathrm{e}} = \boldsymbol{I}_{6N} + \underbrace{\sum\nolimits_{s_0=0}^{\infty} \frac{\Delta t^{s_0+1}}{(s_0+1)!} \boldsymbol{\Psi}^{(s_0)}_n}_{m=0} + \mathscr{O}(\Delta t^2)$$

and similarly for $\boldsymbol{Z}^{\mathrm{e}}_F$; thus we have

$$\mathscr{B}^{\mathrm{e}}_1 = \boldsymbol{\Psi}_n \quad \text{and} \quad \boldsymbol{Z}^{\mathrm{e}}_{F,1} = \widetilde{\boldsymbol{F}}_n. \tag{5.20}$$

Therefore we have first-order accuracy if and only if

$$\mathscr{C}_1 = \boldsymbol{\Psi}_n \quad \text{and} \quad \boldsymbol{Z}_{a,1} = \widetilde{\boldsymbol{F}}_n. \tag{5.21}$$

Using (5.11) and $(5.5)_{2,3}$, this gives us

$$\mathscr{G}_1 = \boldsymbol{0}_{3N}, \quad \mathscr{M}_0 = \boldsymbol{M}, \quad \mathscr{X}_0 = \mathscr{F}_n, \quad \boldsymbol{F}_{a,0} = \boldsymbol{F}_n \quad \text{and} \quad \boldsymbol{V}_{a,0} = \boldsymbol{0}, \tag{5.22}$$

after using (5.14). This property is known as *consistency*, and implies convergence for stable algorithms.

5.3. Second-order accuracy

For $p = 2$, the conditions include those for $p = 1$ and also

$$\mathscr{B}_2^{\mathrm{e}} - \frac{1}{2}\mathscr{C}_0\mathscr{B}_2^{\mathrm{e}} - \frac{1}{2}\mathscr{C}_1\mathscr{B}_1^{\mathrm{e}} - \frac{1}{2}\mathscr{C}_2\mathscr{B}_0^{\mathrm{e}} - \frac{1}{2}\mathscr{C}_2 = \mathbf{0}_{6N} \quad \text{and}$$
$$\boldsymbol{Z}_{F,2}^{\mathrm{e}} - \frac{1}{2}\mathscr{C}_0\boldsymbol{Z}_{F,2}^{\mathrm{e}} - \frac{1}{2}\mathscr{C}_1\boldsymbol{Z}_{F,1}^{\mathrm{e}} - \frac{1}{2}\mathscr{C}_2\boldsymbol{Z}_{F,0}^{\mathrm{e}} = \boldsymbol{Z}_{a,2}$$

from (5.16). After incorporating (5.17), (5.18), (5.20) and (5.21), these reduce to

$$\mathscr{C}_2 = \mathscr{B}_2^{\mathrm{e}} - \frac{1}{2}(\boldsymbol{\Psi}_n)^2 \quad \text{and} \quad \boldsymbol{Z}_{a,2} = \boldsymbol{Z}_{F,2}^{\mathrm{e}} - \frac{1}{2}\boldsymbol{\Psi}_n\widetilde{\boldsymbol{F}}_n.$$

From (5.9) we see that

$$\mathscr{B}^{\mathrm{e}} = \boldsymbol{I}_{6N} + \underbrace{\sum_{s_0=0}^{\infty}\frac{\Delta t^{s_0+1}}{(s_0+1)!}\boldsymbol{\Psi}_n^{(s_0)}}_{m=0} + \underbrace{\sum_{s_0=0}^{\infty}\frac{\Delta t^{s_0+2}}{(s_0+2)!}\sum_{s_1=0}^{s_0}\binom{s_0+1}{s_1+1}\boldsymbol{\Psi}_n^{(s_0-s_1)}\boldsymbol{\Psi}_n^{(s_1)}}_{m=1} + \mathscr{O}(\Delta t^3)$$

and similarly for $\boldsymbol{Z}_F^{\mathrm{e}}$; thus we have

$$\mathscr{B}_2^{\mathrm{e}} = \frac{1}{2}\dot{\boldsymbol{\Psi}}_n + \frac{1}{2}(\boldsymbol{\Psi}_n)^2 \quad \text{and} \quad \boldsymbol{Z}_{F,2}^{\mathrm{e}} = \frac{1}{2}\dot{\widetilde{\boldsymbol{F}}}_n + \frac{1}{2}\boldsymbol{\Psi}_n\widetilde{\boldsymbol{F}}_n.$$

Therefore we have second-order accuracy if and only if

$$\mathscr{C}_2 = \frac{1}{2}\dot{\boldsymbol{\Psi}}_n \quad \text{and} \quad \boldsymbol{Z}_{a,2} = \frac{1}{2}\dot{\widetilde{\boldsymbol{F}}}_n. \tag{5.23}$$

Using (5.11) and $(5.5)_{2,3}$, this becomes

$$\mathscr{G}_2 = \mathscr{M}_1 = \mathbf{0}_{3N}, \quad \mathscr{X}_1 = \frac{1}{2}\dot{\mathscr{F}}_n, \quad \boldsymbol{F}_{a,1} = \frac{1}{2}\dot{\boldsymbol{F}}_n \quad \text{and} \quad \boldsymbol{V}_{a,1} = \boldsymbol{0}, \tag{5.24}$$

after again using (5.14).

It appears as though this process can be continued indefinitely. This is indeed the case, and can be shown inductively as follows: Suppose Algorithm MC is $(p-1)$th-order accurate for $p \geqslant 0$. Since $\mathscr{B}_0^{\mathrm{e}} = \boldsymbol{I}_{6N}$ we have, from (5.16),

$$\mathscr{C}_s = \mathscr{B}_s^{\mathrm{e}} - \frac{1}{2}\sum_{r=0}^{s-1}\mathscr{C}_r\mathscr{B}_{s-r}^{\mathrm{e}} \quad \text{and} \quad \boldsymbol{Z}_{a,s} = \boldsymbol{Z}_{F,s}^{\mathrm{e}} - \frac{1}{2}\sum_{r=0}^{s}\mathscr{C}_r\boldsymbol{Z}_{F,s-r}^{\mathrm{e}} \tag{5.25}$$

as the requirements for pth-order accuracy. Since $\mathscr{C}_r; 0 \leqslant r \leqslant s-1$ and $\boldsymbol{Z}_{F,s}$, $\mathscr{B}_s^{\mathrm{e}}; 0 \leqslant s < \infty$ are all known, (5.25) can be immediately solved (in order) to furnish $\mathscr{C}_s$ and $\boldsymbol{Z}_{a,s}$. Hence with appropriate choices for $\mathscr{G}$, $\mathscr{X}$, $\mathscr{M}$, $\boldsymbol{F}_a$ and $\boldsymbol{V}_a$, *Algorithm MC can be made arbitrarily accurate*. The criteria for accuracy up to fifth order are given in Table 1, and are continued up to eighth order in [40]. We note the symmetries of the expressions for $\mathscr{G}$ and $\boldsymbol{V}_a$, and also for $\mathscr{X}$ and $\boldsymbol{F}_a$.

From Table 1, we see immediately that *for time-integration schemes with constant $\mathscr{G} := \mathscr{G}_0$ or $\mathscr{M} := \mathscr{M}_0$, the limit is second-order accuracy for problems with general strain energy functions $\phi(I)$*, which is consistent with our results in [32]. Therefore schemes defined by (4.16), that conserve energy elementally, cannot be higher than second-order accurate; in turn, this means that higher-order schemes are unlikely to conserve energy and preserve relative equilibria, as discussed in Section 4.5. We also note that they will not retain the sparsity of time-integration schemes of the form (3.10), and will thus be computationally more expensive.

We now verify that the accuracy requirements given in this section do not conflict with the conservation conditions from Section 4 by encapsulating the fact that higher-order accuracy does not hinder conservation of a physical quantity in the following result:

Proposition 6. *Let $\boldsymbol{f}[\boldsymbol{Z}(t)]$ be a constant quantity of the motion governed by (5.5). Then*

$$\boldsymbol{\epsilon} \in \mathscr{O}(\Delta t^{p+1}) \;\Rightarrow\; \boldsymbol{f}(\boldsymbol{Z}_{n+1}) - \boldsymbol{f}(\boldsymbol{Z}_n) \in \mathscr{O}(\Delta t^{p+q}),$$

where $q \geqslant 1$, assuming $\boldsymbol{Z}_n = \boldsymbol{Z}(t_n)$. That is to say, any pth-order algorithm will conserve a constant of motion up to order p or higher.

The proof is given in Appendix A.7. Hence *any algorithm that does not conserve one or more of the constants of motion must be limited in its order of accuracy*.

Table 1
Cumulative conditions for pth-order accuracy

p	Conditions needed
0	$\mathscr{G}_0 = \mathbf{0}_{3N}$
1	$\mathscr{G}_1 = \mathbf{0}_{3N}, \quad \mathscr{X}_0 = \mathscr{F}_n, \quad \mathscr{M}_0 = \mathbf{M}, \quad \mathbf{F}_{a,0} = \mathbf{F}_n, \quad \mathbf{V}_{a,0} = \mathbf{0}$
2	$\mathscr{G}_2 = \mathbf{0}_{3N}, \quad \mathscr{X}_1 = \frac{1}{2}\dot{\mathscr{F}}_n, \quad \mathscr{M}_1 = \mathbf{0}_{3N}, \quad \mathbf{F}_{a,1} = \frac{1}{2}\dot{\mathbf{F}}_n, \quad \mathbf{V}_{a,1} = \mathbf{0}$
3	$\mathscr{G}_3 = \frac{1}{12}\dot{\mathscr{F}}_n\mathbf{M}^{-1}, \quad \mathscr{X}_2 = \frac{1}{6}\ddot{\mathscr{F}}_n + \frac{1}{12}\mathscr{F}_n\mathbf{M}^{-1}\mathscr{F}_n, \quad \mathscr{M}_2 = -\frac{1}{12}\mathscr{F}_n,$ $\mathbf{F}_{a,2} = \frac{1}{6}\ddot{\mathbf{F}}_n + \frac{1}{12}\mathscr{F}_n\mathbf{M}^{-1}\mathbf{F}_n, \quad \mathbf{V}_{a,2} = \frac{1}{12}\mathbf{M}^{-1}\dot{\mathbf{F}}_n$
4	$\mathscr{G}_4 = \frac{1}{24}\ddot{\mathscr{F}}_n\mathbf{M}^{-1}, \quad \mathscr{X}_3 = \frac{1}{24}(\mathscr{F}_n^{(3)} + \mathscr{F}_n\mathbf{M}^{-1}\dot{\mathscr{F}}_n + \dot{\mathscr{F}}_n\mathbf{M}^{-1}\mathscr{F}_n),$ $\mathscr{M}_3 = -\frac{1}{24}\dot{\mathscr{F}}_n, \quad \mathbf{F}_{a,3} = \frac{1}{24}(\mathbf{F}_n^{(3)} + \mathscr{F}_n\mathbf{M}^{-1}\dot{\mathbf{F}}_n + \dot{\mathscr{F}}_n\mathbf{M}^{-1}\mathbf{F}_n),$ $\mathbf{V}_{a,3} = \frac{1}{24}\mathbf{M}^{-1}\ddot{\mathbf{F}}_n$
5	$\mathscr{G}_5 = \frac{1}{80}\mathscr{F}_n^{(3)}\mathbf{M}^{-1} + \frac{1}{120}\dot{\mathscr{F}}_n\mathbf{M}^{-1}\mathscr{F}_n\mathbf{M}^{-1} + \frac{1}{240}\mathscr{F}_n\mathbf{M}^{-1}\dot{\mathscr{F}}_n\mathbf{M}^{-1},$ $\mathscr{X}_4 = \frac{7}{240}\dot{\mathscr{F}}_n\mathbf{M}^{-1}\dot{\mathscr{F}}_n + \frac{1}{80}(\ddot{\mathscr{F}}_n\mathbf{M}^{-1}\mathscr{F}_n + \mathscr{F}_n\mathbf{M}^{-1}\ddot{\mathscr{F}}_n) + \frac{1}{120}(\mathscr{F}_n^{(4)} + \mathscr{F}_n\mathbf{M}^{-1}\mathscr{F}_n\mathbf{M}^{-1}\mathscr{F}_n),$ $\mathscr{M}_4 = -\frac{1}{60}\ddot{\mathscr{F}}_n - \frac{1}{720}\mathscr{F}_n\mathbf{M}^{-1}\mathscr{F}_n,$ $\mathbf{F}_{a,4} = \frac{7}{240}\dot{\mathscr{F}}_n\mathbf{M}^{-1}\dot{\mathbf{F}}_n + \frac{1}{80}(\mathscr{F}_n\mathbf{M}^{-1}\ddot{\mathbf{F}}_n + \ddot{\mathscr{F}}_n\mathbf{M}^{-1}\mathbf{F}_n) + \frac{1}{120}(\mathbf{F}_n^{(4)} + \mathscr{F}_n\mathbf{M}^{-1}\mathscr{F}_n\mathbf{M}^{-1}\mathbf{F}_n),$ $\mathbf{V}_{a,4} = \frac{1}{80}\mathbf{M}^{-1}\mathbf{F}_n^{(3)} + \frac{1}{120}\mathbf{M}^{-1}\mathscr{F}_n\mathbf{M}^{-1}\dot{\mathbf{F}}_n + \frac{1}{240}\mathbf{M}^{-1}\dot{\mathscr{F}}_n\mathbf{M}^{-1}\mathbf{F}_n$

6. Example momentum-conserving algorithms

We now give a few examples of momentum-conserving algorithms that fit into the general framework provided by Algorithm MC, and discuss their properties.

6.1. Current second-order algorithms

The four momentum-conserving time-integration schemes mentioned briefly in Section 4 all have the form (3.10). To repeat, the symplectic-momentum mid-point algorithm [15,23,24,38] defines

$$\mathscr{X} := \left\langle \frac{\phi'_{ij}(\|\boldsymbol{r}^{ij}_{1/2}\|)}{\|\boldsymbol{r}^{ij}_{1/2}\|}\boldsymbol{I}_3 \right\rangle, \tag{6.1}$$

where $\boldsymbol{r}^{ij} := \boldsymbol{r}^j - \boldsymbol{r}^i$. This algorithm neither conserves energy nor preserves relative equilibria in general. We will refer to this algorithm as SM. The energy–momentum mid-point algorithm [8,11,22,41] has

$$\mathscr{X} := \left\langle \frac{\phi_{ij\Delta}}{\frac{1}{2}(l^2_{ijn+1} - l^2_{ijn})}\boldsymbol{I}_3 \right\rangle, \tag{6.2}$$

for $l_{ij} = \|\boldsymbol{r}^j - \boldsymbol{r}^i\|$. This algorithm conserves energy [8] and also preserves relative equilibria [12,18]. We will refer to this algorithm as EM. The assumed distance method [37,25] is given by

$$\mathscr{X} := \left\langle \frac{\phi'_{ij}(l_{ij1/2})}{l_{ij1/2}} \boldsymbol{I}_3 \right\rangle, \tag{6.3}$$

where $l_{ij1/2} = \frac{1}{2}(l_{ijn} + l_{ijn+1})$. This algorithm is not energy-conserving in general, but coincides with the energy–momentum algorithm for quadratic functions ϕ_{ij} [37,42]: it also preserves relative equilibria [42]. Lastly, the energy-conserving algorithm of [15], which can be constructed by applying a global energy condition to the symplectic-momentum algorithm, has

$$\mathscr{X} := \kappa \left\langle \frac{\phi'_{ij}(\|\boldsymbol{r}^{ij}_{1/2}\|)}{\|\boldsymbol{r}^{ij}_{1/2}\|} \boldsymbol{I}_3 \right\rangle, \tag{6.4}$$

with κ defined by (4.8), which coincides with (4.9). This algorithm does not preserve relative equilibria in general. Each of these algorithms is time-reversible, as can be seen from (4.21). They are also second-order accurate. Note that the second-order accuracy of these algorithms in their given form, with $\boldsymbol{F}_a := \boldsymbol{F}$, tacitly assumes that the force $\boldsymbol{F}$ is constant. In the case of a non-constant force, this definition would need to be altered to $\boldsymbol{F}_a := \boldsymbol{F}_{1/2}$, as can be seen from Table 1.

We now analyse the paths of relative equilibria produced by the energy–momentum (EM) and assumed distance algorithms; in these cases we have, from Lemma 3, $\lambda_{\mathrm{r}} = \frac{1}{2}\theta^{\mathrm{RE}} / \tan\left(\frac{1}{2}\theta^{\mathrm{RE}}\right)$ and $\lambda_{\mathrm{t}} = 1$. Thus these algorithms will produce paths of relative equilibria with

$$\lambda_{\mathrm{r}} = \frac{\tan^{-1}\left(\frac{1}{2}\omega_0 \Delta t\right)}{\frac{1}{2}\omega_0 \Delta t} \quad \text{and} \quad \lambda_{\mathrm{t}} = 1. \tag{6.5}$$

These paths do *not* follow the trajectory of the exact solution in general, as mentioned in Section 4.5. In the case of pure translation, however, we have $\omega_0 = 0$ and thus the exact solution is recovered. For general steady-state motion, the translational part $\boldsymbol{R}^c_k$ is computed exactly, and the overall error in the positions $\boldsymbol{R}_n$ at time-step n is given by

$$\boldsymbol{R}_n - \boldsymbol{R}(t_n) = [\exp(\lambda_{\mathrm{r}} n \Delta t \widehat{\boldsymbol{\Omega}}_0) - \exp(n \Delta t \widehat{\boldsymbol{\Omega}}_0)]\overline{\boldsymbol{R}}_0 \tag{6.6}$$

from $(4.17)_1$. This error is contained entirely in the *rotational part* of the motion and, owing to its being defined as a difference in *positions* between the exact and the numerical solution, is necessarily *bounded* by the radius of the structure (as measured from the centre of mass). In other words, while the error in rotation (and consequently the length of the path traced) will accumulate as $t \to \infty$, the error in position will stay bounded and only change periodically in time. Given that $\tan^{-1} x < x \ \forall x > 0$, $(6.5)_1$ tells us that $\lambda_{\mathrm{r}} < 1$, and thus the rotation of the structure lags behind the exact solution, with the error in rotation given as $(1-\lambda_{\mathrm{r}})n\omega_0\Delta t = 0$ at time-step n.

Note that the local error for λ_{r} defined by (6.5) turns out to be

$$\boldsymbol{R}_{n+1} - \boldsymbol{R}(t_{n+1}) = [\mathrm{cay}(\Delta t \boldsymbol{\Omega}_0) - \exp(\Delta t \widehat{\boldsymbol{\Omega}}_0)]\overline{\boldsymbol{R}}_n,$$

where the Cayley transform $\mathrm{cay}(\boldsymbol{\omega}_0 \Delta t) = \boldsymbol{I} + \frac{1}{1+\frac{1}{4}\|\boldsymbol{\omega}_0 \Delta t\|^2}\left(\Delta t \hat{\boldsymbol{\omega}}_0 + \frac{\Delta t^2}{2}\hat{\boldsymbol{\omega}}_0^2\right)$ [43] in this equation is only a second-order approximation to the exponential mapping $\exp(\hat{\boldsymbol{\omega}}_0 \Delta t) = \boldsymbol{I} + \frac{\sin\|\boldsymbol{\omega}_0\Delta t\|}{\|\boldsymbol{\omega}_0 \Delta t\|}\Delta t \hat{\boldsymbol{\omega}}_0 + \frac{1-\cos\|\boldsymbol{\omega}_0\Delta t\|}{\|\boldsymbol{\omega}_0 \Delta t\|^2}\Delta t^2 \hat{\boldsymbol{\omega}}_0^2$ [36] and is responsible for the position error.

6.2. Higher-order accurate algorithms

To illustrate the theory developed in Sections 4 and 5, we now present a sample fourth-order accurate, momentum-conserving algorithm that is time-reversible and preserves relative equilibria. We will call this algorithm M4; we acknowledge that there are many other possibilities. For fourth-order accuracy we require, from Table 1,

$$\begin{aligned}
\mathscr{G} &= \frac{1}{12}\dot{\mathscr{F}}_n \boldsymbol{M}^{-1}\Delta t^3 + \frac{1}{24}\ddot{\mathscr{F}}_n \boldsymbol{M}^{-1}\Delta t^4 + \mathscr{O}(\Delta t^5),\\
\mathscr{X} &= \mathscr{F}_n + \frac{1}{2}\dot{\mathscr{F}}_n \Delta t + \left(\frac{1}{6}\ddot{\mathscr{F}}_n + \frac{1}{12}\mathscr{F}_n \boldsymbol{M}^{-1}\mathscr{F}_n\right)\Delta t^2\\
&\quad + \frac{1}{24}\left(\mathscr{F}^{(3)}_n + \mathscr{F}_n \boldsymbol{M}^{-1}\dot{\mathscr{F}}_n + \dot{\mathscr{F}}_n \boldsymbol{M}^{-1}\mathscr{F}_n\right)\Delta t^3 + \mathscr{O}(\Delta t^4),\\
\mathscr{M} &= \boldsymbol{M} - \frac{1}{12}\mathscr{F}_n \Delta t^2 - \frac{1}{24}\dot{\mathscr{F}}_n \Delta t^3 + \mathscr{O}(\Delta t^4),\\
\boldsymbol{F}_a &= \boldsymbol{F}_n + \frac{1}{2}\dot{\boldsymbol{F}}_n \Delta t + \left(\frac{1}{6}\ddot{\boldsymbol{F}}_n + \frac{1}{12}\mathscr{F}_n \boldsymbol{M}^{-1}\boldsymbol{F}_n\right)\Delta t^2\\
&\quad + \frac{1}{24}\left(\boldsymbol{F}^{(3)}_n + \mathscr{F}_n \boldsymbol{M}^{-1}\dot{\boldsymbol{F}}_n + \dot{\mathscr{F}}_n \boldsymbol{M}^{-1}\boldsymbol{F}_n\right)\Delta t^3 + \mathscr{O}(\Delta t^4) \quad \text{and}
\end{aligned}$$

$$\boldsymbol{V}_a = \frac{1}{12}\boldsymbol{M}^{-1}\dot{\boldsymbol{F}}_n\Delta t^2 + \frac{1}{24}\boldsymbol{M}^{-1}\ddot{\boldsymbol{F}}_n\Delta t^3 + \mathcal{O}(\Delta t^4) \tag{6.7}$$

We can therefore define

$$\begin{aligned}
&\mathscr{G} := \frac{\Delta t^2}{12}\mathscr{F}_\Delta \boldsymbol{M}^{-1}, \quad \mathscr{M} := \boldsymbol{M}\left(\boldsymbol{M} + \frac{\Delta t^2}{12}\mathscr{F}_{1/2}\right)^{-1}\boldsymbol{M},\\
&\mathscr{X} := \widetilde{\mathscr{X}} + \frac{\Delta t^2}{12}\widetilde{\mathscr{X}}\boldsymbol{M}^{-1}\widetilde{\mathscr{X}}, \quad \text{where } \widetilde{\mathscr{X}} := \mathscr{F}_{1/2} - \frac{\Delta t}{12}\dot{\mathscr{F}}_\Delta,\\
&\boldsymbol{F}_a := \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} \boldsymbol{F}(t)\mathrm{d}t + \frac{\Delta t^2}{12}\mathscr{F}_{1/2}\boldsymbol{M}^{-1}\boldsymbol{F}_{1/2} \quad \text{and} \quad \boldsymbol{V}_a := \frac{\Delta t}{12}\boldsymbol{M}^{-1}\boldsymbol{F}_\Delta
\end{aligned} \tag{6.8}$$

in order to fulfil the respective accuracy criteria, which are verified by taking a Taylor series expansion of each expression. (The definition of $\boldsymbol{F}_a$ obviously presumes the function $\boldsymbol{F}(t)$ to be integrable.) The term $\dot{\mathscr{F}}_\Delta = \dot{\mathscr{F}}_{n+1} - \dot{\mathscr{F}}_n$ in the definition of $\mathscr{X}$ involves dot-product calculations when evaluating the entries $\dot{f}_{ijn}$ and $\dot{f}_{ijn+1}$, which are known to cause numerical difficulties when dealing with stiff problems [32]. Therefore we do not advocate the use of this algorithm in practice.

Regarding the properties of the algorithm defined by (6.8), we see from the structure of $\mathscr{F}$ given in (2.17) that

$$\mathscr{I}_N\mathscr{G} = \frac{\Delta t^2}{12}\mathscr{I}_N\mathscr{F}_\Delta\boldsymbol{M}^{-1} = \frac{\Delta t^2}{12}\mathscr{O}_N\boldsymbol{M}^{-1} = \mathscr{O}_N,$$

and similarly $\mathscr{I}_N\mathscr{X} = \mathscr{O}_N$. Thus $\mathscr{G}$ and $\mathscr{X}$ satisfy the requirements for linear momentum conservation given in (4.2), and for a constant force $\boldsymbol{F}$, we have

$$\boldsymbol{F}_a = \boldsymbol{F} + \frac{\Delta t^2}{12}\mathscr{F}_{1/2}\boldsymbol{M}^{-1}\boldsymbol{F}.$$

Hence $\sum_{i=1}^N \boldsymbol{F}_a^i = \sum_{i=1}^N \boldsymbol{F}^i + \frac{\Delta t^2}{12}\mathscr{I}_N\mathscr{F}_{1/2}\boldsymbol{M}^{-1}\boldsymbol{F} = \boldsymbol{0}$ whenever $\sum_{i=1}^N \boldsymbol{F}^i = \boldsymbol{0}$. Therefore linear momentum is conserved for $\sum_{i=1}^N \boldsymbol{F}^i = \boldsymbol{0}$, as desired. Angular momentum conservation is also assured for $\boldsymbol{F} = \boldsymbol{0}$ due to the symmetry of $\mathscr{M}$ and $\mathscr{X}$. Time reversibility follows from (4.21), since $\mathscr{F}_\Delta^{(s)\,\mathrm{TR}} = -\mathscr{F}_\Delta^{(s)}\ \forall s$ and $\mathscr{F}_{1/2}^{\mathrm{TR}} = \mathscr{F}_{1/2}$.

Under relative equilibrium conditions, we have from (6.8), when $\boldsymbol{F} = \boldsymbol{0}$,

$$\begin{aligned}
&\mathscr{G}^{\mathrm{RE}} = \boldsymbol{0}_{3N}, \quad \mathscr{M}^{\mathrm{RE}} = \boldsymbol{M}\left(\boldsymbol{M} + \frac{\Delta t^2}{12}\mathscr{F}_0\right)^{-1}\boldsymbol{M}, \quad \boldsymbol{F}_a^{\mathrm{RE}} = \boldsymbol{V}_a^{\mathrm{RE}} = \boldsymbol{0} \quad \text{and}\\
&\mathscr{X}^{\mathrm{RE}} = \mathscr{F}_0 + \frac{\Delta t^2}{12}\mathscr{F}_0\boldsymbol{M}^{-1}\mathscr{F}_0.
\end{aligned}$$

When inserted into conditions (4.18), this leads to, after rearranging,

$$\begin{aligned}
&\frac{\Delta t^2}{12}\mathscr{F}_0\boldsymbol{M}^{-1}\mathscr{F}_0\overline{\boldsymbol{R}}_0 = (\alpha-1)\mathscr{F}_0\overline{\boldsymbol{R}}_0, \quad \alpha\boldsymbol{M}\overline{\boldsymbol{V}}_0 = \left(\boldsymbol{M} + \frac{\Delta t^2}{12}\mathscr{F}_0\right)\overline{\boldsymbol{V}}_0\\
&\text{and} \quad \left(\boldsymbol{M} + \frac{\Delta t^2}{12}\mathscr{F}_0\right)\boldsymbol{V}_0^{\mathrm{c}} = \lambda_{\mathrm{t}}\boldsymbol{M}\boldsymbol{V}_0^{\mathrm{c}},\\
&\text{where} \quad \alpha := \lambda_{\mathrm{r}}\frac{\tan\left(\frac{1}{2}\theta^{\mathrm{RE}}\right)}{\frac{1}{2}\theta^{\mathrm{RE}}} \quad \text{and} \quad \theta^{\mathrm{RE}} := \lambda_{\mathrm{r}}\omega_0\Delta t.
\end{aligned} \tag{6.9}$$

Using (2.37) and the fact that $\widehat{\boldsymbol{\Omega}}_0^3 = -\omega_0^2\widehat{\boldsymbol{\Omega}}_0$ from (2.32), it can be shown that

$$(\boldsymbol{M}^{-1}\mathscr{F}_0)^s\overline{\boldsymbol{R}}_0 = -\omega_0^{2(s-1)}\widehat{\boldsymbol{\Omega}}_0^2\overline{\boldsymbol{R}}_0 \quad \text{and} \quad (\boldsymbol{M}^{-1}\mathscr{F}_0)^s\overline{\boldsymbol{V}}_0 = \omega_0^{2s}\overline{\boldsymbol{V}}_0 \qquad \forall s \in \mathbb{Z}^+.$$

It follows that M4 produces paths of relative equilibria with

$$\lambda_{\mathrm{r}} = \frac{\tan^{-1}\left(\frac{1}{2}\alpha\omega_0\Delta t\right)}{\frac{1}{2}\omega_0\Delta t} \ \text{ for } \alpha = 1 + \frac{\omega_0^2\Delta t^2}{12} \quad \text{and} \quad \lambda_{\mathrm{t}} = 1; \tag{6.10}$$

thus the translational part of any relative equilibrium motion is exactly recovered.

6.3. Angle-preserving algorithms

In this section, we extend to multi-element problems the ideas introduced in Section 8.4 of [32] for central-force algorithms designed to eliminate the error in rotation (or *period error*) for steady-state problems. It is possible to remove this error entirely by modifying the definitions of $\mathscr{X}$ and $\mathscr{M}$. For $\mathscr{G}^{\mathrm{RE}} := \boldsymbol{0}_{3N}$, $\boldsymbol{F}_a^{\mathrm{RE}} := \boldsymbol{0}$ and $\boldsymbol{V}_a^{\mathrm{RE}} := \boldsymbol{0}$ we have, from (4.18)

$$\left(\mathscr{X}^{\mathrm{RE}} - \lambda_{\mathrm{r}} \frac{\tan\left(\frac{1}{2}\theta^{\mathrm{RE}}\right)}{\frac{1}{2}\theta^{\mathrm{RE}}} \mathscr{F}_0\right)\overline{\boldsymbol{R}}_0 = \boldsymbol{0}, \quad \left(\lambda_{\mathrm{r}} \frac{\tan\left(\frac{1}{2}\theta^{\mathrm{RE}}\right)}{\frac{1}{2}\theta^{\mathrm{RE}}} \mathscr{M}^{\mathrm{RE}} - \boldsymbol{M}\right)\overline{\boldsymbol{V}}_0 = \boldsymbol{0} \quad \text{and} \quad (\boldsymbol{M} - \lambda_{\mathrm{t}}\mathscr{M}^{\mathrm{RE}})\boldsymbol{V}_0^{\mathrm{c}} = \boldsymbol{0}. \tag{6.11}$$

To remove the error in rotation, we need to ensure that $\lambda_{\mathrm{r}} = 1$, which leads to

$$\mathscr{X}^{\mathrm{RE}}\overline{\boldsymbol{R}}_0 = \beta^{\mathrm{RE}}\mathscr{F}_0\overline{\boldsymbol{R}}_0 \quad \text{and} \quad \mathscr{M}^{\mathrm{RE}}\overline{\boldsymbol{V}}_0 = \frac{1}{\beta^{\mathrm{RE}}}\boldsymbol{M}\overline{\boldsymbol{V}}_0 \qquad \text{for } \beta^{\mathrm{RE}} = \frac{\tan\left(\frac{1}{2}\omega_0\Delta t\right)}{\frac{1}{2}\omega_0\Delta t}. \tag{6.12}$$

Thus algorithms that have $\mathscr{X}^{\mathrm{RE}}\overline{\boldsymbol{R}}_0 = \mathscr{F}_0\overline{\boldsymbol{R}}_0$ and $\mathscr{M}^{\mathrm{RE}}\overline{\boldsymbol{V}}_0 = \boldsymbol{M}\overline{\boldsymbol{V}}_0$, such as EM and the assumed distance method, can be modified via $\mathscr{X} \mapsto \beta\mathscr{X}$, $\mathscr{M} \mapsto \frac{1}{\beta}\mathscr{M}$ to give (6.12), where β is defined as

$$\beta := \frac{\tan\left(\frac{1}{2}\theta\right)}{\frac{1}{2}\theta}, \quad \text{with } \theta \text{ defined so that } \theta^{\mathrm{RE}} = \omega_0\Delta t. \tag{6.13}$$

(Note that (6.13) is equivalent to $(8.3)_2$ in [32].) These algorithms will return the exact angle of rotation for general relative equilibrium problems.

We now define the *angle-preserving scheme* EMθ, arising from EM via the modification

$$\mathscr{G} := \boldsymbol{0}_{3N}, \quad \mathscr{X} := \beta\langle\xi_{ij}\boldsymbol{I}_3\rangle, \quad \mathscr{M} := \frac{1}{\beta}\boldsymbol{M}, \quad \boldsymbol{F}_a := \alpha\boldsymbol{F} \quad \text{and} \quad \boldsymbol{V}_a := \boldsymbol{0}, \tag{6.14}$$

with β defined by $(6.13)_1$ and ξ_{ij} by (4.14); the definition of θ will follow shortly. Note that EMθ retains the property of elemental energy conservation, being of the form (4.16). From Lemma 3, we have

$$\lambda_{\mathrm{r}} = 1 \quad \text{and} \quad \lambda_{\mathrm{t}} = \beta^{\mathrm{RE}} = \frac{\tan\left(\frac{1}{2}\omega_0\Delta t\right)}{\frac{1}{2}\omega_0\Delta t}, \tag{6.15}$$

which may be compared with (6.5). Since $\tan x > x \ \forall 0 < x < \frac{\pi}{2}$, the translational movement with this scheme is *faster* than the exact solution. The position errors are given by

$$\boldsymbol{R}_n - \boldsymbol{R}(t_n) = (\lambda_{\mathrm{t}} - 1)n\Delta t\boldsymbol{V}_0^{\mathrm{c}}, \tag{6.16}$$

as seen from (4.17). This error is not bounded in time; thus for a general relative equilibrium problem, EMθ will accumulate (translational) errors in positions as $t \to \infty$. This is in contrast to schemes given by (3.10), where the position error is bounded in time, but the rotation error accumulates as $t \to \infty$. From (4.17), however, it can be seen that the error in momenta $\boldsymbol{P}_n$ is zero.

We desire an algorithm that calculates both translational and rotational components of the positions (and also the momenta) exactly for general steady-state problems: such algorithms are possible, as mentioned in Section 4.4, although they are unlikely to conserve energy (for they can do so neither globally nor elementally, as seen from Lemmas 2 and 3). By way of an example, we define the angle-preserving algorithm Aθ by

$$\mathscr{G} := \boldsymbol{0}_{3N}, \quad \mathscr{X} := \beta\mathscr{F}_{1/2}, \quad \mathscr{M} := \boldsymbol{M} + c\Delta t^2\mathscr{F}_{1/2}, \quad \boldsymbol{F}_a := \boldsymbol{F} \quad \text{and} \quad \boldsymbol{V}_a := \boldsymbol{0} \tag{6.17}$$

for some $c(\theta)$ to be determined, with β and θ defined by (6.13). Under relative equilibrium conditions, this algorithm immediately satisfies conditions $(4.18)_{3,5}$, and $(4.18)_{1,4}$ are satisfied by $\lambda_{\mathrm{r}} = \lambda_{\mathrm{t}} = 1$ given that $\mathscr{F}_{1/2}^{\mathrm{RE}} = \mathscr{F}_0$, and $\mathscr{F}_0\boldsymbol{V}_0^{\mathrm{c}} = \boldsymbol{0}$ due to the form of $\mathscr{F}_0$. Eq. $(4.18)_2$ then becomes

$$(\lambda_{\mathrm{r}}\beta^{\mathrm{RE}} - 1)\boldsymbol{M}\overline{\boldsymbol{V}}_0 + \lambda_{\mathrm{r}}\beta^{\mathrm{RE}}c^{\mathrm{RE}}\Delta t^2\mathscr{F}_0\overline{\boldsymbol{V}}_0 = \boldsymbol{0},$$

which simplifies to

$$[\lambda_{\mathrm{r}}\beta^{\mathrm{RE}}(1 + c^{\mathrm{RE}}\omega_0^2\Delta t^2) - 1]\boldsymbol{M}\overline{\boldsymbol{V}}_0 = \boldsymbol{0}.$$

(Note from (2.32) that $\widehat{\boldsymbol{\Omega}}_0^3 = -\omega_0^2\widehat{\boldsymbol{\Omega}}_0$, and that $\mathscr{F}_0\overline{\boldsymbol{R}}_0 = -\widehat{\boldsymbol{\Omega}}_0\boldsymbol{M}\overline{\boldsymbol{V}}_0$ and $\overline{\boldsymbol{V}}_0 = \widehat{\boldsymbol{\Omega}}_0\overline{\boldsymbol{R}}_0$ from (2.37).) This can be solved for c^{RE} to give

$$c^{\mathrm{RE}} = \frac{1 - \lambda_{\mathrm{r}}\beta^{\mathrm{RE}}}{\lambda_{\mathrm{r}}\beta^{\mathrm{RE}}\omega_0^2\Delta t^2}. \tag{6.18}$$

So for c defined as

$$c := \frac{\frac{1}{2}\theta - \tan\left(\frac{1}{2}\theta\right)}{\theta^2\tan\left(\frac{1}{2}\theta\right)},$$

(6.18) is satisfied for $\lambda_{\mathrm{r}} = 1$; hence $(4.18)_2$ is also. Note that Aθ does not conserve energy in general, although it does produce exact solutions to all steady-state problems: a fact that could have relevance when designing algorithms for solving stiff, approximately steady-state problems at large time-steps.

We now turn to the requirement on θ that $\theta^{\mathrm{RE}} = \omega_0 \Delta t$. There are several plausible definitions of θ that satisfy this. The simplest involves taking the average of all of the angles moved through by the vectors $\bar{\boldsymbol{r}}^i, i = 1, \ldots, N$ (representing the position of each node relative to the centre of mass) during one time-step. This does not take into account the actual displacement of each node, however, so a better choice would be a weighted average of the angles θ_i that takes into account the associated lengths $\bar{l}^i_n$ and $\bar{l}^i_{n+1}$, namely

$$\theta := \frac{\sum_{i=1}^{N} \bar{l}^i_{1/2} \theta_i}{\sum_{i=1}^{N} \bar{l}^i_{1/2}} \quad \text{for } \theta_i = \cos^{-1}\left(\frac{\bar{\boldsymbol{r}}^i_n \cdot \bar{\boldsymbol{r}}^i_{n+1}}{\bar{l}^i_{n+1} \bar{l}^i_n}\right) \text{ and } \bar{l}^i := \|\bar{\boldsymbol{r}}^i\|. \tag{6.19}$$

For a relative equilibrium problem, we have $\theta_i = \omega_0 \Delta t \ \forall i$ from (4.17), and thus $\theta^{\mathrm{RE}} = \omega_0 \Delta t$.

In summary, both algorithms EMθ and Aθ are designed to eliminate the rotational error in a general steady-state problem. Linear and angular momenta are conserved by these schemes, since the form of $\mathscr{X}$ and $\mathscr{M}$ are unchanged. Energy is conserved by EMθ but not by Aθ. Each is time-reversible, since $\theta^{\mathrm{TR}} = \theta$, and also second-order accurate, since

$$\frac{\tan\left(\frac{1}{2}\theta\right)}{\frac{1}{2}\theta} = 1 + \frac{1}{12}\theta^2 + \mathscr{O}(\theta^4)$$

with $\theta \in \mathscr{O}(\Delta t)$.

7. Numerical results

We will now test the new algorithms described in the previous chapter on a couple of model problems, to verify the properties discussed in Sections 4 and 5. We will run each experiment with a range of time-step sizes, and assess the relative errors in the positions $\boldsymbol{R}$ and velocities $\boldsymbol{V}$ at two different sampling times, in order to verify the results. These errors are calculated as

$$\frac{\|\boldsymbol{R}_n - \boldsymbol{R}(t_n)\|}{\|\boldsymbol{R}(t_n)\|} \quad \text{and} \quad \frac{\|\boldsymbol{V}_n - \boldsymbol{V}(t_n)\|}{\|\boldsymbol{V}(t_n)\|}$$

respectively, where $\{\boldsymbol{R}_n, \boldsymbol{V}_n\}$ denotes the approximate solution and $\{\boldsymbol{R}(t_n), \boldsymbol{V}(t_n)\}$ the reference solution. The reference solution for each problem was obtained by running the established energy-conserving scheme EM at a suitably low time-step size, as will be fully described in each case. In order to minimise the effects of round-off error, quadruple precision arithmetic was used when generating the reference solutions, and in some cases for the tests themselves, as will be mentioned.

7.1. Spring-mass system

Our first pair of examples is based on the spring-mass system described in Section 5.1 of [15], shown on the left-hand side of Fig. 1. This consists of four separate masses, with each pair of masses connected by a spring. The strain energy function for each pair is

$$\phi_{ij}(l_{ij}) := \frac{1}{2} k_{ij} (l_{ij} - \bar{l}_{ij})^2, \tag{7.1}$$

where k_{ij} and $\bar{l}_{ij}$ represent the spring stiffness and undeformed distance between masses i and j, respectively. This strain energy function gives rise to the *Engineering strain* measure (e.g. [44, Section 3.1.1]). The model parameters are $k_{ij} = 1$,

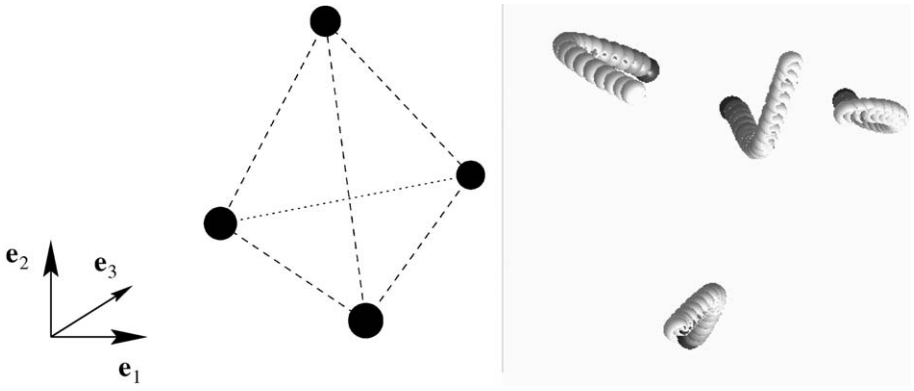

Fig. 1. Spring-mass system at rest and in motion.

$m^i = 1$ and $\bar{l}_{ij} = 1\ \forall i, j$ from [15], thus each spring has a natural length of unity. We note that the problem is *non-stiff*. The initial position and momentum vectors $\boldsymbol{r}_0^i$ and $\boldsymbol{p}_0^i$ were randomly generated by the authors as

$$\boldsymbol{r}_0^1 = \langle 0.2340, -0.2166, -0.0109\rangle, \quad \boldsymbol{p}_0^1 = \langle 0.04095, -0.01483, 0.04325\rangle,$$
$$\boldsymbol{r}_0^2 = \langle 0.0772, 0.7605, 0.0061\rangle, \quad \boldsymbol{p}_0^2 = \langle -0.02980, 0.04400, -0.02959\rangle,$$
$$\boldsymbol{r}_0^3 = \langle 0.8054, 0.6466, -0.1059\rangle, \quad \boldsymbol{p}_0^3 = \langle -0.02328, -0.01432, -0.03716\rangle,$$
$$\boldsymbol{r}_0^4 = \langle 0.3903, 0.6187, 0.9678\rangle \quad \text{and} \quad \boldsymbol{p}_0^4 = \langle 0.04152, 0.00114, 0.02621\rangle.$$

We will analyse two versions of this problem: a force-free case (giving a fully conservative system, as used in [15]), and an asymptotically conservative case with a decaying external force $\boldsymbol{F}$: note that this second version involves a *time-dependent* force. The total response time in each case is 30 s, and we measure the relative errors in the solutions obtained at $t = 10$ and $t = 30$ s accordingly. The reference solutions for all versions were run using algorithm EM with a time-step size of $\Delta t = 10^{-5}$ and convergence tolerance of 10^{-18} for the Newton–Raphson iteration. The image on the right-hand side of Fig. 1 depicts the trajectories of the masses during the initial seconds of the motion in the force-free case, as given by the reference solution, with the darkest colour denoting the initial configuration.

This problem is used to show the theoretical properties of the fourth-order accurate algorithm M4: we will therefore test this algorithm alongside the established algorithms SM and EM, for comparison. (Given the nature of the strain energy function in (7.1), the scheme EM and the assumed distance method are equivalent, as was proved in [42].) Each algorithm will be run using the following time-step sizes: $\Delta t = 1$,0.25, 0.0625 and 0.015625. All tests will be carried out using double precision arithmetic, with a convergence tolerance of 10^{-10}.

7.1.1. Force-free case

Our first version of the spring-mass system has no external force, and is thus identical to that in [15] (and subsequently [42]). Note that the total response time was 50 s in [15]: in all other respects, our results may be compared to those given there. Both energy and momenta are constant, as given in Fig. 2; the small values of the momenta reflect the fact that this problem is dominated by axial vibration. The amount of rotational and translational movement is tiny in comparison, so the group motions are very small. From Fig. 8 of [15] we can estimate the period of axial vibration for each pair of masses to be approximately 5 s: thus all of the time-steps used are small enough to resolve this mode approximately.

Fig. 3 shows the corresponding energy and momenta given by algorithm M4 at the largest time-step size. We can see that, in accordance with the properties listed in Section 6, this scheme conserves linear and angular momenta. (The established algorithms SM and EM also conserve momenta, and EM additionally conserves energy.)

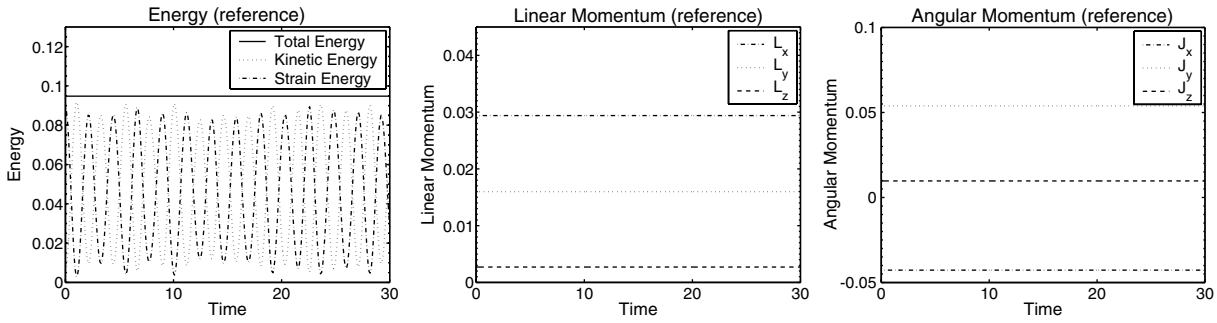

Fig. 2. Reference data for the force-free spring-mass system.

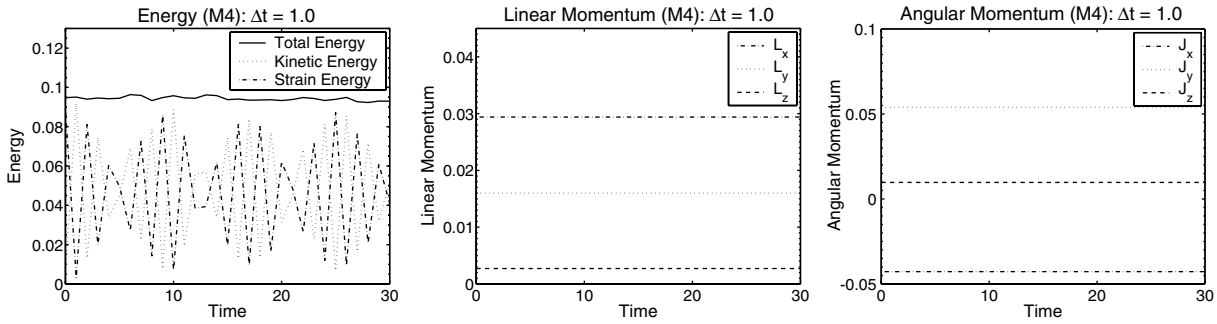

Fig. 3. Energy and momenta as given by M4.

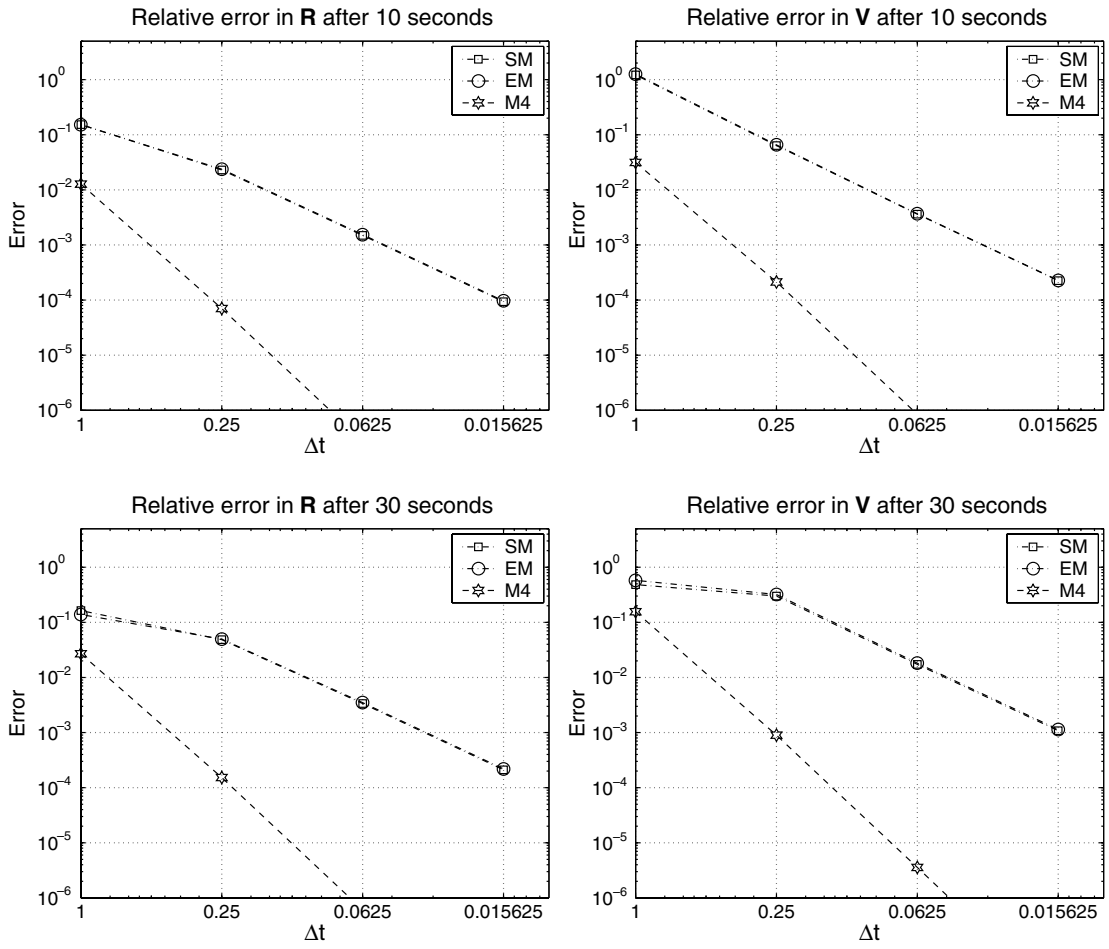

Fig. 4. Relative errors in $\boldsymbol{R}$ and $\boldsymbol{V}$ for the force-free spring-mass system.

From Fig. 4; we see that the respective second- and fourth-order accuracy characteristics of these three algorithms are borne out by the error graphs. In particular, the scheme M4 has now demonstrated fourth-order accuracy as well as conservation of momenta. The actual errors in the positions at $\Delta t = 1$ are acceptably small, being roughly 12–15% for the second-order schemes and 2% for M4. The velocities are not accurately represented by the second-order schemes at this time-step size, however, with errors of over 100% incurred.

7.1.2. Decaying force

For this version of the problem, we add an arbitrary constant force $\boldsymbol{F}^i$ (acting on mass i) for each of the four masses, multiplied by a damping factor in each coordinate direction. The constant components are

$$\widehat{\boldsymbol{F}}^1 = \langle 0.0025, 0.01, 0.005\rangle, \quad \widehat{\boldsymbol{F}}^2 = \langle 0.005, 0.0075, 0.005\rangle,$$
$$\widehat{\boldsymbol{F}}^3 = \langle 0.0075, 0.005, 0.005\rangle \quad \text{and} \quad \widehat{\boldsymbol{F}}^4 = \langle 0.01, 0.0025, 0.005\rangle,$$

and the actual forces $\boldsymbol{F}^i \equiv \langle F^i_x, F^i_y, F^i_z\rangle$ are

$$\boldsymbol{F}^i := \langle \mathrm{e}^{-t/5}\widehat{F}^i_x, \mathrm{e}^{-2t/5}\widehat{F}^i_y, \mathrm{e}^{-3t/5}\widehat{F}^i_z\rangle$$

for each mass i, producing a different effect in each of the three directions. All the other data remain the same as for the force-free case. Therefore the system becomes conservative in the limit $t \to \infty$, with energy and momenta tending to constant values as shown in Fig. 5, where P.A.L. stands for the potential of the applied loads.

Fig. 6 shows the corresponding energy and momenta given by M4. Indeed, the values of the momentum components calculated at the end of the analysis, using the coarse time-step size $\Delta t = 1$, are remarkably close to the reference values. The differences in linear momentum are roughly 3% for each of the second-order algorithms; this similarity can be anticipated given that the velocity update for each algorithm is the same. The fourth-order scheme M4 gives the same linear

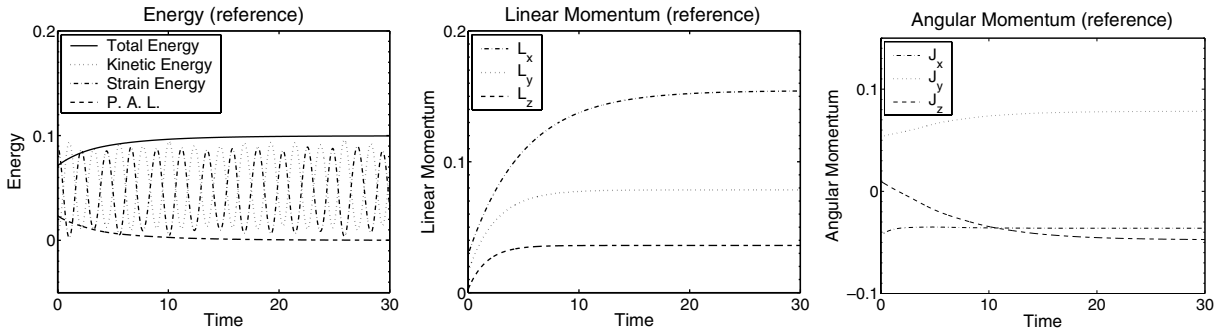

Fig. 5. Reference data for the spring-mass system with a decaying force.

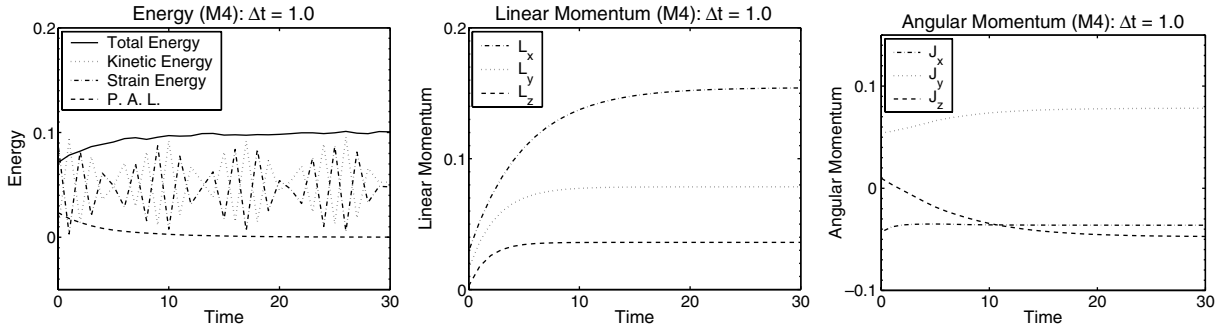

Fig. 6. Energy and momenta as given by M4.

momentum data as the reference solution to nine significant digits. The differences in angular momentum vary, at approximately 1% for EM and 0.06% for M4.

Fig. 7 shows that the relative errors in positions and velocities for the problem with a time-dependent decaying force are actually very similar to those in the force-free case shown in Fig. 4, and thus all observations made previously also apply here. Thus the time-dependence of the forcing term had no effect on accuracy here, and *fourth-order accuracy can be observed for M4 in the case of a non-constant force*. The other algorithms exhibit second-order accuracy, using the midpoint definition $\boldsymbol{F}_a := \boldsymbol{F}_{1/2}$ for the algebraic force.

7.2. Truss structure in relative equilibrium

This example concerns the motion of a truss structure under relative equilibrium conditions, as shown in Fig. 8. It consists of eight bar elements that connect each of the five nodes at (0, 1, 0), (1, 0, 0), (0, −1, 0), (−1, 0, 0) and (0, 0, 0) to one another. The structure is therefore two-dimensional, existing entirely within the plane $z = 0$. The natural length of the elements is consequently $\bar{l}_{ij} = 1$ for those aligned with the coordinate axes, and $\bar{l}_{ij} = \sqrt{2}$ for the rest. The mass per unit length of each bar is equal to unity, hence the element masses are equal in magnitude to their natural lengths. Each element has the strain energy function

$$\phi_{ij}(l_{ij}) := \frac{1}{2} k_{ij} \left(\frac{l_{ij}^2 - \bar{l}_{ij}^2}{2\bar{l}_{ij}} \right)^2, \tag{7.2}$$

which gives rise to *Green's strain* (e.g. [44, Section 3.1.2]). Young's modulus of the material is such that $E_{ij}A_{ij} = 10^2$ throughout the structure, and so the stiffness of the elements is $k_{ij} \equiv \frac{E_{ij}A_{ij}}{\bar{l}_{ij}} = 10^2$ for those along an axis and $k_{ij} = \frac{10^2}{\sqrt{2}}$ for the others.

The initial position and momentum vectors for relative equilibria are obtained from (2.37). In this example, the centre of mass has initial position $\boldsymbol{r}^{\mathrm{c}}{}_0 = \boldsymbol{0}$, and we set its constant velocity to be $\boldsymbol{v}^{\mathrm{c}}{}_0 = \langle 0, 0, 0.75 \rangle$, which is orthogonal to the plane of the structure, and thus aligned with an axis of inertia. The constant angular velocity is $\boldsymbol{\omega}_0 = \langle 0, 0, -1 \rangle$, thereby giving $\boldsymbol{\omega} \times \boldsymbol{v}^{\mathrm{c}} = \boldsymbol{0}$ as specified in Proposition 2.1. Therefore the initial vectors $\boldsymbol{r}_0^i$ and $\boldsymbol{v}_0^i$ are calculated as

$$\boldsymbol{r}_0^1 = \langle 0, 0, 0 \rangle, \quad \boldsymbol{v}_0^1 = \langle 0, 0, 0.75 \rangle,$$
$$\boldsymbol{r}_0^2 = \langle 0, 1.0052720575, 0 \rangle, \quad \boldsymbol{v}_0^2 = \langle 1.0052720575, 0, 0.75 \rangle,$$

$\boldsymbol{r}_0^3 = \langle -1.0052720575, 0, 0\rangle, \quad \boldsymbol{v}_0^3 = \langle 0, 1.0052720575, 0.75\rangle,$
$\boldsymbol{r}_0^4 = \langle 0, -1.0052720575, 0\rangle, \quad \boldsymbol{v}_0^4 = \langle -1.0052720575, 0, 0.75\rangle,$
$\boldsymbol{r}_0^5 = \langle 1.0052720575, 0, 0\rangle \quad \text{and} \quad \boldsymbol{v}_0^5 = \langle 0, -1.0052720575, 0.75\rangle$

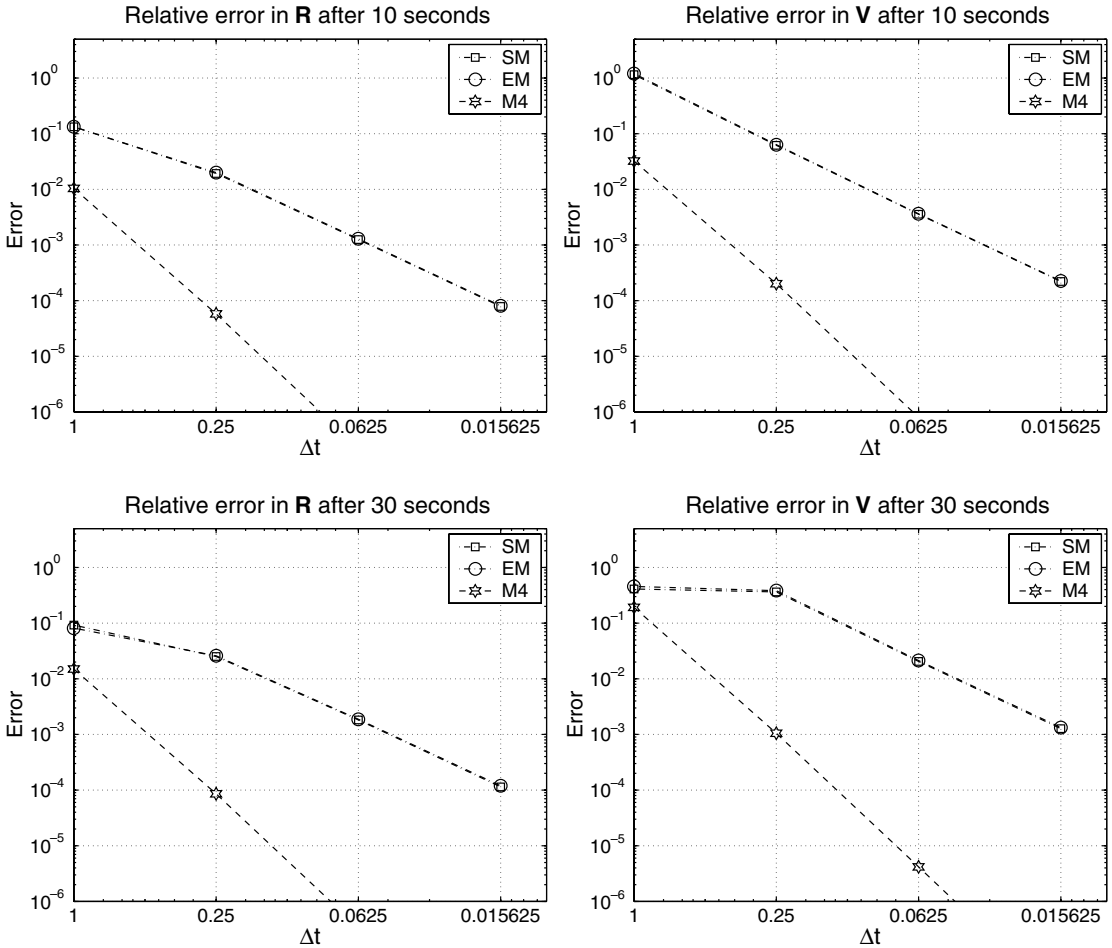

Fig. 7. Relative errors in $\boldsymbol{R}$ and $\boldsymbol{V}$ for the spring-mass system with a decaying force.

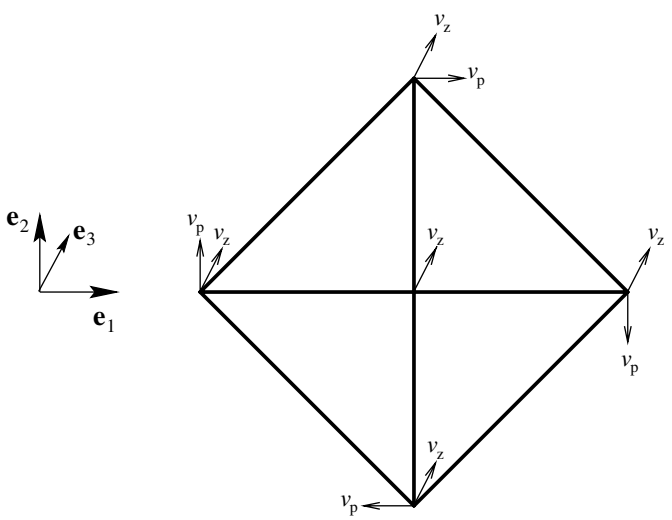

Fig. 8. Truss structure in its initial configuration under relative equilibrium conditions.

to 11 significant digits, with $\boldsymbol{P}_0 = \boldsymbol{M}\boldsymbol{V}_0$. This amounts to the structure being stretched by a distance 0.0052720575 along both of its coordinate axes (with unit vectors $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ in the diagram). In the absence of an external force, these starting conditions imply that the structure will undergo uniform in-plane rotation, with simultaneous uniform translation in the direction of $\boldsymbol{e}_3$. The image in Fig. 9 depicts this motion during the first few seconds, as given by the reference solution, with the darkest colour denoting the initial configuration.

We will analyse this problem for a total response time of 9 s, and measure the relative errors in the solutions obtained at $t = 3$ and $t = 9$ s. The reference solution was run using EM with a time-step size of $\Delta t = 10^{-6}$ and Newton–Raphson convergence tolerance of 10^{-15}. We note that this system is fully conservative. In addition, the relative equilibrium starting conditions mean that the *individual components of energy remain constant* for this problem, as does the length of each element. This fact is illustrated in Fig. 10, which shows the total energy and the distance $\|\boldsymbol{r}^2(t) - \boldsymbol{r}^1(t)\|$ as given by the reference solution. Note that the kinetic energy comprises almost all of the total energy, and is thus hidden in the graph.

For the strain energy function given in (7.2), the energy–momentum and assumed distance method of Section 6 are distinct in general. For steady-state problems, however, they coincide [42]. The purpose of this example is to show the accuracy characteristics of the angle-preserving algorithms EMθ and Aθ, so we will run tests with SM and EM for comparison. The time-step sizes remain as given in Section 7.1, and the experiments are conducted using double precision arithmetic, with a convergence tolerance of 10^{-10}.

Fig. 11 shows the relative errors in positions and velocities for each of the algorithms tested. The errors are significantly larger for SM than the other second-order schemes, reflecting the fact that this algorithm does not preserve relative equilibria. A surprising aspect of these results is the similarity between the position errors from EM and EMθ. This is to a large

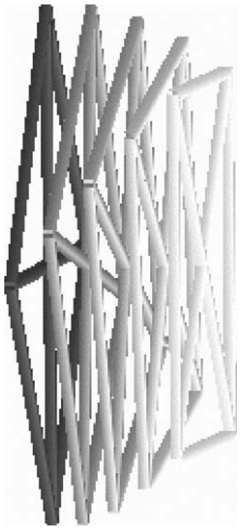

Fig. 9. Truss structure in motion under relative equilibrium conditions.

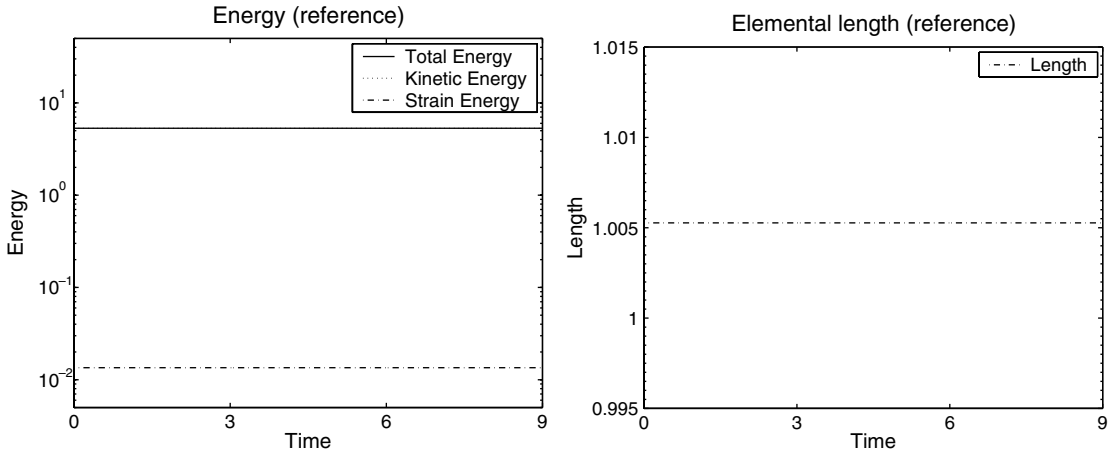

Fig. 10. Reference data for the truss structure in relative equilibrium.

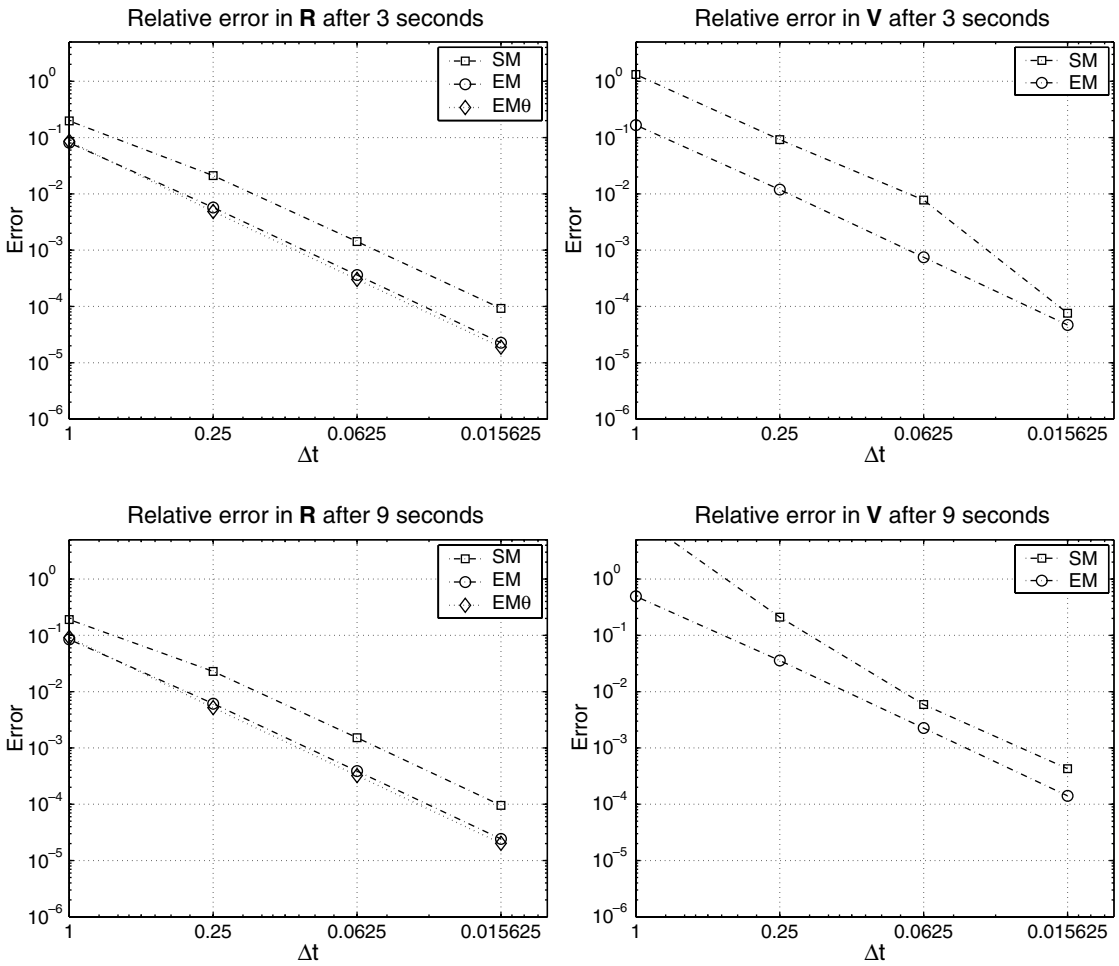

Fig. 11. Relative errors in $\boldsymbol{R}$ and $\boldsymbol{V}$ for the truss structure in relative equilibrium.

extent coincidental, given that the errors are from different sources, although representative of the fact that both modes of motion will be present in a general problem.

Table 2 shows that the overall position error for EMθ consists entirely of translational error: the angle of rotation is reproduced exactly, as expected. Conversely, SM and EM incur some error in the rotational mode, but maintain the exact translation of the structure. We see that both sets of errors (where non-zero) increase linearly with the sampling time, thus confirming that the translational and angular velocities returned by each algorithm are constant. Note that the non-zero period errors indicate that the algorithmic solutions from SM and EM lag *behind* the true solution, whereas the non-zero translational errors show that the solution obtained from EMθ is *ahead* of it.

We also note the absence of any results for EMθ in the velocity graphs, as this quantity is calculated exactly (to within machine precision) by this algorithm, as explained in Section 6.3; hence the errors are zero. On the other hand, the error in positions with EMθ depends on both $\boldsymbol{v}^{c}{}_{0}$ and $\boldsymbol{\omega}_0$, and is unbounded, whereas for EM it depends on $\boldsymbol{\omega}_0$ only, from $(6.5)_1$ and (6.6), and stays bounded in time as a consequence of the fact that it is entirely caused by the (unbounded) rotational error. Only algorithm Aθ reproduces both aspects of the motion exactly.

Table 2
Period and translation errors in positions at $\Delta t = 0.25$ for the truss structure in relative equilibrium

Algorithm	Period error Sampling time (s)			Translation error Sampling time (s)		
	3	6	9	3	6	9
SM	0.0567	0.114	0.172	0	0	0
EM	0.0155	0.031	0.0464	0	0	0
EMθ	0	0	0	0.0118	0.0236	0.0354
Aθ	0	0	0	0	0	0

8. Conclusions

In this work, the design of conservative algorithms with higher-order accuracy was investigated for multi-element truss structures (or, equivalently, multi-particle dynamics), and conditions under which they can be developed were given that follow on from our earlier work on the central-force problem [32]. A general framework for algorithms that conserve linear and angular momenta was presented, and conditions for energy conservation were also given. Criteria for the preservation of relative equilibrium states and also for higher-order accuracy were elaborated in detail, and the exact solution to a general non-linear problem was derived (in power-series form).

Time-reversible algorithms that conserve linear and angular momenta can be designed to have arbitrarily high orders of accuracy; these algorithms retain their accuracy properties in the presence of non-conservative external forces. This can be achieved without recourse to extra stages or calculation or additional degrees of freedom in a manner proposed by Argyris et al. [1,29] and LaBudde and Greenspan [30,31], although the resulting systems of equations will no longer be as sparse for higher-order schemes.

Momentum-conserving algorithms in this framework can be designed to conserve energy or preserve relative equilibrium states, although they are unlikely to do both unless they meet the far more restrictive criteria for *elemental* energy conservation, satisfied only by a small group of algorithms based on the second-order energy–momentum algorithm of Simo et al. [8]. However, it is possible to design algorithms that capture relative equilibrium solutions exactly and, in future, we intend to explore the numerical properties of algorithms that can reproduce exact relative equilibrium solutions, in regard to achieving numerical stability and non-linear iterative convergence when solving stiff problems at large time-steps.

Acknowledgement

This work has been financially supported by the Engineering and Physical Sciences Research Council of Great Britain under a Case Studentship and the Advanced Research Fellowship AF/100089.

Appendix A. Proofs of algorithmic properties

A.1. Proposition 1 (linear momentum conservation)

From Algorithm MC, we have

$$\mathscr{L}_\Delta = \mathscr{I}_N \boldsymbol{P}_\Delta = \mathscr{I}_N(\Delta t \boldsymbol{F}_a - \Delta t \mathscr{X} \boldsymbol{R}_{1/2} - \mathscr{G} \boldsymbol{P}_{1/2}).$$

Since $\boldsymbol{R}_{1/2}$ and $\boldsymbol{P}_{1/2}$ are arbitrary, we have $\frac{1}{\Delta t}\mathscr{L}_\Delta = \mathscr{I}_N \boldsymbol{F}_a$ in general if and only if

$$\mathscr{I}_N \mathscr{X} = \mathscr{I}_N \mathscr{G} = \mathscr{O}_N.$$

Therefore we must have $\sum_{i=1}^N \mathscr{G}^{ij} = \sum_{i=1}^N \mathscr{X}^{ij} = \boldsymbol{0}_3 \ \forall 1 \leqslant j \leqslant N$, as given in (4.2). □

A.2. Proposition 2 (angular momentum conservation)

We first note that, for any vectors $\boldsymbol{r}, \boldsymbol{p} \in \mathbb{R}^3$, we have the identity

$$\boldsymbol{r}_{n+1} \times \boldsymbol{p}_{n+1} - \boldsymbol{r}_n \times \boldsymbol{p}_n = \boldsymbol{r}_\Delta \times \boldsymbol{p}_{1/2} + \boldsymbol{r}_{1/2} \times \boldsymbol{p}_\Delta,$$

which can be verified directly. We also note that since $\mathscr{M}$ comprises unit submatrices, the same is true for $\mathscr{M}^{-1}$, i.e.

$$(\mathscr{M}^{-1})^{ij} = \tilde{\mu}^{ij} \boldsymbol{I}_3.$$

From Algorithm MC and (3.9), we have

$$\begin{aligned}
\mathscr{J}_\Delta &= \sum_{i=1}^N \left(\boldsymbol{r}^i_\Delta \times \boldsymbol{p}^i_{1/2} + \boldsymbol{r}^i_{1/2} \times \boldsymbol{p}^i_\Delta \right) \\
&= \sum_{i=1}^N \left(\left[\sum_{j=1}^N \left(\Delta t \tilde{\mu}^{ij} \boldsymbol{p}^j_{1/2} + g^{ji} \boldsymbol{r}^j_{1/2} \right) - \Delta t \boldsymbol{v}^i_a \right] \times \boldsymbol{p}^i_{1/2} - \boldsymbol{r}^i_{1/2} \times \left[\sum_{j=1}^N \left(\Delta t x^{ij} \boldsymbol{r}^j_{1/2} + g^{ij} \boldsymbol{p}^j_{1/2} \right) - \Delta t \boldsymbol{F}^i_a \right] \right) \\
&= \sum_{i=1}^N \left[\sum_{j=1}^N \left(\Delta t \tilde{\mu}^{ij} \boldsymbol{p}^j_{1/2} \times \boldsymbol{p}^i_{1/2} + g^{ji} \boldsymbol{r}^j_{1/2} \times \boldsymbol{p}^i_{1/2} - g^{ij} \boldsymbol{r}^i_{1/2} \times \boldsymbol{p}^j_{1/2} - \Delta t x^{ij} \boldsymbol{r}^i_{1/2} \times \boldsymbol{r}^j_{1/2} \right) + \Delta t \left(\boldsymbol{r}^i_{1/2} \times \boldsymbol{F}^i_a + \boldsymbol{p}^i_{1/2} \times \boldsymbol{v}^i_a \right) \right] \\
&= \sum_{i=1}^N \left[\sum_{j=1}^N \left(\Delta t \tilde{\mu}^{ij} \boldsymbol{p}^j_{1/2} \times \boldsymbol{p}^i_{1/2} - \Delta t x^{ij} \boldsymbol{r}^i_{1/2} \times \boldsymbol{r}^j_{1/2} \right) + \Delta t \left(\boldsymbol{r}^i_{1/2} \times \boldsymbol{F}^i_a + \boldsymbol{p}^i_{1/2} \times \boldsymbol{v}^i_a \right) \right],
\end{aligned}$$

where the vectors $\boldsymbol{p}^j_{1/2} \times \boldsymbol{p}^i_{1/2}$ and $\boldsymbol{r}^i_{1/2} \times \boldsymbol{r}^j_{1/2}$ are unrelated for all i and j. Thus we have

$$\frac{1}{\Delta t}\mathscr{J}_\Delta = \sum_{i=1}^{N}\left(\boldsymbol{r}^i_{1/2} \times \boldsymbol{F}^i_a + \boldsymbol{p}^i_{1/2} \times \boldsymbol{v}^i_a\right)$$

in general if and only if $\tilde{\mu}^{ij} = \tilde{\mu}^{ji}, x^{ij} = x^{ji}$. Thus $\mathscr{M}^{-1}$ and $\mathscr{X}$ are symmetric, and since

$$\mathscr{M}^{-1}\mathscr{M} = \boldsymbol{I}_{3N} = (\mathscr{M}^{-1})^{\mathrm{T}}\mathscr{M}^{\mathrm{T}},$$

we see that $\mathscr{M} = \mathscr{M}^{\mathrm{T}} \iff \mathscr{M}^{-1} = (\mathscr{M}^{-1})^{\mathrm{T}}$; hence $\mathscr{M}$ itself must be symmetric. □

A.3. Proposition 3 (energy conservation)

We first note the identity

$$\frac{1}{2}\boldsymbol{P}_{n+1}\cdot\boldsymbol{M}^{-1}\boldsymbol{P}_{n+1} - \frac{1}{2}\boldsymbol{P}_n\cdot\boldsymbol{M}^{-1}\boldsymbol{P}_n = \boldsymbol{P}_\Delta\cdot\boldsymbol{M}^{-1}\boldsymbol{P}_{1/2},$$

which can also be easily verified. Algorithm MC then gives

$$H_\Delta = \boldsymbol{P}_\Delta\cdot\boldsymbol{M}^{-1}\boldsymbol{P}_{1/2} + \phi_\Delta - U_\Delta = (\Delta t\boldsymbol{F}_a - \Delta t\mathscr{X}\boldsymbol{R}_{1/2} - \mathscr{G}\boldsymbol{P}_{1/2})\cdot\boldsymbol{M}^{-1}\boldsymbol{P}_{1/2} + \phi_\Delta - U_\Delta, \tag{A.1}$$

from which it can be seen that

$$H_\Delta = -U^{\mathrm{NC}}_\Delta \iff (\Delta t\boldsymbol{F}_a - \mathscr{G}\boldsymbol{P}_{1/2})\cdot\boldsymbol{M}^{-1}\boldsymbol{P}_{1/2} + \phi_\Delta - U^{\mathrm{C}}_\Delta = \Delta t\mathscr{X}\boldsymbol{R}_{1/2}\cdot\boldsymbol{M}^{-1}\boldsymbol{P}_{1/2}.$$

Writing $\mathscr{X} = \kappa\overline{\mathscr{X}}$ now furnishes the result given in (4.8). □

A.4. Proposition 4 (preservation of relative equilibria)

We first require a preliminary lemma:

Lemma A.1. *Let $\boldsymbol{A} \in \mathbb{R}^{m\times m}$ be an arbitrary real square matrix, and t, t_1, t_2 be arbitrary real scalars. Define $\boldsymbol{B} := \mathrm{diag}(\boldsymbol{A}) \in \mathbb{R}^{mn\times mn}$ and let $\boldsymbol{C} \in \mathbb{R}^{mn\times mn}$ be a real matrix with arbitrary unit submatrices $\boldsymbol{C}^{ij} := c_{ij}\boldsymbol{I}_m \in \mathbb{R}^{m\times m}$. Then the following statements are true*:

(i) $\boldsymbol{A}$ *commutes with* $\exp(t\boldsymbol{A})$,
(ii) $\exp[(t_1+t_2)\boldsymbol{A}] = \exp(t_1\boldsymbol{A})\exp(t_2\boldsymbol{A})$,
(iii) $\exp(t\boldsymbol{A})$ *is non-singular, with* $\exp(t\boldsymbol{A})^{-1} = \exp(-t\boldsymbol{A})$, *and*
(iv) $\boldsymbol{B}$ *commutes with* $\boldsymbol{C}$.

Proof. From (2.38), we have

$$\boldsymbol{A}\exp(t\boldsymbol{A}) = \boldsymbol{A}\sum_{s=0}^{\infty}\frac{t^s}{s!}\boldsymbol{A}^s = \sum_{s=0}^{\infty}\frac{t^s}{s!}\boldsymbol{A}^{s+1} = \left(\sum_{s=0}^{\infty}\frac{t^s}{s!}\boldsymbol{A}^s\right)\boldsymbol{A} = \exp(t\boldsymbol{A})\boldsymbol{A}$$

which establishes (i). Also from (2.38), we have

$$\exp(t_1\boldsymbol{A})\exp(t_2\boldsymbol{A}) = \left(\sum_{q=0}^{\infty}\frac{(t_1)^q}{q!}\boldsymbol{A}^q\right)\left(\sum_{r=0}^{\infty}\frac{(t_2)^r}{r!}\boldsymbol{A}^r\right) = \sum_{s=0}^{\infty}\left(\sum_{q+r=s}\frac{(t_1)^q}{q!}\frac{(t_2)^r}{r!}\right)\boldsymbol{A}^s = \sum_{s=0}^{\infty}\frac{1}{s!}\left(\sum_{r=0}^{s}\frac{s!}{(s-r)!r!}(t_1)^{s-r}(t_2)^r\right)\boldsymbol{A}^s$$
$$= \sum_{s=0}^{\infty}\frac{(t_1+t_2)^s}{s!}\boldsymbol{A}^s = \exp[(t_1+t_2)\boldsymbol{A}]$$

which establishes (ii), and (iii) follows by putting $t_1 = t$ and $t_2 = -t$, since $\exp(\boldsymbol{0}_m) := \boldsymbol{I}_m$. Finally, for $\boldsymbol{B}$ and $\boldsymbol{C}$ as given, we have

$$(\boldsymbol{BC})^{ij} = \sum_{k=1}^{n}\boldsymbol{B}^{ik}\boldsymbol{C}^{kj} = c_{ij}\boldsymbol{A} \quad \text{since } \boldsymbol{B}^{ij} = \boldsymbol{0}_m \text{ for } i \neq j,$$

and also

$$(\boldsymbol{CB})^{ij} = \sum_{k=1}^{n}\boldsymbol{C}^{ik}\boldsymbol{B}^{kj} = c_{ij}\boldsymbol{A};$$

hence $\boldsymbol{BC} = \boldsymbol{CB}$ as given in (iv). □

We will also make use of the identity

$$\exp(\hat{\boldsymbol{\theta}}) - \boldsymbol{I}_3 = \frac{\tan\left(\frac{1}{2}\theta\right)}{\theta}[\exp(\hat{\boldsymbol{\theta}}) + \boldsymbol{I}_3]\hat{\boldsymbol{\theta}} \quad \text{for } \theta := \|\boldsymbol{\theta}\|, \tag{A.2}$$

which can be established using the formula

$$\exp(\hat{\boldsymbol{\theta}}) = \boldsymbol{I}_3 + \frac{\sin(\theta)}{\theta}\hat{\boldsymbol{\theta}} + 2\frac{\sin^2\left(\frac{1}{2}\theta\right)}{\theta^2}\hat{\boldsymbol{\theta}}^2 \tag{A.3}$$

given in [36]. From (2.32) we note that $\hat{\boldsymbol{\theta}}^3 = -\theta^2\hat{\boldsymbol{\theta}}$, and so using standard trigonometric identities we get

$$\begin{aligned}[\exp(\hat{\boldsymbol{\theta}}) + \boldsymbol{I}_3]\hat{\boldsymbol{\theta}} &= 2\left[1 - \sin^2\left(\frac{1}{2}\theta\right)\right]\hat{\boldsymbol{\theta}} + \frac{\sin(\theta)}{\theta}\hat{\boldsymbol{\theta}}^2 = \frac{\cos\left(\frac{1}{2}\theta\right)}{\sin\left(\frac{1}{2}\theta\right)}\left[2\sin\left(\frac{1}{2}\theta\right)\cos\left(\frac{1}{2}\theta\right)\hat{\boldsymbol{\theta}} + 2\frac{\sin^2\left(\frac{1}{2}\theta\right)}{\theta}\hat{\boldsymbol{\theta}}^2\right] \\ &= \frac{\theta}{\tan\left(\frac{1}{2}\theta\right)}\left[\frac{\sin(\theta)}{\theta}\hat{\boldsymbol{\theta}} + 2\frac{\sin^2\left(\frac{1}{2}\theta\right)}{\theta^2}\hat{\boldsymbol{\theta}}^2\right] = \frac{\theta}{\tan\left(\frac{1}{2}\theta\right)}[\exp(\hat{\boldsymbol{\theta}}) - \boldsymbol{I}_3],\end{aligned}$$

which leads to (A.2).

We can now begin the proof of Proposition 4. First, we define the abbreviation

$$\boldsymbol{E}(k\Delta t) := \exp(\lambda_{\mathrm{r}} k\Delta t\widehat{\boldsymbol{\Omega}}_0) \equiv \mathrm{diag}[\exp(\lambda_{\mathrm{r}} k\Delta t\hat{\boldsymbol{\omega}}_0)]. \tag{A.4}$$

From (4.17), we then have

$$\boldsymbol{R}_k = \boldsymbol{R}_0^{\mathrm{c}} + \lambda_{\mathrm{t}} k\Delta t\boldsymbol{V}_0^{\mathrm{c}} + \boldsymbol{E}(k\Delta t)\overline{\boldsymbol{R}}_0 \quad \text{and} \quad \boldsymbol{P}_k = \boldsymbol{M}[\boldsymbol{V}_0^{\mathrm{c}} + \boldsymbol{E}(k\Delta t)\overline{\boldsymbol{V}}_0] \qquad \forall k,$$

which leads to

$$\begin{aligned}\boldsymbol{R}_\Delta &= \lambda_{\mathrm{t}}\Delta t\boldsymbol{V}_0^{\mathrm{c}} + (\boldsymbol{E}[(n+1)\Delta t] - \boldsymbol{E}(n\Delta t))\overline{\boldsymbol{R}}_0, \\ \boldsymbol{R}_{1/2} &= \boldsymbol{R}_0^{\mathrm{c}} + \lambda_{\mathrm{t}}\left(n + \frac{1}{2}\right)\Delta t\boldsymbol{V}_0^{\mathrm{c}} + \frac{1}{2}(\boldsymbol{E}[(n+1)\Delta t] + \boldsymbol{E}(n\Delta t))\overline{\boldsymbol{R}}_0, \\ \boldsymbol{P}_\Delta &= \boldsymbol{M}(\boldsymbol{E}[(n+1)\Delta t] - \boldsymbol{E}(n\Delta t))\overline{\boldsymbol{V}}_0 \quad \text{and} \\ \boldsymbol{P}_{1/2} &= \boldsymbol{M}\boldsymbol{V}_0^{\mathrm{c}} + \frac{1}{2}\boldsymbol{M}(\boldsymbol{E}[(n+1)\Delta t] + \boldsymbol{E}(n\Delta t))\overline{\boldsymbol{V}}_0.\end{aligned} \tag{A.5}$$

From Lemma A.1(ii), we can write $\boldsymbol{E}[(n+1)\Delta t] = \boldsymbol{E}(n\Delta t)\boldsymbol{E}(\Delta t)$, and from Lemma A.1(iv) we see that $\boldsymbol{E}(k\Delta t)$ and $\widehat{\boldsymbol{\Omega}}_0$ commute with any matrix composed of scaled unit submatrices, which obviously includes $\mathscr{G}$, $\mathscr{X}$, $\mathscr{M}$ and $\boldsymbol{M}$ and their (relevant) inverses.

Thus by inserting (A.5) into (3.8), we obtain

$$\begin{aligned}&\frac{1}{\Delta t}\boldsymbol{E}(n\Delta t)\left([\boldsymbol{E}(\Delta t) - \boldsymbol{I}_{3N}] + \frac{1}{2}[\boldsymbol{E}(\Delta t) + \boldsymbol{I}_{3N}]\mathscr{G}^{\mathrm{RE}}\right)\boldsymbol{M}\overline{\boldsymbol{V}}_0 + \frac{1}{\Delta t}\mathscr{G}^{\mathrm{RE}}\boldsymbol{M}\boldsymbol{V}_0^{\mathrm{c}} \\ &\quad = -\mathscr{X}^{\mathrm{RE}}\left[\boldsymbol{R}_0^{\mathrm{c}} + \lambda_{\mathrm{t}}\left(n + \frac{1}{2}\right)\Delta t\boldsymbol{V}_0^{\mathrm{c}}\right] - \frac{1}{2}\boldsymbol{E}(n\Delta t)[\boldsymbol{E}(\Delta t) + \boldsymbol{I}_{3N}]\mathscr{X}^{\mathrm{RE}}\overline{\boldsymbol{R}}_0, \\ &\lambda_{\mathrm{t}}\boldsymbol{V}_0^{\mathrm{c}} + \frac{1}{\Delta t}\boldsymbol{E}(n\Delta t)\left([\boldsymbol{E}(\Delta t) - \boldsymbol{I}_{3N}] - \frac{1}{2}[\boldsymbol{E}(\Delta t) + \boldsymbol{I}_{3N}]\mathscr{G}^{\mathrm{RE\,T}}\right)\overline{\boldsymbol{R}}_0 - \frac{1}{\Delta t}\mathscr{G}^{\mathrm{RE\,T}}\left[\boldsymbol{R}_0^{\mathrm{c}} + \lambda_{\mathrm{t}}\left(n + \frac{1}{2}\right)\Delta t\boldsymbol{V}_0^{\mathrm{c}}\right] \\ &\quad = (\mathscr{M}^{\mathrm{RE}})^{-1}\boldsymbol{M}\boldsymbol{V}_0^{\mathrm{c}} + \frac{1}{2}\boldsymbol{E}(n\Delta t)[\boldsymbol{E}(\Delta t) + \boldsymbol{I}_{3N}](\mathscr{M}^{\mathrm{RE}})^{-1}\boldsymbol{M}\overline{\boldsymbol{V}}_0\end{aligned} \tag{A.6}$$

if $\boldsymbol{F}_a^{\mathrm{RE}}$ and $\boldsymbol{V}_a^{\mathrm{RE}}$ are zero. Now from Propositions 1 and 2 (linear and angular momentum conservation) we know that $\sum_{j=1}^{N}\mathscr{X}^{ij} = \boldsymbol{0}_3\ \forall 1 \leqslant i \leqslant N$; hence

$$\mathscr{X}^{\mathrm{RE}}\boldsymbol{R}_0^{\mathrm{c}} = \left\langle \sum_{j=1}^{N}\mathscr{X}^{1j}\boldsymbol{r}_0^{\mathrm{c}} \quad \ldots \quad \sum_{j=1}^{N}\mathscr{X}^{Nj}\boldsymbol{r}_0^{\mathrm{c}} \right\rangle = \boldsymbol{0} \quad \text{and similarly } \mathscr{X}^{\mathrm{RE}}\boldsymbol{V}_0^{\mathrm{c}} = \boldsymbol{0}.$$

Also, Proposition 1 shows that $\mathscr{I}_N\mathscr{G} = \mathscr{O}_N \Longleftrightarrow \mathscr{G}^{\mathrm{T}}\mathscr{I}_N^{\mathrm{T}} = \mathscr{O}_N^{\mathrm{T}}$, and hence

$$\mathscr{G}^{\mathrm{RE\,T}}\boldsymbol{R}_0^{\mathrm{c}} = \mathscr{G}^{\mathrm{RE\,T}}\mathscr{I}_N^{\mathrm{T}}\boldsymbol{r}_0^{\mathrm{c}} = \boldsymbol{0} \quad \text{and similarly } \mathscr{G}^{\mathrm{RE\,T}}\boldsymbol{V}_0^{\mathrm{c}} = \boldsymbol{0}. \tag{A.7}$$

Now, Lemma A.1(iii) tells us that $\boldsymbol{E}(k\Delta t)^{-1} = \boldsymbol{E}(-k\Delta t)$, thus we can multiply each equation in (A.6) by $\boldsymbol{E}(-n\Delta t)$ to get, after rearranging,

$$\frac{1}{\Delta t}\left([\boldsymbol{E}(\Delta t)-\boldsymbol{I}_{3N}]+\frac{1}{2}[\boldsymbol{E}(\Delta t)+\boldsymbol{I}_{3N}]\mathscr{G}^{\mathrm{RE}}\right)\boldsymbol{M}\overline{\boldsymbol{V}}_0+\frac{1}{2}[\boldsymbol{E}(\Delta t)+\boldsymbol{I}_{3N}]\mathscr{X}^{\mathrm{RE}}\overline{\boldsymbol{R}}_0=-\frac{1}{\Delta t}\boldsymbol{E}(-n\Delta t)\mathscr{G}^{\mathrm{RE}}\boldsymbol{M}\boldsymbol{V}_0^{\mathrm{c}},$$
$$\frac{1}{\Delta t}\left([\boldsymbol{E}(\Delta t)-\boldsymbol{I}_{3N}]-\frac{1}{2}[\boldsymbol{E}(\Delta t)+\boldsymbol{I}_{3N}]\mathscr{G}^{\mathrm{RE\,T}}\right)\overline{\boldsymbol{R}}_0-\frac{1}{2}[\boldsymbol{E}(\Delta t)+\boldsymbol{I}_{3N}](\mathscr{M}^{\mathrm{RE}})^{-1}\boldsymbol{M}\overline{\boldsymbol{V}}_0=\boldsymbol{E}(-n\Delta t)[(\mathscr{M}^{\mathrm{RE}})^{-1}\boldsymbol{M}-\lambda_{\mathrm{t}}\boldsymbol{I}_{3N}]\boldsymbol{V}_0^{\mathrm{c}}. \tag{A.8}$$

We now invoke the relationship

$$\boldsymbol{E}(\Delta t)-\boldsymbol{I}_{3N}=\frac{\tan\left(\frac{1}{2}\theta^{\mathrm{RE}}\right)}{\omega_0}[\boldsymbol{E}(\Delta t)+\boldsymbol{I}_{3N}]\widehat{\boldsymbol{\Omega}}_0\quad\text{for }\omega_0:=\|\boldsymbol{\omega}_0\|\text{ and }\theta^{\mathrm{RE}}:=\lambda_{\mathrm{r}}\omega_0\Delta t, \tag{A.9}$$

which is easily deduced from (A.2). Substituting (A.9) into (A.8) brings

$$[\boldsymbol{E}(\Delta t)+\boldsymbol{I}_{3N}]\left(\frac{\tan\left(\frac{1}{2}\theta^{\mathrm{RE}}\right)}{\omega_0\Delta t}\widehat{\boldsymbol{\Omega}}_0\boldsymbol{M}\overline{\boldsymbol{V}}_0+\frac{1}{2\Delta t}\mathscr{G}^{\mathrm{RE}}\boldsymbol{M}\overline{\boldsymbol{V}}_0+\frac{1}{2}\mathscr{X}^{\mathrm{RE}}\overline{\boldsymbol{R}}_0\right)=-\frac{1}{\Delta t}\boldsymbol{E}(-n\Delta t)\mathscr{G}^{\mathrm{RE}}\boldsymbol{M}\boldsymbol{V}_0^{\mathrm{c}},$$
$$[\boldsymbol{E}(\Delta t)+\boldsymbol{I}_{3N}]\left(\frac{\tan\left(\frac{1}{2}\theta^{\mathrm{RE}}\right)}{\omega_0\Delta t}\widehat{\boldsymbol{\Omega}}_0\overline{\boldsymbol{R}}_0-\frac{1}{2\Delta t}\mathscr{G}^{\mathrm{RE\,T}}\overline{\boldsymbol{R}}_0-\frac{1}{2}(\mathscr{M}^{\mathrm{RE}})^{-1}\boldsymbol{M}\overline{\boldsymbol{V}}_0\right)=\boldsymbol{E}(-n\Delta t)[(\mathscr{M}^{\mathrm{RE}})^{-1}\boldsymbol{M}-\lambda_{\mathrm{t}}\boldsymbol{I}_{3N}]\boldsymbol{V}_0^{\mathrm{c}}. \tag{A.10}$$

Now it can be shown by direct calculation that $[\exp(\lambda_{\mathrm{r}}\Delta t\hat{\boldsymbol{\omega}}_0)+\boldsymbol{I}_3]^{-1}$ exists provided that $\theta^{\mathrm{RE}}\neq(2m+1)\pi; m\in\mathbb{Z}$. Hence for $-\pi<\theta^{\mathrm{RE}}<\pi$ we can define

$$\widetilde{\boldsymbol{E}}=[\boldsymbol{E}(\Delta t)+\boldsymbol{I}_{3N}]^{-1}\boldsymbol{E}(-n\Delta t)$$

and multiply (A.10) by $[\boldsymbol{E}(\Delta t)+\boldsymbol{I}_{3N}]^{-1}$ to get

$$\frac{\tan\left(\frac{1}{2}\theta^{\mathrm{RE}}\right)}{\omega_0\Delta t}\widehat{\boldsymbol{\Omega}}_0\boldsymbol{M}\overline{\boldsymbol{V}}_0+\frac{1}{2\Delta t}\mathscr{G}^{\mathrm{RE}}\boldsymbol{M}\overline{\boldsymbol{V}}_0+\frac{1}{2}\mathscr{X}^{\mathrm{RE}}\overline{\boldsymbol{R}}_0=-\frac{1}{\Delta t}\widetilde{\boldsymbol{E}}\mathscr{G}^{\mathrm{RE}}\boldsymbol{M}\boldsymbol{V}_0^{\mathrm{c}},$$
$$\frac{\tan\left(\frac{1}{2}\theta^{\mathrm{RE}}\right)}{\omega_0\Delta t}\widehat{\boldsymbol{\Omega}}_0\overline{\boldsymbol{R}}_0-\frac{1}{2\Delta t}\mathscr{G}^{\mathrm{RE\,T}}\overline{\boldsymbol{R}}_0-\frac{1}{2}(\mathscr{M}^{\mathrm{RE}})^{-1}\boldsymbol{M}\overline{\boldsymbol{V}}_0=\widetilde{\boldsymbol{E}}[(\mathscr{M}^{\mathrm{RE}})^{-1}\boldsymbol{M}-\lambda_{\mathrm{t}}\boldsymbol{I}_{3N}]\boldsymbol{V}_0^{\mathrm{c}}. \tag{A.11}$$

Pre-multiplying either equation in (A.11) by $\widehat{\boldsymbol{\Omega}}_0$ makes the right-hand side disappear, and dot-multiplying either by $\langle\boldsymbol{\omega}_0\ \cdots\ \boldsymbol{\omega}_0\rangle$ makes the $\overline{\boldsymbol{V}}_0$ (or $\widehat{\boldsymbol{\Omega}}_0\overline{\boldsymbol{R}}_0$) terms disappear. Hence each of the $\overline{\boldsymbol{R}}_0$, $\overline{\boldsymbol{V}}_0$ and $\boldsymbol{V}_0^{\mathrm{c}}$ terms in (A.11) must be individually zero, i.e.

$$\frac{\tan\left(\frac{1}{2}\theta^{\mathrm{RE}}\right)}{\frac{1}{2}\omega_0\Delta t}\widehat{\boldsymbol{\Omega}}_0\boldsymbol{M}\overline{\boldsymbol{V}}_0+\mathscr{X}^{\mathrm{RE}}\overline{\boldsymbol{R}}_0=\boldsymbol{0},\quad\frac{\tan\left(\frac{1}{2}\theta^{\mathrm{RE}}\right)}{\frac{1}{2}\omega_0\Delta t}\widehat{\boldsymbol{\Omega}}_0\overline{\boldsymbol{R}}_0-(\mathscr{M}^{\mathrm{RE}})^{-1}\boldsymbol{M}\overline{\boldsymbol{V}}_0=\boldsymbol{0},$$
$$\mathscr{G}^{\mathrm{RE}}=\boldsymbol{0}_{3N}\quad\text{and}\quad[(\mathscr{M}^{\mathrm{RE}})^{-1}\boldsymbol{M}-\lambda_{\mathrm{t}}\boldsymbol{I}_{3N}]\boldsymbol{V}_0^{\mathrm{c}}=\boldsymbol{0}, \tag{A.12}$$

since $\widetilde{\boldsymbol{E}}^{-1}$ has been shown to exist. Introducing the initial conditions (2.37) into $(\mathrm{A.12})_1$ and multiplying $(\mathrm{A.12})_2$ by $\mathscr{M}^{\mathrm{RE}}$ results in (4.18). Finally, we note as before that (4.18) assures only that (4.17) is a possible solution returned by Algorithm MC; to guarantee that it will be returned, the algorithm must also give a unique solution for $\boldsymbol{R}_{n+1}$ and $\boldsymbol{P}_{n+1}$ given $\boldsymbol{R}_n$, $\boldsymbol{P}_n$ and Δt, for all n. □

A.5. *Lemma 1 (preservation of relative equilibria and conservation of energy)*

Along a relative equilibrium path, we have the difference in energy as

$$H_\Delta=\frac{1}{2}\boldsymbol{P}_{n+1}\cdot\boldsymbol{M}^{-1}\boldsymbol{P}_{n+1}-\frac{1}{2}\boldsymbol{P}_n\cdot\boldsymbol{M}^{-1}\boldsymbol{P}_n+\phi_\Delta$$

from (4.5), since there is no external force. We also have, from (4.17) and (A.4),

$$\boldsymbol{P}_k\cdot\boldsymbol{M}^{-1}\boldsymbol{P}_k=\boldsymbol{M}[\boldsymbol{V}_0^{\mathrm{c}}+\boldsymbol{E}(k\Delta t)\overline{\boldsymbol{V}}_0]\cdot[\boldsymbol{V}_0^{\mathrm{c}}+\boldsymbol{E}(k\Delta t)\overline{\boldsymbol{V}}_0]=\boldsymbol{M}\boldsymbol{V}_0^{\mathrm{c}}\cdot\boldsymbol{V}_0^{\mathrm{c}}+2\boldsymbol{M}\boldsymbol{E}(k\Delta t)\overline{\boldsymbol{V}}_0\cdot\boldsymbol{V}_0^{\mathrm{c}}+\boldsymbol{M}\boldsymbol{E}(k\Delta t)\overline{\boldsymbol{V}}_0\cdot\boldsymbol{E}(k\Delta t)\overline{\boldsymbol{V}}_0$$

by symmetry of $\boldsymbol{M}$. From (A.3) it can be shown, using standard trigonometric identities, that for any $\hat{\boldsymbol{\theta}}\in\mathbb{R}^{3\times3}$ we have

$$\exp(\hat{\boldsymbol{\theta}})^{-1}=\exp(\hat{\boldsymbol{\theta}})^{\mathrm{T}},$$

as is widely known for rotation matrices. Thus $\boldsymbol{E}(k\Delta t)^{-1}=\boldsymbol{E}(k\Delta t)^T$, and from (Lemma A.1)(iv) we have

$$\boldsymbol{M}\boldsymbol{E}(k\Delta t)\overline{\boldsymbol{V}}_0\cdot\boldsymbol{E}(k\Delta t)\overline{\boldsymbol{V}}_0=\boldsymbol{E}(k\Delta t)^{\mathrm{T}}\boldsymbol{E}(k\Delta t)\boldsymbol{M}\overline{\boldsymbol{V}}_0\cdot\overline{\boldsymbol{V}}_0=\boldsymbol{M}\overline{\boldsymbol{V}}_0\cdot\overline{\boldsymbol{V}}_0;$$

thus we can now write

$$\boldsymbol{P}_k \cdot \boldsymbol{M}^{-1}\boldsymbol{P}_k = \boldsymbol{M}\boldsymbol{V}_0^{\mathrm{c}} \cdot \boldsymbol{V}_0^{\mathrm{c}} + 2\boldsymbol{M}\boldsymbol{E}(k\Delta t)\overline{\boldsymbol{V}}_0 \cdot \boldsymbol{V}_0^{\mathrm{c}} + \boldsymbol{M}\overline{\boldsymbol{V}}_0 \cdot \overline{\boldsymbol{V}}_0. \tag{A.13}$$

Also, from (2.32), (2.35)$_1$ and (A.3) we have

$$\exp(\alpha\hat{\boldsymbol{\omega}}_0)\boldsymbol{v}_0^{\mathrm{c}} = \left[\boldsymbol{I}_3 + \frac{\sin(\alpha\omega_0)}{\omega_0}\hat{\boldsymbol{\omega}}_0 + 2\frac{\sin^2\left(\frac{1}{2}\alpha\omega_0\right)}{\omega_0^2}\hat{\boldsymbol{\omega}}_0^2\right]\boldsymbol{v}_0^{\mathrm{c}} = \boldsymbol{v}_0^{\mathrm{c}}\ \forall\alpha,$$

and since $\boldsymbol{E}(k\Delta t)^{-1} = \boldsymbol{E}(-k\Delta t)$, we can write

$$\boldsymbol{M}\boldsymbol{E}(k\Delta t)\overline{\boldsymbol{V}}_0 \cdot \boldsymbol{V}_0^{\mathrm{c}} = \boldsymbol{M}\overline{\boldsymbol{V}}_0 \cdot \boldsymbol{E}(-k\Delta t)\boldsymbol{V}_0^{\mathrm{c}} = \boldsymbol{M}\overline{\boldsymbol{V}}_0 \cdot \boldsymbol{V}_0^{\mathrm{c}}.$$

Therefore (A.13) becomes

$$\boldsymbol{P}_k \cdot \boldsymbol{M}^{-1}\boldsymbol{P}_k = \boldsymbol{M}\boldsymbol{V}_0^{\mathrm{c}} \cdot \boldsymbol{V}_0^{\mathrm{c}} + 2\boldsymbol{M}\overline{\boldsymbol{V}}_0 \cdot \boldsymbol{V}_0^{\mathrm{c}} + \boldsymbol{M}\overline{\boldsymbol{V}}_0 \cdot \overline{\boldsymbol{V}}_0 = \boldsymbol{P}_0 \cdot \boldsymbol{M}^{-1}\boldsymbol{P}_0\ \forall k,$$

and thus $\frac{1}{2}\boldsymbol{P}_{n+1} \cdot \boldsymbol{M}^{-1}\boldsymbol{P}_{n+1} - \frac{1}{2}\boldsymbol{P}_n \cdot \boldsymbol{M}^{-1}\boldsymbol{P}_n = 0$.

Finally, since $\phi = \sum_{i,j=1}^{N}\phi_{ij}$ is a function of the element lengths $l_{ij} = \|\boldsymbol{r}^j - \boldsymbol{r}^i\|$ only, the difference in strain energy is solely dependent on the differences $\|\boldsymbol{r}^j - \boldsymbol{r}^i\|_\Delta$ in the lengths of the bars. From (4.17) we have

$$\boldsymbol{r}_k^i = \boldsymbol{r}_0^{\mathrm{c}} + \lambda_{\mathrm{t}}k\Delta t\boldsymbol{v}_0^{\mathrm{c}} + \exp(\lambda_{\mathrm{r}}k\Delta t\hat{\boldsymbol{\omega}}_0)(\boldsymbol{r}_0^i - \boldsymbol{r}_0^{\mathrm{c}}) \ \Rightarrow\ \boldsymbol{r}_k^j - \boldsymbol{r}_k^i = \exp(\lambda_{\mathrm{r}}k\Delta t\hat{\boldsymbol{\omega}}_0)(\boldsymbol{r}_0^j - \boldsymbol{r}_0^i),$$

and so

$$\|\boldsymbol{r}_k^j - \boldsymbol{r}_k^i\| = \sqrt{\exp(\lambda_{\mathrm{r}}k\Delta t\hat{\boldsymbol{\omega}}_0)(\boldsymbol{r}_0^j - \boldsymbol{r}_0^i) \cdot \exp(\lambda_{\mathrm{r}}k\Delta t\hat{\boldsymbol{\omega}}_0)(\boldsymbol{r}_0^j - \boldsymbol{r}_0^i)} = \|\boldsymbol{r}_0^j - \boldsymbol{r}_0^i\|\ \forall k.$$

Hence $\phi_\Delta = 0$, and thus $H_\Delta = 0$. □

A.6. Proposition 5 (time reversibility)

From (4.20), we see that Algorithm MC is time-reversible if and only if

$$\mathscr{B}_{n+1}^{\mathrm{TR}} = \mathscr{B}_{n+1}^{-1} \quad \text{and} \quad \boldsymbol{Z}_F^{\mathrm{TR}} = -\mathscr{B}_{n+1}^{-1}\boldsymbol{Z}_F. \tag{A.14}$$

Using (3.12), we can write (A.14) as

$$\left(\boldsymbol{I}_{6N} - \frac{1}{2}\mathscr{C}_{n+1}^{\mathrm{TR}}\right)^{-1}\left(\boldsymbol{I}_{6N} + \frac{1}{2}\mathscr{C}_{n+1}^{\mathrm{TR}}\right) = \left(\boldsymbol{I}_{6N} + \frac{1}{2}\mathscr{C}_{n+1}\right)^{-1}\left(\boldsymbol{I}_{6N} - \frac{1}{2}\mathscr{C}_{n+1}\right) \quad \text{and}$$
$$\left(\boldsymbol{I}_{6N} - \frac{1}{2}\mathscr{C}_{n+1}^{\mathrm{TR}}\right)^{-1}\boldsymbol{Z}_a^{\mathrm{TR}} = -\left(\boldsymbol{I}_{6N} + \frac{1}{2}\mathscr{C}_{n+1}\right)^{-1}\boldsymbol{Z}_a,$$

which leads to

$$\mathscr{C}_{n+1}^{\mathrm{TR}} = -\mathscr{C}_{n+1} \quad \text{and} \quad \boldsymbol{Z}_a^{\mathrm{TR}} = -\boldsymbol{Z}_a.$$

Eq. (4.21) then follows from (3.11). □

A.7. Proposition 6

If an algorithm is pth-order accurate, then (5.1) shows that

$$\boldsymbol{\epsilon} = \boldsymbol{Z}_{n+1} - \boldsymbol{Z}(t_{n+1}) \in \mathscr{O}(\Delta t^{p+1}) \tag{A.15}$$

when $\boldsymbol{Z}_n = \boldsymbol{Z}(t_n)$. Therefore we have

$$\boldsymbol{f}(\boldsymbol{Z}_{n+1}) = \boldsymbol{f}\ [\boldsymbol{Z}(t_{n+1}) + \boldsymbol{\epsilon}] = \boldsymbol{f}\ [\boldsymbol{Z}(t_{n+1})] + [\boldsymbol{f}\ \otimes\nabla](t_{n+1}) \cdot \boldsymbol{\epsilon} + \mathscr{O}(\|\boldsymbol{\epsilon}\|^2)$$

from Taylor's theorem, and since $\boldsymbol{f}[\boldsymbol{Z}(t)]$ is a constant of motion, we have $\boldsymbol{f}[\boldsymbol{Z}(t_{n+1})] = \boldsymbol{f}[\boldsymbol{Z}(t_n)] = \boldsymbol{f}(\boldsymbol{Z}_n)$. Thus we can write

$$\boldsymbol{f}(\boldsymbol{Z}_{n+1}) - \boldsymbol{f}(\boldsymbol{Z}_n) = [\boldsymbol{f} \otimes \nabla](t_{n+1}) \cdot \boldsymbol{\epsilon} + \mathcal{O}(\|\boldsymbol{\epsilon}\|^2),$$

where the matrix $[\boldsymbol{f} \otimes \nabla](t_{n+1})$ is independent of Δt (since it is constant). Hence $\boldsymbol{f}(\boldsymbol{Z}_{n+1}) - \boldsymbol{f}(\boldsymbol{Z}_n)$ can be at most $\mathcal{O}(\Delta t^{p+1})$ due to (A.15). □

References

[1] J.H. Argyris, P.C. Dunne, T. Angelopoulos, Dynamic response by large step integration, Earthquake Engrg. Struct. Dynam. 2 (1973) 185–203.
[2] C. Lanczos, The Variational Principles of Mechanics, Dover, New York, 1970.
[3] G. Dahlquist, A special stability problem for linear multistep methods, BIT 3 (1963) 27–43.
[4] T.J.R. Hughes, The Finite Element method: Linear Static and Dynamic Finite Element Analysis, Prentice-Hall, Englewood Cliffs, NJ, 1987.
[5] T. Belytschko, D.F. Schoeberle, On the unconditional stability of an implicit algorithm for nonlinear structural dynamics, J. Appl. Mech. 97 (1975) 865–869.
[6] T.J.R. Hughes, A note on the stability of Newmark's algorithm in nonlinear structural dynamics, Int. J. Numer. Methods Engrg. 6 (1976) 383–386.
[7] R.D. Krieg, Unconditional stability in numerical time integration methods, J. Appl. Mech. (1973) 417–421.
[8] J.C. Simo, N. Tarnow, The discrete energy–momentum method. Conserving algorithms for non-linear elastodynamics, Journal of Applied Mathematics and Physics (ZAMP). 43 (1992) 757–792.
[9] J.E. Marsden, M. West, Discrete mechanics and variational integrators, Acta Numer. (2001) 357–514.
[10] G. Zhong, J.E. Marsden, Lie–Poisson Hamilton–Jacobi theory and Lie–Poisson integrators, Phys. Lett. A 133 (3) (1988) 134–139.
[11] J.C. Simo, O. Gonzalez, Assessment of energy–momentum and symplectic schemes for stiff dynamical systems. Papers—American Society of Mechanical Engineers—All Series, 93(4), 1993. Presented at the ASME Winter Annual Meeting, New Orleans, Louisiana, November 28–December 3, 1993.
[12] O. Gonzalez, J.C. Simo, On the stability of symplectic and energy–momentum algorithms for non-linear Hamiltonian systems with symmetry, Comput. Methods Appl. Mech. Engrg. 134 (1996) 197–222.
[13] M. Ortiz, A note on energy conservation and stability of nonlinear time-stepping algorithms, Comput. Struct. 24 (1) (1986) 167–168.
[14] 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.
[15] J.C. Simo, N. Tarnow, K.K. Wong, Exact energy–momentum conserving algorithms and symplectic schemes for non-linear dynamics, Comput. Methods Appl. Mech. Engrg. 100 (1992) 63–116.
[16] J.C. Simo, D. Lewis, J.E. Marsden, Stability of relative equilibria. Part I: The reduced energy–momentum method, Arch. Rat. Mech. Anal. 115 (1991) 15–59.
[17] J.C. Simo, T.A. Posbergh, J.E. Marsden, Stability of relative equilibria. Part II: Application to nonlinear elasticity, Arch. Rat. Mech. Anal. 115 (1991) 61–100.
[18] F. Armero, 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, Comput. Methods Appl. Mech. Engrg. 190 (2001) 2603–2649.
[19] F. Armero, E. Petőcz, Formulation and analysis of conserving algorithms for frictionless dynamic contact/impact problems, Comput. Methods Appl. Mech. Engrg. 158 (3–4) (1998) 269–300.
[20] F. Armero, I. Romero, On the formulation of high-frequency dissipative time-stepping algorithms for nonlinear dynamics. Part II: Second order methods, Comput. Methods Appl. Mech. Engrg. 190 (2001) 6783–6824.
[21] H. Yoshida, Construction of higher order symplectic integrators, Phys. Lett. A 150 (5) (1990) 262–268.
[22] S. Reich, Enhancing energy conserving methods, BIT 36 (1996) 122–134.
[23] J.M. Sanz-Serna, M.P. Calvo, Numerical Hamiltonian Problems, Chapman and Hall, London, 1994.
[24] E. Hairer, G. Wanner, Solving Ordinary Differential Equations II, second ed., Springer-Verlag, Berlin, Germany, 1996.
[25] 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.
[26] E. Forest, Sixth-order Lie group integrators, Journal of Computational Physics 99 (1992) 209–213.
[27] N. Tarnow, J.C. Simo, How to render second order accurate time-stepping algorithms fourth order accurate while retaining the stability and conservation properties, Comput. Methods Appl. Mech. Engrg. 115 (1994) 233–252.
[28] P. Betsch, P. Steinmann, Inherently energy conserving time finite elements for classical mechanics, J. Comput. Phys. 160 (2000) 88–116.
[29] J.H. Argyris, P.C. Dunne, T. Angelopoulos, Non-linear oscillations using the finite element technique, Comput. Methods Appl. Mech. Engrg. 2 (1973) 203–250.
[30] R.A. LaBudde, 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 (1976) 323–346.
[31] R.A. LaBudde, D. Greenspan, Energy and momentum conserving methods of arbitrary order for the numerical integration of equations of motion II. Motion of a system of particles, Numer. Math. 26 (1976) 1–16.
[32] E. Graham, G. Jelenić, A general framework for conservative single-step time-integration schemes with higher-order accuracy for a central-force system, Comput. Methods Appl. Mech. Engrg. 192 (2003) 3585–3618.
[33] J.E. Marsden, T.J.R. Hughes, Mathematical Foundations of Elasticity, Dover, Mineola, New York, 1994.
[34] M.L. Boas, Mathematical Methods in the Physical Sciences, John Wiley & Sons, Chichester, England, 1983.
[35] R.A. Horn, C.R. Johnson, Matrix Analysis, first ed., Cambridge University Press, Cambridge, England, 1996.
[36] J. Argyris, An excursion into large rotations, Comput. Methods Appl. Mech. Engrg. 32 (1982) 85–155.
[37] U.M. Ascher, S. Reich, On some difficulties in integrating highly oscillatory Hamiltonian systems, Lecture Notes Comput. Sci. Engrg. 4 (1998) 281–296.
[38] F. Kang, Difference schemes for Hamiltonian formalism and symplectic geometry, J. Comput. Math. 4 (1986) 279–289.
[39] D. Kuhl, E. Ramm, Constraint energy momentum algorithm and its application to non-linear dynamics of shells, Comput. Methods Appl. Mech. Engrg. 136 (1996) 293–315.

[40] E. Graham, Higher-order accuracy in implicit, conservative, single-step time-integration schemes for non-linear structural dynamics, Ph.D. thesis, Imperial College of Science, Technology and Medicine, London, England, October 2003.
[41] D. Greenspan, Completely conservative covariant numerical methodology, Comput. Math. Appl. 29 (4) (1995) 37–43.
[42] E. Graham, G. Jelenić, M.A. Crisfield, A note on the equivalence of two recent time-integration schemes for *N*-body problems, Commun. Numer. Methods Engrg. 18 (2002) 615–620.
[43] J.C. Simo, N. Tarnow, M. Doblare, Non-linear dynamics of three-dimensional rods: exact energy and momentum conserving algorithms, Int. J. Numer. Methods Engrg. 38 (1995) 1431–1473.
[44] M.A. Crisfield, Non-linear Finite Element Analysis of Solids and Structures, vol. 1, John Wiley & Sons, Chichester, UK, 1991.