Variational time integrators

A. Lew[1], J. E. Marsden[2], M. Ortiz[1,*,†] and M. West[2]

[1]*Graduate Aeronautical Laboratories 105-50, California Institute of Technology, Pasadena, CA 91125, U.S.A.*
[2]*Control and Dynamical Systems 107-81, California Institute of Technology, Pasadena, CA 91125, U.S.A.*

SUMMARY

The purpose of this paper is to review and further develop the subject of variational integration algorithms as it applies to mechanical systems of engineering interest. In particular, the conservation properties of both synchronous and asynchronous variational integrators (AVIs) are discussed in detail. We present selected numerical examples which demonstrate the excellent accuracy, conservation and convergence characteristics of AVIs. In these tests, AVIs are found to result in substantial speed-ups, at equal accuracy, relative to explicit Newmark. A mathematical proof of convergence of the AVIs is also presented in this paper. Finally, we develop the subject of horizontal variations and configurational forces in discrete dynamics. This theory leads to exact path-independent characterizations of the configurational forces acting on discrete systems. Notable examples are the configurational forces acting on material nodes in a finite element discretisation; and the J-integral at the tip of a crack in a finite element mesh. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: elastodynamics; geometric integration; multi-time-step; discrete mechanics; subcycling; variational integrators

1. INTRODUCTION

The purpose of this paper is to review and further develop the subject of variational integration algorithms as it applies to mechanical systems of engineering interest. The idea behind this class of algorithms is to discretize the variational formulation of a given problem. These problems may be either conservative or dissipative and forced. For conservative problems, we focus on discretizing Hamilton's principle of stationary action in Lagrangian mechanics, while for dissipative/forced problems one can, for example, discretize the Lagrange–d'Alembert principle. While the idea of discretizing variational formulations of mechanics is standard for elliptic problems, in the form of Galerkin and finite element methods (e.g. References [1, 2]),

it has only been applied relatively recently to derive variational time-stepping algorithms for mechanical systems.

The basic idea: For conservative systems, one typically starts with a Lagrangian $L(q, \dot{q})$ with $q = (q^1, q^2, \ldots, q^n)$ as configuration space variables and requires Hamilton's principle, namely make the action integral stationary:

$$\delta \int_a^b L(q, \dot{q})\, \mathrm{d}t = 0$$

for fixed endpoints. This of course leads to the Euler–Lagrange equations:

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{q}^i} - \frac{\partial L}{\partial \dot{q}^i} = 0$$

This theory has a PDE counterpart in which one makes stationary a spacetime integral appropriate for classical field theories, such as elasticity and fluids.

In discrete mechanics, we make an approximation of the action integral, which gives the *discrete Lagrangian*:

$$L_d(q_0, q_1, \Delta t) \approx \int_0^{\Delta t} L(q(t), \dot{q}(t))\, \mathrm{d}t$$

where $q(t)$ is the *exact solution* joining q_0 and q_1 of the Euler–Lagrange equations for L; the time step is Δt. Note that in this theory, the discrete Lagrangian is thought of as a function of two points and a time interval (or more generally, of two points in configuration space and two times). One then forms a discrete action sum and requires it to be stationary, which leads to the *discrete Euler–Lagrange equations*, the counterpart of the continuous Euler–Lagrange equations. These discrete equations comprise the integration algorithm.

For forced or dissipative systems, one can apply the same methodology to the Lagrange–d'Alembert principle:

$$\delta \int_a^b L(q, \dot{q})\, \mathrm{d}t + \int_a^b F\delta q\, \mathrm{d}t = 0$$

where F is the given external force field.

Accuracy and computing what matters: Perhaps the most often asked question about variational integrators is whether or not they improve the accuracy of computations. For individual trajectories in a dynamically rich problem where even round-off error can quickly accumulate due to the divergence of nearby trajectories, the answer is of course not—they are no better than traditional algorithms, but also no worse. In the realm of variational integrators there are the usual trade-offs between accuracy and efficiency.

However, there is much more to the story. In many problems, one is not just interested in the accuracy of individual trajectories; in fact, for complex dynamical processes integrated over intermediate to long time scales, accuracy of individual trajectories is not an appropriate thing to require any more than one might try to fully resolve a complex fluid computation. Rather, one should ask if statistical quantities are captured accurately and for this problem, variational integrators are demonstrably superior. There are many examples verifying this in the literature (such as the accurate computation of complex invariant sets in References [3, 4]), but one of

them that is striking and easy to understand is the problem of computing the 'heat' of a system of interacting particles. As we explain in the body of the paper, this is the type of quantity that variational methods can compute accurately and, for many questions, this is the type of quantity one wants to compute.

Conserved quantities: Consistent with computing statistical quantities accurately, one must of course also compute the basic conservation laws accurately. One of the very nice things about a discrete variational structure is that the link between symmetry and conserved quantities is exactly the same as it is in the continuous theory. That is, if one constructs the discrete variational principle to respect the symmetry (such as rotational invariance for angular momentum), then there will be a corresponding conserved quantity that is exactly respected by the discrete algorithm. Of course it is well known (and examples are given in the main body of the paper) that standard unstructured, even more highly accurate algorithms do not have this property.

Energy: Energy, associated with time-translation symmetry in the conservative case, plays a special role. If one does time adaption by asking that the discrete action sum be stationary with respect to the time discretization, then discrete conservation of energy is also obtained. However, this is sometimes at conflict with other considerations, such as stability analysis or computational efficiency. Amazingly, however, for constant time-stepping algorithms (or constant at each point in space in the case of asynchronous variational integrators (AVIs)), energy is still nearly conserved. The reason for this is a deeper property of variational algorithms, called symplecticity (a symplectic map is synonymous with a time-independent canonical transformation in the classical literature). This by itself, may seem somewhat mathematical and irrelevant at first sight, but it is the key to understanding the remarkable energy behaviour of variational schemes. We shall explain the notion of symplecticity in concrete and easy to understand terms in the text, but it is a deep notion that underlies all of modern geometric mechanics. In particular, through a process called *backward error analysis*, it is the key to understanding this approximate energy conservation. The basic idea is to show that the algorithm is exactly energy conserving (up to exponentially small terms) for a *nearby* Hamiltonian system. This is important work of many people; we refer to Reference [5] for one of these and to Reference [6] for additional references.

Note on symplecticity: In some cases, symplecticity reduces to well known and easy to understand principles, including the Betti reciprocity principle and other well-known reciprocity principles in mechanics.

In regard to energy preserving properties, symplecticity and momentum conservation, we should mention that there are *a priori* restrictions on what an algorithm can do. In the world of constant time-stepping algorithms, in general (the exception being certain integrable systems), they cannot be energy preserving, symplectic and momentum preserving, as was shown by Ge and Marsden [7]. This led to two general classes of algorithms, namely symplectic-momentum and energy-momentum algorithms. Over the years, we have come to favour the variational context since it can give, with the appropriate choices, any of these, as was explained. However, a direct approach to the design of energy-momentum integrators was pursued by Gonzalez and Simo [8] and related references.

We should also mention that there is a considerable advantage to the design and construction of symplectic methods using the variational approach. For instance, one can directly approximate the action integral to a certain degree of accuracy using one's favourite quadrature method and a variational integrator of corresponding accuracy will be the result. The older methods of using approximations to solutions to the Hamilton Jacobi equation to produce a generating

function for a symplectic transformation (see, e.g. Reference [3] and references therein) are more awkward to implement. Approximation of the action integral (which gives Jacobi's solution to the Hamilton–Jacobi equation) is a more efficient way of accomplishing the same end.

We should also mention that there is a Hamiltonian side to this theory, including the variational way of constructing integrators based on Hamilton's phase space, as opposed to the configuration space variational principle, but we shall not discuss these aspects here (see for example, References [9–11] for the traditional Hamiltonian approaches as well as Marsden and West [6] for remarks on the variational approach).

Constraints: If there are constraints on a system, then one can still use, very effectively, variational methods and they naturally produce standard constraint algorithms. The story starts with the SHAKE algorithm, which was first proposed by Ryckaert *et al.* [12] as a constrained version of the Verlet algorithm. A constrained version of the velocity Verlet integrator, RATTLE, was given by Anderson [13], which was later shown by Leimkuhler and Skeel [14] to be a symplectic integrator. All of these results naturally fall out of the variational approach with the constraints realized in terms of the Lagrangian augmented by suitable penalty functions. Variational methods have also been applied to collision algorithms, as in References [15–17] and references therein. The main achievement in these works is to show that properties of variational integrators remain valid right through the collision process.

PDE algorithms and AVIs: The extension of the idea of variational integrators to the PDE context was made by Marsden *et al.* [18]. They showed, in a demonstration example, that the method was very promising for variational integrators in a spacetime context. As was mentioned, this approach was considerably developed, extended and applied in Reference [19]. The asynchronous algorithms developed in that paper, and further in the present work, share many features in common with multi-time-step integration algorithms, sometimes termed subcycling methods. These algorithms have been developed in References [20, 21], mainly to allow high-frequency elements to advance at smaller time steps than the low-frequency ones. In its original version, the method grouped the nodes of the mesh and assigned to each group a different time step. Adjacent groups of nodes were constrained to have integer time-step ratios (see Reference [21]), a condition that was relaxed in References [20, 22]. Recently an implicit multi-time-step integration method was developed and analysed in Reference [23]. We also mention the related work of Hughes and Liu [24] and Hughes *et al.* [25]. The freedom to choose the time step for each element, subject to stability considerations, as well as the way nodes are updated, are the distinguishing features of the asynchronous algorithms discussed in this paper.

Connections to other work: There are also many connections between the multi-time-step impulse method (also known as Verlet-I and r-RESPA), which is popular in molecular dynamics applications, and the AVI algorithm developed in this paper (see References [26, 27]). Thus, when applied to a system of ODEs the AVI method may be regarded as a fully asynchronous generalization of the impulse method. Many algorithms that are widely used in engineering applications are variational or are close to being variational. For instance, Kane *et al.* [28] have shown that Newmark's algorithm is variational, although not with respect to the original Lagrangian but with respect to a closely related one. These variational underpinnings may indeed account for the robustness and good behaviour of the algorithm.

To fully understand the methods discussed in this paper, it is best to work intrinsically on manifolds as opposed to relying on generalized co-ordinates. However, in this paper we confine ourselves to the latter approach in the interest of readability. We do include, however, geometrical asides in order to hint at the more general manifold picture. For a comprehensive

review of the mathematical foundations of variational integrators in the ODE context, as well as a survey of the previous literature, the interested reader is referred to Reference [6].

2. DISCRETE DYNAMICS AND VARIATIONAL INTEGRATORS

In this section, we give a brief overview of how discrete variational mechanics can be used to derive variational integrators. We begin by reviewing the derivation of the Euler–Lagrange equations, and then show how to mimic this process on a discrete level.

2.1. Continuous time Lagrangian dynamics

For concreteness, consider the Lagrangian system $L(q, \dot{q}) = \frac{1}{2}\dot{q}^{\mathsf{T}}M\dot{q} - V(q)$, where M is a symmetric positive-definite mass matrix and V is a potential function. We work in $\mathbb{R}^n$ or in generalized co-ordinates and will use vector notation for simplicity, so $q = (q^1, q^2, \ldots, q^n)$. In the standard approach of Lagrangian mechanics, we form the action function by integrating L along a curve $q(t)$ and then compute variations of the action while holding the endpoints of the curve $q(t)$ fixed. This gives

$$\delta S(q) = \delta \int_0^T L(q(t), \dot{q}(t))\,\mathrm{d}t = \int_0^T \left[\frac{\partial L}{\partial q}\,\delta q + \frac{\partial L}{\partial \dot{q}}\,\delta \dot{q} \right]\mathrm{d}t$$

$$= \int_0^T \left[\frac{\partial L}{\partial q} - \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{q}} \right) \right]\delta q\,\mathrm{d}t + \left[\frac{\partial L}{\partial \dot{q}}\delta q \right]_0^{\mathsf{T}} \tag{1}$$

where we have used integration by parts. The final term is zero because we assume that $\delta q(T) = \delta q(0) = 0$. Requiring that the variations of the action be zero for all δq implies that the integrand must be zero for each time t, giving the well-known *Euler–Lagrange equations*

$$\frac{\partial L}{\partial q}(q, \dot{q}) - \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{q}}(q, \dot{q}) \right) = 0 \tag{2}$$

For the particular form of the Lagrangian chosen above, this is just

$$M\ddot{q} = -\nabla V(q)$$

which is Newton's equation: mass times acceleration equals force. It is well known that the system described by the Euler–Lagrange equations has many special properties. In particular, the flow on state space is symplectic, meaning that it conserves a particular two-form, and if there are symmetry actions on phase space then there are corresponding conserved quantities of the flow, known as momentum maps. We will return to these ideas later in the article.

2.2. Discrete time Lagrangian dynamics

We will now see how discrete variational mechanics performs an analogue of the above derivation. Rather than taking a position q and velocity $\dot{q}$, consider now two positions q_0 and q_1 and a time step $\Delta t \in \mathbb{R}$. These positions should be thought of as being two points on a curve at time Δt apart, so that $q_0 \approx q(0)$ and $q_1 \approx q(\Delta t)$.

We now consider a discrete Lagrangian $L_d(q_0, q_1, \Delta t)$, which we think of as approximating the action integral along the curve segment between q_0 and q_1. For concreteness, consider the very simple approximation to the integral $\int_0^T L \, dt$ given by using the rectangle rule[‡] (the length of the interval times the value of the integrand with the velocity vector replaced by $(q_1 - q_0)/\Delta t$):

$$L_d(q_0, q_1, \Delta t) = \Delta t \left[\frac{1}{2} \left(\frac{q_1 - q_0}{\Delta t} \right)^{\mathsf{T}} M \left(\frac{q_1 - q_0}{\Delta t} \right) - V(q_0) \right] \tag{3}$$

Next consider a discrete curve of points $\{q_k\}_{k=0}^N$ and calculate the discrete action along this sequence by summing the discrete Lagrangian on each adjacent pair. Following the continuous derivation above, we compute variations of this action sum with the boundary points q_0 and q_N held fixed. This gives

$$\delta S_d(\{q_k\}) = \delta \sum_{k=0}^{N-1} L_d(q_k, q_{k+1}, \Delta t) = \sum_{k=0}^{N-1} [D_1 L_d(q_k, q_{k+1}, \Delta t)\delta q_k + D_2 L_d(q_k, q_{k+1}, \Delta t)\delta q_{k+1}]$$

$$= \sum_{k=1}^{N-1} [D_2 L_d(q_{k-1}, q_k, \Delta t) + D_1 L_d(q_k, q_{k+1}, \Delta t)]\delta q_k$$

$$+ D_1 L_d(q_0, q_1, \Delta t)\delta q_0 + D_2 L_d(q_{N-1}, q_N, \Delta t)\delta q_N \tag{4}$$

where we have used a discrete integration by parts (rearranging the summation). Henceforth, $D_i L_d$ indicates the slot derivative with respect to the ith argument of L_d. If we now require that the variations of the action be zero for any choice of δq_k with $\delta q_0 = \delta q_N = 0$, then we obtain the *discrete Euler–Lagrange equations*

$$D_2 L_d(q_{k-1}, q_k, \Delta t) + D_1 L_d(q_k, q_{k+1}, \Delta t) = 0 \tag{5}$$

which must hold for each k. For the particular L_d chosen above, we compute

$$D_2 L_d(q_{k-1}, q_k, \Delta t) = M \left(\frac{q_k - q_{k-1}}{\Delta t} \right)$$

$$D_1 L_d(q_k, q_{k+1}, \Delta t) = -\left[M \left(\frac{q_{k+1} - q_k}{\Delta t} \right) + (\Delta t)\nabla V(q_k) \right]$$

and so the discrete Euler–Lagrange equations are

$$M \left(\frac{q_{k+1} - 2q_k + q_{k-1}}{(\Delta t)^2} \right) = -\nabla V(q_k)$$

This is clearly a discretization of Newton's equations, using a simple finite difference rule for the derivative.

If we take initial conditions (q_0, q_1) then the discrete Euler–Lagrange equations define a recursive rule for calculating the sequence $\{q_k\}_{k=0}^N$. Regarded in this way, they define a map

---

[‡]As we shall see later, more sophisticated quadrature rules lead to higher-order accurate integrators.
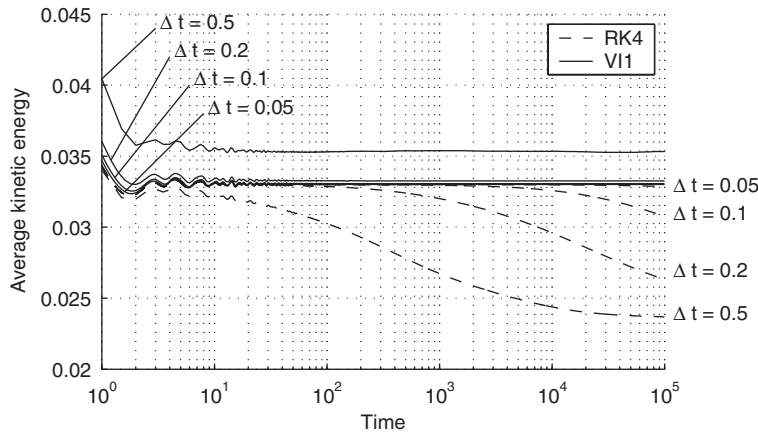
Figure 1. Average kinetic energy (6) as a function of T for a non-linear spring–mass lattice system, using a first-order variational integrator (VI1) and a fourth-order Runge–Kutta method (RK4) and a range of time steps Δt. Observe that the Runge–Kutta method suffers substantial numerical dissipation, unlike the variational method.

$(q_k, q_{k+1}) \mapsto (q_{k+1}, q_{k+2})$ which we can think of as a one-step integrator for the system defined by the continuous Euler–Lagrange equations.

Heat calculation example: As we will consider in detail in Section 3, variational integrators are interesting because they inherit many of the conservative properties of the original Lagrangian system. As an example of this, we consider the numerical approximation of the heat of a coupled spring–mass lattice model. The numerical heat for time T is defined to be the numerical approximation of

$$\bar{K}(T) = \frac{1}{T} \int_0^T \frac{1}{2} \|\dot{q}\|^2 \, \mathrm{d}t \tag{6}$$

while the true heat of the system is the limit of the quantity,

$$\bar{K} = \lim_{T \to \infty} \bar{K}(T) \tag{7}$$

The temperature of the system, which is an intensive—as opposed to extensive—quantity, is the heat K divided by the heat capacity nd, where n is the number of masses and d is the dimension of space. We assume that the system is ergodic and that this limit exists. In Figure 1, we plot the numerical approximations to (6) at $T = 10^5$ computed using a first-order variational integrator (VI1) and a fourth-order Runge–Kutta method (RK4). As the time step is decreased the numerical solution tends towards the true solution.

Note, however, that the lack of dissipation in the variational integrator means that for quite large time steps it computes the averaged kinetic energy much better. To make this precise, we consider the harmonic approximation to the lattice system (that is, the linearization), for which we can compute limit (7) analytically. The error in the numerically computed heat is plotted in Figure 2 for a range of different time steps Δt and final times T, using the same first-order variational method (VI1) and fourth-order Runge–Kutta method (RK4), as well as a fourth-order variational integrator (VI4).
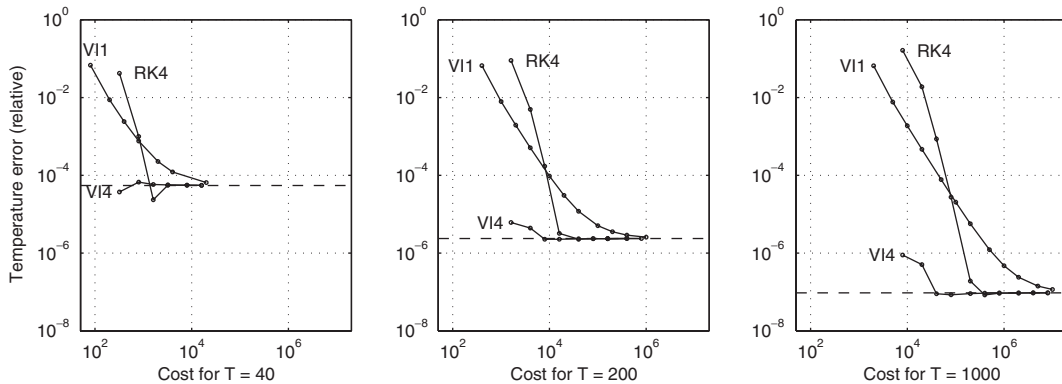
Figure 2. Error in numerically approximated heat for the harmonic (linear) approximation to the lattice system from Figure 1, using a first-order variational integrator (VI1), a fourth-order Runge–Kutta method (RK4) and a fourth-order variational integrator (VI4). The three plots have different final times T, while the cost is increased within each plot by decreasing the time step Δt. For each T the dashed horizontal line is the exact value of $\bar{K}(T) - \bar{K}$, which is the minimum error that the numerical approximation can achieve without increasing T. Observe that the low-order variational method VI1 beats the traditional RK4 method for larger errors, while the high-order variational method VI4 combines the advantages of both high order and variational structure to always win.

To compute the heat (7) numerically we must clearly let $T \to \infty$ and $\Delta t \to 0$. Both of these limits increase the cost of the simulation, and so there is a trade-off between them. As we see in Figure 2, for a fixed T there is some Δt which adequately resolves the integral (6), and so the error cannot decrease any further without increasing T. To see this, take a numerical approximation $\bar{K}(T, \Delta t)$ to (6) and decompose the error as

$$\underbrace{\bar{K}(T, \Delta t) - \bar{K}}_{\text{total error}} = \underbrace{\bar{K}(T, \Delta t) - \bar{K}(T)}_{\text{discretization error}} + \underbrace{\bar{K}(T) - \bar{K}}_{\text{limit error}} \qquad (8)$$

Decreasing Δt will reduce the discretization error, but at some point this will become negligible compared to the limit error, which will only tend to zero as T is increased.

The striking feature of Figure 2 is that the variational integrators perform far better than a traditional Runge–Kutta method. For large error tolerances, such as 1 or 5% error (10^{-2} or 5×10^{-2} in Figure 2), the first-order variational method is very cheap and simple. For higher precision, the fourth-order Runge–Kutta method eventually becomes cheaper than the first-order variational integrator, but the fourth-order variational method combines the advantages of both and is always the method of choice.

Of course, such sweeping statements as above have to be interpreted and used with great care, as in the precise statements in the text that follows. For example, if the integration step size is too large, then sometimes energy can behave very badly, even for a variational integrator (see, for example, Reference [8]). It is likewise well known that energy conservation does not guarantee trajectory accuracy. These points will be discussed further below.

2.3. Variational integrators

We are primarily interested in discrete Lagrangian mechanics for deriving integrators for mechanical systems. Any integrator which is the discrete Euler–Lagrange equation for some discrete Lagrangian is called a *variational integrator*. As we have seen above, variational integrators can be implemented by taking two configurations q_0 and q_1 of the system, which should approximate $q_0 \approx q(t_0)$ and $q_1 \approx q(t_0 + \Delta t)$, and then solving the discrete Euler–Lagrange equations (5) for q_2. This process can then be repeated to calculate an entire discrete trajectory. The map $(q_{k-1}, q_k) \mapsto (q_k, q_{k+1})$ defined by the discrete Euler–Lagrange equations is known as the *discrete evolution map*.

Position-momentum form: For mechanical systems it is more common to specify the initial conditions as a position and a velocity (or momentum), rather than two positions. To rewrite a variational integrator in a position-momentum form we first observe that we can define the momentum at time step k to be

$$p_k = D_2 L_d(q_{k+1}, q_k, \Delta t) = -D_1 L_d(q_k, q_{k-1}, \Delta t) \tag{9}$$

The two expressions for p_k are equal because this equality is precisely the discrete Euler–Lagrange equations (5). Using this definition we can write the *position-momentum form* of a variational integrator as

$$p_k = -D_1 L_d(q_k, q_{k+1}, \Delta t) \tag{10a}$$

$$p_{k+1} = D_2 L_d(q_k, q_{k+1}, \Delta t) \tag{10b}$$

Given an initial condition (q_0, p_0) we can solve the implicit equation (10a) to find q_1, and then evaluate (10b) to give p_1. We then have (q_1, p_1) and we can repeat the procedure. The sequence $\{q_k\}_{k=0}^{N}$ so obtained will clearly satisfy the regular discrete Euler–Lagrange equations (5) for all k, due to the definition (9) of p_k.

Order of accuracy: We remarked above that a discrete Lagrangian should be thought of as approximating the continuous action integral. We will now make this statement precise. We say that a discrete Lagrangian is of *order r* if

$$L_d(q_0, q_1, \Delta t) = \int_0^{\Delta t} L(q(t), \dot{q}(t)) \, dt + \mathcal{O}(\Delta t)^{r+1} \tag{11}$$

where $q(t)$ is the unique solution of the Euler–Lagrange equations for L with $q(0) = q_0$ and $q(\Delta t) = q_1$. It can then be proven [6, Theorem 2.3.1] that if L_d is of order r then the corresponding variational integrator is also of order r, so that

$$q_k = q(k \Delta t) + \mathcal{O}(\Delta t)^{r+1}$$

To design high-order variational integrators we must therefore construct discrete Lagrangians which accurately approximate the action integral.

Symmetric methods: One useful observation when calculating the order of integrators is that symmetric methods always have even order. We say that a discrete Lagrangian is *symmetric* if

$$L_d(q_0, q_1, \Delta t) = -L_d(q_1, q_0, -\Delta t) \tag{12}$$

This implies [6, Theorem 2.4.1] that the resulting variational integrator will also be symmetric, and will thus automatically be of even order. We will use this fact below.

Geometric aside: Definition (9) of p_k defines a map $Q \times Q \to T^*Q$. In fact we can define two such maps, known as the discrete Legendre transforms, by $\mathbb{F}L_d^+(q_0, q_1) = (q_1, D_2 L_d(q_0, q_1))$ and $\mathbb{F}L_d^-(q_0, q_1, \Delta t) = (q_0, -D_1 L_d(q_0, q_1))$. The position-momentum form (10) of the discrete Euler–Lagrange equations is thus given by $\tilde{F}_{L_d}^{\Delta t} = \mathbb{F}L_d^{\pm} \circ F_{L_d}^{\Delta t} \circ (\mathbb{F}L_d^{\pm})^{-1}$ and is a map $\tilde{F}_{L_d}^{\Delta t} : T^*Q \to T^*Q$, where $F_{L_d}^{\Delta t} : Q \times Q \to Q \times Q$ is the discrete evolution map. This shows that variational integrators are really one step methods, although they may initially appear to be two step.

2.4. Examples of discrete Lagrangians

We now consider some examples of discrete Lagrangians.

Generalized mid-point rule: The classical mid-point rule for the system $\dot{x} = f(x)$ is given by $x_{k+1} - x_k = (\Delta t) f((x_{k+1} + x_k)/2)$. If we add a parameter $\alpha \in [0, 1]$ where the force evaluation occurs (so $\alpha = \frac{1}{2}$ is the standard mid-point) then we can write the corresponding discrete Lagrangian

$$L_d^{\mathrm{mp}, \alpha}(q_0, q_1, \Delta t) = (\Delta t) L \left((1 - \alpha) q_0 + \alpha q_1, \frac{q_1 - q_0}{\Delta t} \right)$$

$$= \frac{\Delta t}{2} \left(\frac{q_1 - q_0}{\Delta t} \right)^{\mathsf{T}} M \left(\frac{q_1 - q_0}{\Delta t} \right) - (\Delta t) V((1 - \alpha) q_0 + \alpha q_1) \qquad (13)$$

The discrete Euler–Lagrange equations (5) are thus

$$M \left(\frac{q_{k+1} - 2q_k + q_{k-1}}{(\Delta t)^2} \right) = -(1 - \alpha) \nabla V((1 - \alpha) q_k + \alpha q_{k+1}) - \alpha \nabla V((1 - \alpha) q_{k-1} + \alpha q_k) \quad (14)$$

and the position-momentum form (10) of the variational integrator is

$$p_k = M \left(\frac{q_{k+1} - q_k}{\Delta t} \right) + (1 - \alpha)(\Delta t) \nabla V((1 - \alpha) q_k + \alpha q_{k+1}) \qquad (15a)$$

$$p_{k+1} = M \left(\frac{q_{k+1} - q_k}{\Delta t} \right) - \alpha(\Delta t) \nabla V((1 - \alpha) q_k + \alpha q_{k+1}) \qquad (15b)$$

This is always an implicit method, and for general $\alpha \in [0, 1]$ it is first-order accurate. When $\alpha = \frac{1}{2}$ it is easy to see that $L_d^{\mathrm{mp}, \alpha}$ is symmetric, and thus the integrator is second order.

Generalized trapezoidal rule: Rather than evaluating the force at an averaged location, we could instead average the evaluated forces. Doing so at a parameter $\alpha \in [0, 1]$ gives a generalization of the trapezoidal rule

$$L_d^{\mathrm{tr}, \alpha}(q_0, q_1, \Delta t) = (\Delta t)(1 - \alpha) L \left(q_0, \frac{q_1 - q_0}{\Delta t} \right) + (\Delta t) \alpha L \left(q_1, \frac{q_1 - q_0}{\Delta t} \right)$$

$$= \frac{\Delta t}{2} \left(\frac{q_1 - q_0}{\Delta t} \right)^{\mathsf{T}} M \left(\frac{q_1 - q_0}{\Delta t} \right) - (\Delta t)((1 - \alpha) V(q_0) + \alpha V(q_1)) \qquad (16)$$

Computing the discrete Euler–Lagrange equations (5) gives

$$M\left(\frac{q_{k+1} - 2q_k + q_{k-1}}{(\Delta t)^2}\right) = -\nabla V(q_k) \tag{17}$$

with corresponding position-momentum (10) form

$$p_k = M\left(\frac{q_{k+1} - q_k}{\Delta t}\right) + (\Delta t)(1 - \alpha)\nabla V(q_k) \tag{18a}$$

$$p_{k+1} = \left(\frac{q_{k+1} - q_k}{\Delta t}\right) - (\Delta t)\alpha\nabla V(q_k) \tag{18b}$$

This method is explicit for all α, and is generally first-order accurate. For $\alpha = \frac{1}{2}$ it is symmetric, and thus becomes second-order accurate.

Observe that there is no α in the discrete Euler–Lagrange equations (17), although it does appear in the position-momentum form (18). This means that the only effect of α is on the starting procedure of this integrator, as thereafter the trajectory will be entirely determined by (17). If we are given an initial position and momentum (q_0, p_0) then we can use (18a) to calculate q_1 and then continue with (17) for future time steps. For this procedure to be second-order accurate it is necessary to take $\alpha = \frac{1}{2}$ in the use of (18a) for the first time step.

Newmark method: The Newmark family of integrators, originally given in Reference [29], are widely used in structural dynamics codes. They are usually written (see, for example, Reference [2]) for the system $L = \frac{1}{2}\dot{q}^\top M\dot{q} - V(q)$ as maps $(q_k, \dot{q}_k) \mapsto (q_{k+1}, \dot{q}_{k+1})$ satisfying the implicit relations

$$q_{k+1} = q_k + (\Delta t)\dot{q}_k + \tfrac{1}{2}(\Delta t)^2[(1 - 2\beta)a(q_k) + 2\beta a(q_{k+1})] \tag{19a}$$

$$\dot{q}_{k+1} = \dot{q}_k + (\Delta t)[(1 - \gamma)a(q_k) + \gamma a(q_{k+1})] \tag{19b}$$

$$a(q) = M^{-1}(-\nabla V(q)) \tag{19c}$$

where the parameters $\gamma \in [0, 1]$ and $\beta \in [0, \frac{1}{2}]$ specify the method. It is simple to check that the method is second order if $\gamma = \frac{1}{2}$ and first-order otherwise, and that it is generally explicit only for $\beta = 0$.

The $\beta = 0$, $\gamma = \frac{1}{2}$ case is well known to be symplectic (see, for example, Reference [30]) with respect to the canonical symplectic form Ω_L. This can be easily seen from the fact that this method is a rearrangement of the position-momentum form of the generalized trapezoidal rule with $\alpha = \frac{1}{2}$. Note that this method is the same as the velocity Verlet method, which is popular in molecular dynamics codes. As we remarked above, if the method (18) is implemented by taking one initial step with (18a) as a starting procedure, and then continued with (17), then this will give a method essentially equivalent to explicit Newmark. To be exactly equivalent, however, and to be second-order accurate, one must take $\alpha = \frac{1}{2}$ in the use of (18a). This will be of importance in Section 4.6.

It is also well known (for example, Reference [30]) that the Newmark algorithm with $\beta \neq 0$ does not preserve the *canonical* symplectic form. Nonetheless it was shown in Reference [28] that the Newmark method with $\gamma = \frac{1}{2}$ and any β can be generated from a discrete Lagrangian,

and it thus preserves a non-canonical symplectic structure. An alternative and independent method of analysing the symplectic members of Newmark has been given by Skeel *et al.* [31], including an interesting non-linear analysis in Reference [32].

Galerkin methods and symplectic Runge–Kutta schemes: Both the generalized mid-point and generalized trapezoidal discrete Lagrangians discussed above can be viewed as particular cases of linear finite element discrete Lagrangians. If we take shape functions

$$\phi_0(\alpha) = 1 - \alpha, \quad \phi_1(\alpha) = \alpha \tag{20}$$

then a general linear Galerkin discrete Lagrangian is given by

$$L_d^{G,1}(q_0, q_1, \Delta t) = \sum_{i=1}^m w_i L\left(\phi_0(\alpha_i)q_0 + \phi_1(\alpha_i)q_1, \frac{\dot{\phi}_0(\alpha_i)q_0 + \dot{\phi}_1(\alpha_i)q_1}{\Delta t}\right) \tag{21}$$

where (α_i, w_i), $i = 1, \ldots, m$, is a set of quadrature points and weights. Taking $m = 1$ and $(\alpha_1, w_1) = (\alpha, 1)$ gives the generalized mid-point rule, while taking $m = 2$, $(\alpha_1, w_1) = (0, 1 - \alpha)$ and $(\alpha_2, w_2) = (1, \alpha)$ gives the generalized trapezoidal rule.

Taking high-order finite element basis functions and quadrature rules is one method to construct high-order variational integrators. In general, we have a set of basis functions ϕ_j, $j = 0, \ldots, s$, and a set of quadrature points (α_i, w_i), $i = 1, \ldots, m$. The resulting Galerkin discrete Lagrangian is then

$$L_d^{G,s,\text{full}}(q_0, \ldots, q_s, \Delta t) = \sum_{i=1}^m w_i L\left(\sum_{j=0}^s \phi_j(\alpha_i)q_j, \frac{1}{\Delta t}\sum_{j=0}^s \dot{\phi}_j(\alpha_i)q_j\right) \tag{22}$$

This $(s + 1)$-point discrete Lagrangian can be used to derive a standard two-point discrete Lagrangian by taking

$$L_d^{G,s}(q_0, q_1, \Delta t) = \underset{Q_1,\ldots,Q_{s-1}}{\text{ext}} L_d^{G,s,\text{full}}(q_0, Q_1, \ldots, Q_{s-1}, q_1, \Delta t) \tag{23}$$

where $\text{ext}\, L_d^{G,s,\text{full}}$ means that $L_d^{G,s,\text{full}}$ should be evaluated at extreme or critical values of $Q_1, \ldots, Q_s$. When $s = 1$ we immediately recover (21). Of course, using the discrete Lagrangian (22) is equivalent to a finite element discretization in time of (1), as in Reference [33] for example.

An interesting feature of Galerkin discrete Lagrangians is that the resulting variational integrator can always be implemented as a partitioned Runge–Kutta method (see Reference [6] for details). Using this technique high-order implicit methods can be constructed, including the collocation Gauss–Legendre family and the Lobatto IIIA–IIIB family of integrators.

2.5. Constrained systems

Many physical systems can be most easily expressed by taking a larger system and imposing constraints, which we take here to mean requiring that a given constraint function g is zero for all configurations. To discretize such problems, we can either work in local co-ordinates on the constraint set $\{q \mid g(q) = 0\}$, or we can work with the full configurations q and use

Lagrange multipliers to enforce $g(q) = 0$. Here we consider the second option, as the first option requires no modification to the variational integrator theory.[§]

Taking variations of the action with Lagrange multipliers added requires that

$$\delta \sum_{k=0}^{N-1} [L_d(q_k, q_{k+1}, h) + \lambda_{k+1} g(q_{k+1})] = 0 \tag{24}$$

and so using (4) gives the *constrained discrete Euler–Lagrange equations*

$$D_2 L_d(q_{k-1}, q_k, h) + D_1 L_d(q_k, q_{k+1}, h) = -\lambda_k \cdot \nabla g(q_k) \tag{25a}$$

$$g(q_{k+1}) = 0 \tag{25b}$$

which can be solved for λ_k and q_{k+1}.[¶] These equations have all of the same conservation properties, such as symplecticity and momentum conservation, as the unconstrained discrete equations.

An interesting example of a constrained variational integrator is the SHAKE method [12], which can be neatly obtained by taking the generalized trapezoidal rule of Section 2.4 with $\alpha = \frac{1}{2}$ and forming the constrained equations as in (25).

2.6. Forcing and dissipation

Now we consider non-conservative systems; those with forcing and those with dissipation. For problems in which the non-conservative forcing dominates there is likely to be little benefit from variational integration techniques. There are many problems, however, for which the system is primarily conservative, but where there are very weak non-conservative effects which must be accurately accounted for. Examples include weakly damped systems, such as photonic drag on satellites, and small control forces, as arise in continuous thrust technologies for spacecraft. In applications such as these the conservative behaviour of variational integrators can be very important, as they do not introduce numerical dissipation in the conservative part of the system, and thus accurately resolve the small non-conservative forces.

Recall that the (continuous) *integral Lagrange–d'Alembert principle* is

$$\delta \int L(q(t), \dot{q}(t)) \, \mathrm{d}t + \int F(q(t), \dot{q}(t)) \delta q \, \mathrm{d}t = 0 \tag{26}$$

where $F(q, v)$ is an arbitrary force function. We define the *discrete Lagrange–d'Alembert principle* to be

$$\delta \sum L_d(q_k, q_{k+1}) + \sum [F_d^-(q_k, q_{k+1}) \delta q_k + F_d^+(q_k, q_{k+1}) \delta q_{k+1}] = 0 \tag{27}$$

[§]In the event that the constraint set is not a vector space, local co-ordinates would require the more general theory of discrete mechanics on smooth manifolds, as in Reference [6].

[¶]Observe that the linearization of the above system is not symmetric, unlike for constrained elliptic problems. This is because we are solving forward in time, rather than for all times at once as in a boundary value problem.

where L_d is the discrete Lagrangian and F_d^- and F_d^+ are the left and right discrete forces. These should approximate the continuous forcing so that

$$F_d^-(q_k, q_{k+1})\delta q_k + F_d^+(q_k, q_{k+1})\delta q_{k+1} \approx \int_{t_k}^{t_{k+1}} F(q(t), \dot{q}(t))\delta q \, \mathrm{d}t$$

Equation (27) defines an integrator $(q_k, q_{k+1}) \mapsto (q_{k+1}, q_{k+2})$ given implicitly by the *forced discrete Euler–Lagrange equations*:

$$D_1 L_d(q_{k+1}, q_{k+2}) + D_2 L_d(q_k, q_{k+1}) + F_d^-(q_{k+1}, q_{k+2}) + F_d^+(q_k, q_{k+1}) = 0 \qquad (28)$$

The simplest example of discrete forces is to take

$$F_d^-(q_k, q_{k+1}) = F(q_k)$$

$$F_d^+(q_k, q_{k+1}) = 0$$

which, together with the discrete Lagrangian (3), gives the forced Euler–Lagrange equations

$$M\left(\frac{q_{k+1} - 2q_k + q_{k-1}}{h^2}\right) = -\nabla V(q_k) + F(q_k)$$

The position-momentum form of a variational integrator with forcing is useful for implementation purposes. This is given by

$$p_k = -D_1 L_d(q_k, q_{k+1}) - F_d^-(q_k, q_{k+1})$$

$$p_{k+1} = D_2 L_d(q_k, q_{k+1}) + F_d^+(q_k, q_{k+1})$$

As an example of a variational integrator applied to a non-conservative system, in Figure 3 we plot the energy evolution of a Lagrangian system with dissipation added, which is simulated using a low-order variational integrator with forcing, as in (28), and a standard high-order Runge–Kutta method. Despite the disadvantage of being low order, the variational method tracks the error decay more accurately as it does not artificially dissipate energy for stability purposes.

3. CONSERVATION PROPERTIES OF VARIATIONAL INTEGRATORS

3.1. Noether's theorem and momentum conservation

One of the important features of variational systems is that symmetries of the system lead to momentum conservation laws of the Euler–Lagrange equations, a classical result known as Noether's theorem.

Consider a one-parameter group of curves $q^\varepsilon(t)$, with $q^0(t) = q(t)$, which have the property that $L(q^\varepsilon(t), \dot{q}^\varepsilon(t)) = L(q(t), \dot{q}(t))$ for all ε. When the Lagrangian is invariant in this manner
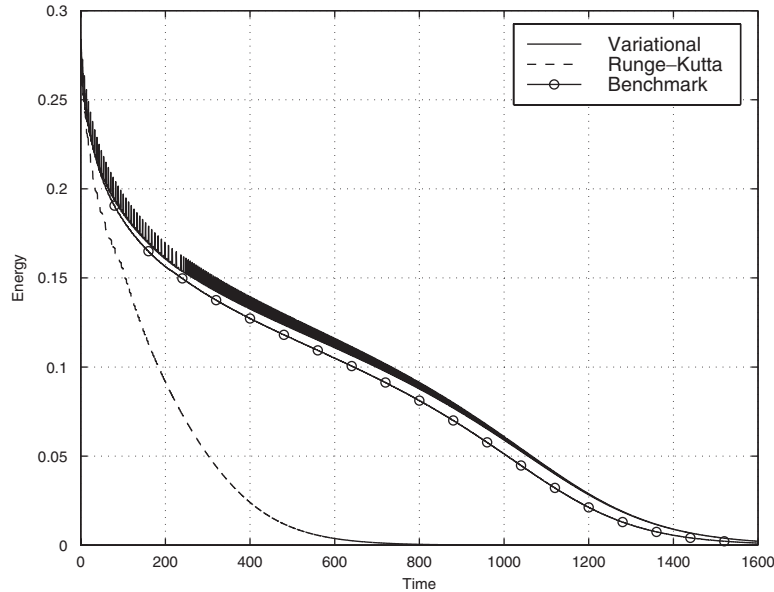
Figure 3. Energy evolution for a dissipative mechanical system, for a second-order variational integrator and a fourth-order Runge–Kutta method. The benchmark solution is a very expensive and accurate simulation. Observe that the variational method correctly captures the rate of decay of the energy, unlike the Runge–Kutta method.

then we have a symmetry of the system, and we write

$$\xi(t) = \left.\frac{\partial q^\varepsilon(t)}{\partial \varepsilon}\right|_{\varepsilon=0} \tag{29}$$

for the infinitesimal symmetry direction.

The fact that the Lagrangian is invariant means that the action integral is also invariant, so its derivative with respect to ε will be zero. If $q(t)$ is a solution trajectory then we can set the Euler–Lagrange term in Equation (1) to zero to obtain

$$0 = \left.\frac{\partial}{\partial \varepsilon}\right|_{\varepsilon=0} \int_0^T L(q(t), \dot{q}(t))\, \mathrm{d}t = \frac{\partial L}{\partial \dot{q}}(q(T), \dot{q}(T))\xi(T) - \frac{\partial L}{\partial \dot{q}}(q(0), \dot{q}(0))\xi(0) \tag{30}$$

The terms on the right-hand side above are the final and initial momenta in the direction ξ, which are thus equal. This is the statement of *Noether's theorem*.

As examples, consider the one-parameter groups $q^\varepsilon(t) = q(t) + \varepsilon v$ and $q^\varepsilon(t) = \exp(\varepsilon\Omega)q(t)$ for any vector v and skew-symmetric matrix Ω. The transformations give translations and rotations, respectively, and evaluating (30) for these cases gives conservation of linear and angular momentum, assuming that the Lagrangian is indeed invariant under these transformations.

Geometric aside: More generally, we may consider an arbitrary Lie group G, with Lie algebra $\mathfrak{g}$, rather than the one-dimensional groups taken above. The analogue of $\xi(t)$ is then the infinitesimal generator $\xi_Q: Q \to TQ$, for any $\xi \in \mathfrak{g}$, corresponding to an action of G on

Q whose lift to TQ leaves L invariant. Equation (30) then becomes $(\partial L/\partial\dot{q})\xi_Q|_0^T = 0$, which means that the momentum map $J_L: TQ \to \mathfrak{g}^*$ is conserved, where $J_L(q, \dot{q})\xi = (\partial L/\partial\dot{q})\xi_Q(q)$. While we must generally take many one-parameter groups, such as translations by any vector v, to show that a quantity such as linear momentum is conserved, with this general framework we can take $\mathfrak{g}$ to be the space of all v's, and thus obtain conservation of linear momentum with only a single group, albeit multidimensional.

3.2. Discrete time Noether's theorem and discrete momenta

A particularly nice feature of the variational derivation of momentum conservation is that we simultaneously derive both the expression for the conserved quantity and the theorem that it is conserved. Using the variational derivation in the discrete time case, we can thus obtain the definition of discrete time momenta, as well as a discrete time Noether's theorem implying that they are conserved.

Take a one-parameter group of discrete time curves $\{q_k^\varepsilon\}_{k=0}^N$, with $q_k^0 = q_k$, such that $L_d(q_k^\varepsilon, q_{k+1}^\varepsilon) = L_d(q_k, q_{k+1})$ for all ε and k. The infinitesimal symmetry for such an invariant discrete Lagrangian is written

$$\xi_k = \left.\frac{\partial q_k^\varepsilon}{\partial\varepsilon}\right|_{\varepsilon=0} \tag{31}$$

Invariance of the discrete Lagrangian implies invariance of the action sum, and so its ε derivative will be zero. Assuming that $\{q_k\}$ is a solution trajectory, then (4) becomes

$$0 = \left.\frac{\partial}{\partial\varepsilon}\right|_{\varepsilon=0} \sum_{k=0}^{N-1} L_d(q_k^\varepsilon, q_{k+1}^\varepsilon) = D_1 L_d(q_0, q_1)\xi_0 + D_2 L_d(q_{N-1}, q_N)\xi_N \tag{32}$$

Observing that $0 = D_1 L_d(q_0, q_1)\xi_0 + D_2 L_d(q_0, q_1)\xi_1$ as L_d is invariant, we thus have the *discrete Noether's theorem*

$$D_2 L_d(q_{N-1}, q_N)\xi = D_2 L_d(q_0, q_1)\xi \tag{33}$$

where the discrete momentum in the direction ξ is given by $D_2 L_d(q_k, q_{k+1})\xi$.

Consider the example discrete Lagrangian (13) with $\alpha = 0$, and assume that $q \in Q \equiv \mathbb{R}^3$ and that V is a function of the norm of q only. This is the case of a particle in a radial potential for example. Then the discrete Lagrangian is invariant under rotations $q_k^\varepsilon = \exp(\varepsilon\Omega)q_k$, for any skew-symmetric matrix $\Omega \in \mathbb{R}^{3\times3}$. Evaluating (33) in this case gives

$$q_N \times M\left(\frac{q_N - q_{N-1}}{t_N - t_{N-1}}\right) = q_1 \times M\left(\frac{q_1 - q_0}{t_1 - t_0}\right) \tag{34}$$

We have thus computed the correct expressions for the discrete angular momentum, and shown that it is conserved. Note that while this expression may seem obvious, in more complicated examples this will not be the case.

Geometric aside: As in the continuous case, we can extend the above derivation to multidimensional groups and define a full discrete momentum map $J_{L_d}: Q \times Q \to \mathfrak{g}^*$ by $J_L(q_0, q_1)\xi = D_2 L_d(q_0, q_1)\xi_Q(q_1)$. In fact there are two discrete momentum maps, corresponding to $D_1 L_d$ and $D_2 L_d$, but they are equal whenever L_d is invariant.

3.3. Continuous time symplecticity

In addition to the conservation of energy and momenta, Lagrangian mechanical systems also conserve another quantity known as a symplectic bilinear form.

Consider a two-parameter set of initial conditions $(q_0^{\varepsilon,v}, v_0^{\varepsilon,v})$ so that $(q^{\varepsilon,v}(t), v^{\varepsilon,v}(t))$ is the resulting trajectory of the system. The corresponding variations are denoted

$$\delta q_1^\varepsilon(t) = \frac{\partial}{\partial v} q^{\varepsilon,v}(t)\bigg|_{v=0} \qquad \delta q_2^v(t) = \frac{\partial}{\partial \varepsilon} q^{\varepsilon,v}(t)\bigg|_{\varepsilon=0} \qquad \delta^2 q(t) = \frac{\partial}{\partial \varepsilon} \frac{\partial}{\partial v} q^{\varepsilon,v}(t)\bigg|_{\varepsilon,v=0}$$

and we write $\delta q_1(t) = \delta q_1^0(t)$, $\delta q_2(t) = \delta q_2^0(t)$ and $q^\varepsilon(t) = q^{\varepsilon,0}(t)$. We now compute the second derivative of the action integral to be

$$\frac{\partial}{\partial \varepsilon}\bigg|_{\varepsilon=0} \frac{\partial}{\partial v}\bigg|_{v=0} S(q^{\varepsilon,v}) = \frac{\partial}{\partial \varepsilon}\bigg|_{\varepsilon=0} (DS(q^\varepsilon)\delta q_1^\varepsilon)$$

$$= \frac{\partial}{\partial \varepsilon}\bigg|_{\varepsilon=0} \left(\frac{\partial L}{\partial v^i}(\delta q_1^\varepsilon)^i(T) - \frac{\partial L}{\partial v^i}(\delta q_1^\varepsilon)^i(0) \right)$$

$$= \frac{\partial^2 L}{\partial q^j \partial v^i} \delta q_1^i(T)\delta q_2^j(T) + \frac{\partial^2 L}{\partial v^j \partial v^i} \delta q_1^i(T)\delta \dot{q}_2^j(T) + \frac{\partial L}{\partial v^i} \delta^2 q^i(T)$$

$$- \frac{\partial^2 L}{\partial q^j \partial v^i} \delta q_1^i(0)\delta q_2^j(0) - \frac{\partial^2 L}{\partial v^j \partial v^i} \delta q_1^i(0)\delta \dot{q}_2^j(0) - \frac{\partial L}{\partial v^i} \delta^2 q^i(0)$$

Here and subsequently, repeated indices in a product indicate a sum over the index range. If we reverse the order of differentiation with respect to ε and v, then by symmetry of mixed partial derivatives we will obtain an equivalent expression. Subtracting this from the above equation then gives

$$\frac{\partial^2 L}{\partial q^j \partial v^i} \left[\delta q_1^i(T)\delta q_2^j(T) - \delta q_2^i(T)\delta q_1^j(T) \right] + \frac{\partial^2 L}{\partial v^j \partial v^i} \left[\delta q_1^i(T)\delta \dot{q}_2^j(T) - \delta q_2^i(T)\delta \dot{q}_1^j(T) \right]$$

$$= \frac{\partial^2 L}{\partial q^j \partial v^i} \left[\delta q_1^i(0)\delta q_2^j(0) - \delta q_2^i(0)\delta q_1^j(0) \right] + \frac{\partial^2 L}{\partial v^j \partial v^i} \left[\delta q_1^i(0)\delta \dot{q}_2^j(0) - \delta q_2^i(0)\delta \dot{q}_1^j(0) \right]$$

Each side of this expression is an antisymmetric bilinear form evaluated on the variations δq_1 and δq_2. The fact that this evaluation gives the same result at $t = 0$ and at $t = T$ implies that the bilinear form itself is preserved by the Euler–Lagrange equations. This bilinear form is called the *symplectic form* of the system, and the fact that it is preserved is called symplecticity of the flow of the Euler–Lagrange equations.

This is a conservation property in the same way that momenta and energy are conservation properties of Lagrangian mechanical systems, and it has a number of important consequences. Examples of this include Liouville's theorem, which states that phase space volume is preserved by the time evolution of the system, and four-fold symmetry of the eigenvalues of linearizations of the system, so that if λ is an eigenvalue, so too are $-\lambda$, $\bar{\lambda}$ and $-\bar{\lambda}$. There are many other important examples; see Reference [34].

Geometric aside: The above derivation can be written using differential geometric notation as follows. The boundary terms in the action variation equation (1) are intrinsically given by $\Theta_L = (\mathbb{F}L)^*\Theta$, the pullback under the Legendre transform of the canonical one-form $\Theta = p_i\,\mathrm{d}q^i$ on T^*Q. We thus have $\mathrm{d}S = (F_L^t)^*\Theta_L - \Theta_L$ and so using $d^2 = 0$ (which is the intrinsic statement of symmetry of mixed partial derivatives) we obtain $0 = d^2S = (F_L^t)^*(\mathrm{d}\Theta_L) - \mathrm{d}\Theta_L$. The symplectic two form above is thus $\Omega_L = -\mathrm{d}\Theta_L$, and we recover the usual statement of symplecticity of the flow F_L^t for Lagrangian systems.

3.4. Discrete time symplecticity

As we have seen above, symplecticity of continuous time Lagrangian systems is a consequence of the variational structure. There is thus an analogous property of discrete Lagrangian systems.

Consider a two-parameter set of initial conditions $(q_0^{\varepsilon,v}, q_1^{\varepsilon,v})$ and let $\{q_k^{\varepsilon,v}\}_{k=0}^N$ be the resulting discrete trajectory. We denote the corresponding variations by

$$\delta q_k^\varepsilon = \left.\frac{\partial}{\partial v}q_k^{\varepsilon,v}\right|_{v=0} \qquad \delta\bar{q}_k^v = \left.\frac{\partial}{\partial\varepsilon}q_k^{\varepsilon,v}\right|_{\varepsilon=0} \qquad \delta^2 q_k = \left.\frac{\partial}{\partial\varepsilon}\frac{\partial}{\partial v}q_k^{\varepsilon,v}\right|_{\varepsilon,v=0}$$

and we write $\delta q_k = \delta q_k^0$, $\delta\bar{q}_k = \delta\bar{q}_k^0$ and $q_k^\varepsilon = q^{\varepsilon,0}$ for $k = 0,\ldots,N$. The second derivative of the action sum is thus given by

$$\begin{aligned}\left.\frac{\partial}{\partial\varepsilon}\right|_{\varepsilon=0}\left.\frac{\partial}{\partial v}\right|_{v=0}S_d(\{q_k^{\varepsilon,v}\}) &= \left.\frac{\partial}{\partial\varepsilon}\right|_{\varepsilon=0}(DS_d(\{q_k^\varepsilon\})\delta q^\varepsilon)\\[2mm]
&= \left.\frac{\partial}{\partial\varepsilon}\right|_{\varepsilon=0}(D_{1i}L_d(q_0^\varepsilon, q_1^\varepsilon)(\delta q_0^\varepsilon)^i + D_{2i}L_d(q_{N-1}^\varepsilon, q_N^\varepsilon)(\delta q_N^\varepsilon)^i)\\[2mm]
&= D_{1j}D_{1i}L_d(q_0, q_1)\delta q_0^i\delta\bar{q}_0^j + D_{2j}D_{1i}L_d(q_0, q_1)\delta q_0^i\delta\bar{q}_1^j\\[2mm]
&\quad + D_{1j}D_{2i}L_d(q_{N-1}, q_N)\delta q_N^i\delta\bar{q}_{N-1}^j + D_{2j}D_{2i}L_d(q_{N-1}, q_N)\delta q_N^i\delta\bar{q}_N^j\\[2mm]
&\quad + D_{1i}L_d(q_0, q_1)\delta^2 q_0^i + D_{2i}L_d(q_{N-1}, q_N)\delta^2 q_N^i\end{aligned}\tag{35}$$

By symmetry of mixed partial derivatives, reversing the order of differentiation above will give an equivalent expression. Subtracting one from the other will thus give zero, and rearranging the resulting equation we obtain

$$D_{1j}D_{2i}L_d(q_{N-1}, q_N)\left[\delta q_N^i\delta\bar{q}_{N-1}^j - \delta\bar{q}_N^i\delta q_{N-1}^j\right] = D_{2j}D_{1i}L_d(q_0, q_1)\left[\delta\bar{q}_0^i\delta q_1^j - \delta q_0^i\delta\bar{q}_1^j\right]\tag{36}$$

We can also repeat the derivation in (35) and (36) for $L_d(q_0^{\varepsilon,v}, q_1^{\varepsilon,v})$, rather than the entire action sum, to obtain

$$D_{2j}D_{1i}L_d(q_0, q_1)\left[\delta\bar{q}_0^i\delta q_1^j - \delta q_0^i\delta\bar{q}_1^j\right] = D_{1j}D_{2i}L_d(q_0, q_1)\left[\delta q_1^i\delta\bar{q}_0^j - \delta\bar{q}_1^i\delta q_0^j\right]\tag{37}$$

which can also be directly seen from the symmetry of mixed partial derivatives. Substituting this into (36) now gives

$$D_{1j}D_{2i}L_d(q_{N-1}, q_N)\left[\delta q_N^i\delta\bar{q}_{N-1}^j - \delta\bar{q}_N^i\delta q_{N-1}^j\right] = D_{1j}D_{2i}L_d(q_0, q_1)\left[\delta q_1^i\delta\bar{q}_0^j - \delta\bar{q}_1^i\delta q_0^j\right]\tag{38}$$
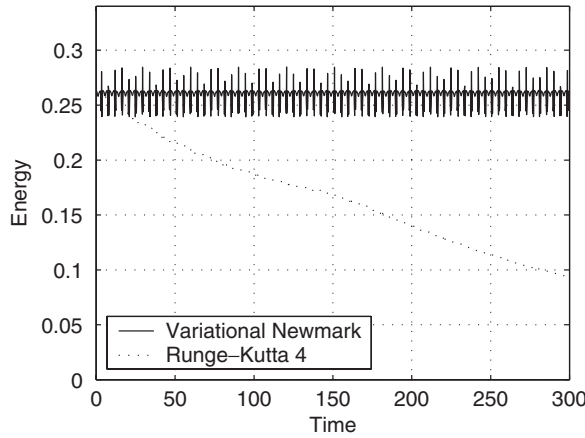
Figure 4. Energy computed with variational second-order Newmark and fourth-order Runge–Kutta. Note that the variational method does not artificially dissipate energy.

We can now see that each side of this equation is an antisymmetric bilinear form, which we call the *discrete symplectic form*, evaluated on the variations δq_k and $\delta\bar{q}_k$. The two sides give this expression at the first time step and the final time step, so we have that the discrete symplectic form is preserved by the time evolution of the discrete system.

In the next section, we will consider some numerical consequences of this property.

Geometric aside: Intrinsically we can identify two one-forms $\Theta^+_{L_d} = D_2 L_d \, \mathrm{d}q_1$ and $\Theta^-_{L_d} = D_1 L_d \, \mathrm{d}q_0$, so that $\mathrm{d}S_d = (F^N_{L_d})^* \Theta^+_{L_d} + \Theta^-_{L_d}$. Using $d^2 = 0$ (symmetry of mixed partial derivatives) gives $0 = d^2 S_d = (F^N_{L_d})^* (\mathrm{d}\Theta^+_{L_d}) + \mathrm{d}\Theta^-_{L_d}$ and so defining the discrete symplectic two forms $\Omega^\pm_{L_d} = -\mathrm{d}\Theta^\pm_{L_d}$ gives $(F^N_{L_d})^* \Omega^+_{L_d} = -\Omega^-_{L_d}$, which is the intrinsic form of (36). However, we observe that $0 = d^2 L_d = \mathrm{d}(\Theta^+_{L_d} + \Theta^-_{L_d}) = -\Omega^+_{L_d} - \Omega^-_{L_d}$ and hence $\Omega^+_{L_d} = -\Omega^-_{L_d}$, which is (37). Combining this with our previous expression then gives $(F^N_{L_d})^* \Omega^+_{L_d} = \Omega^+_{L_d}$ as the intrinsic form of (38), which is a discrete symplecticity of the evolution.

Observe that using the discrete Legendre transforms we have $\Theta^\pm_{L_d} = (\mathbb{F}L^\pm_d)^* \Theta$, where $\Theta = p_i \, \mathrm{d}q^i$ is the canonical one-form on T^*Q. Expression (38) thus shows that the map $\mathbb{F}L^+_d \circ F_{L_d} \circ (\mathbb{F}L^+_d)^{-1}$ preserves the canonical symplectic two-form Ω on T^*Q. Variational integrators are thus symplectic methods in the standard sense.

3.5. Backward error analysis

We now briefly consider why preservation of a symplectic form may be advantageous numerically. We first consider a numerical example, and then the theory which explains it.

Approximate energy conservation: If we use a variational method to simulate a non-linear model system and plot the energy versus time, then we obtain a graph like that in Figure 4. For comparison, this graph also shows the energy curve for a simulation with a standard method such as RK4 (the common fourth-order Runge–Kutta method).

The system being simulated here is purely conservative and so there should be no loss of energy over time. The striking aspect of this graph is that while the energy associated with a
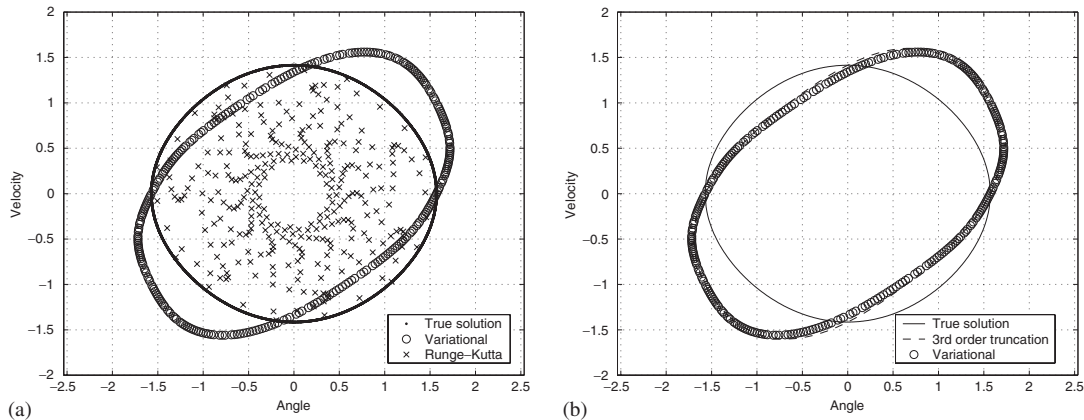
Figure 5. Phase space plots of: (a) symplectic method compared to non-symplectic Runge–Kutta method for the pendulum system; and (b) third-order truncation of the modified energy.

standard method decays due to numerical damping, for the Newmark method the energy error remains bounded. This may be understood by recognizing that the integrator is symplectic, that is, it preserves the same two form on state space as the true system.

Backward error analysis: To understand the above numerics it is necessary to use the concept of backward error analysis, whereby we construct a *modified system* which is a model for the numerical method. Here we only give a simple outline of the procedure; for details and proofs see References [19, 35].

Consider an ODE $\dot{x} = f(x)$ and an integrator $x_{k+1} = F(x_k)$ which approximates the evolution of f. For a given initial condition x_0 let $x(t)$ be the true solution of $\dot{x} = f(x)$ and let $\{x_k\}_{k=0}^{N}$ be the discrete time approximation generated by F. Now let $\dot{\bar{x}} = \bar{f}(\bar{x})$ be a second ODE, called the *modified system*,$^{\|}$ for which the resulting trajectory $\bar{x}(t)$ exactly samples the points x_k, so that $\bar{x}(k\Delta t) = x_k$. We say that the *backward error* of F to f is the error $\|\bar{f} - f\|$, measured in an appropriate norm. This contrasts with the usual *forward error* of F to f given by $\|x_k - x(k\Delta t)\|$, which is also known as trajectory error.

We can now understand why variational integrators are different to standard methods. Namely, their modified systems are Lagrangian systems.[**] That is, for a given Lagrangian L, a variational integrator is exactly solving a system $\bar{L}$, which is close to L. This means that the discrete trajectory (which is exactly sampling the trajectory of $\bar{L}$) has all of the properties of a conservative mechanical system, such as energy conservation.

In the particular case of energy, we see in Figure 5(a) the phase space portrait of the exact solution of the non-linear pendulum equations (a conservative Lagrangian system), together with

---

$^{\|}$In fact for most integrators F there will not be an exact modified system $\bar{f}$, but instead there will only be a system such that $\bar{x}(k\Delta t) = x_k + C\exp(-1/(\Delta t))$, so the modified system is said to be *exponentially close* to the discrete time integrator. For Δt sufficiently small this is effectively as good as a true modified system.
[**]This is typically done in the literature on the Hamiltonian side, by showing that symplectic integrators have Hamiltonian modified systems. As all variational integrators are symplectic and regular Hamiltonian systems are Lagrangian the result follows.

the discrete time trajectories of a variational integrator and a standard Runge–Kutta method. Clearly the Runge–Kutta method is dissipative, and so its trajectory limits to the origin. In contrast, the variational integrator has an exactly conserved energy (the energy of the modified system $\bar{L}$) and so it remains on its level set for all time.

While it is not generally possible to calculate the modified system exactly, it is possible to find $\mathcal{O}(\Delta t)^r$ truncations of it. In Figure 5(b) the energy level set of the third-order truncation of the modified equation is plotted, and we see that indeed the variational integrator remains almost on it. This also explains why the energy plots for variational integrators contain a typical oscillation about the true energy. The modified energy level set will be close to the true energy level set everywhere, but it will typically be inside it at some locations and outside it at others. As the discrete trajectory evolves on the modified energy level set it thus has a non-modified energy which is sometimes greater than and sometimes less that the true value, but which never deviates very far. This results in the characteristic energy oscillation.

4. ASYNCHRONOUS VARIATIONAL INTEGRATORS (AVIs)

In this section, we describe a class of asynchronous time integrators.

4.1. Systems of particles

Consider a system of N_a particles with dynamics described by a Lagrangian of the form

$$L(x, \dot{x}) = \sum_a \tfrac{1}{2} m_a \|\dot{x}_a\|^2 - \sum_K V_K(x) \tag{39}$$

where m_a is the mass of particle a at position $x_a \in \mathbb{R}^3$ and V_K is the potential energy of subsystem K. It is this decomposition of the total potential energy which permits the asynchronicity of the time discretization.

Example (*Molecular dynamics*): One common example of systems of form (39) arises in molecular problems. Here the potentials V_K represent the interaction between pairs (or triples) of particles, so that we have $V_{ab} = V(\|x_b - x_a\|)$ for all $a<b$ and some given $V(r)$.

Example (*Finite elements*): Continuum mechanical systems can be specified by considering the deformation mapping $\varphi : \mathcal{B} \to \mathbb{R}^3$ which acts on points X in the reference configuration $\mathcal{B} \subset \mathbb{R}^3$. The deformation of local or infinitesimal neighbourhoods is described by the deformation gradient $F = \nabla_X \varphi$, where $\nabla_X \varphi$ denotes only the spatial gradient of φ. The time derivative of φ is denoted with $\dot{\varphi}$, while accelerations are indicated with $\ddot{\varphi}$. The Lagrangian is of the form[††]

$$L(\varphi, \dot{\varphi}) = \int_{\mathcal{B}} \left(\frac{\rho_0}{2} \|\dot{\varphi}\|^2 - W(\nabla_X \varphi, X) \right) dX \tag{40}$$

which describes the dynamics of hyperelastic materials. In Equation (40) ρ_0 is the density of the continuum in the reference configuration and $W(F, X)$ is the free-energy density of the material. We assume that $W(F, X)$ satisfies the requirement of material frame indifference (see,

---

[††]Of course terms representing body forces arising from a potential energy could be added to the Lagrangian density (40). Similarly, the Lagrangian density could depend explicity on time. These cases do not present any essential difficulties and can be found Reference [19]. For simplicity, we will not consider them here.

e.g. Reference [36]). Recall that for hyperelastic materials the first Piola–Kirchhoff stress tensor is given by

$$P_I^i = \frac{\partial}{\partial F_I^i} W(F, X) \tag{41}$$

The Euler–Lagrange equations for (40) translate into the infinitesimal balance of linear momentum

$$\nabla_X \cdot P - \rho_0 \ddot{\varphi} = 0 \tag{42}$$

Consider now the spatial semidiscretization of (40) by finite elements. We let $\mathscr{T}$ be a triangulation of $\mathscr{B}$. The corresponding finite-dimensional space of finite element solutions consists of deformation mappings of the form

$$\varphi_h(X) = \sum_{a \in \mathscr{T}} x_a N_a(X) \tag{43}$$

where N_a is the shape function corresponding to node a and x_a represents the position of the node in the deformed configuration.

We let m_a be the lumped mass associated to node a arising from discretization (43) and some lumping scheme. The elemental potential energies $V_K(x_K)$ are given by

$$V_K(x_K) = \int_K W(\nabla \varphi_h, X) \, dX \tag{44}$$

where x_K is the vector of positions of all the nodes in element K. Of course, terms in the potential energy resulting from boundary conditions or body forces should be added to expression in (44). We thus have a finite-dimensional system in the form of (39).

4.2. Asynchronous time discretizations

While asynchronous methods can be applied to any Lagrangian system with a decomposed potential energy, here we use the terminology associated with finite element discretizations. To apply the methods to a problem such as molecular dynamics, simply think of 'elements' as being 'pairwise potentials', and 'nodes' as being 'particles'.

Time discretization: For each element K we choose a time step Δt_K. Different elements could have different time steps.[‡‡] We denote with $t_K^j = j \Delta t_K$, $j = 0, \ldots, N_K$, the time of the jth time step for element K, with N_K being the smallest integer such that $t_K^j \geqslant T$. From this, we construct the spacetime discretization as shown in Figure 6. As shown in Figure 6, we also denote with t_a^i, $i = 0, \ldots, N_a$, the time instant for node a. Non-constant time steps per element are considered later in Section 6.2.

[‡‡]It is straightforward to extend the definition of AVI methods given here to allow each elemental time step to vary as the integration advances in time, either in response to error driven adaptivity criteria, or to some other information such as local energy balance laws (see Section 6.2).
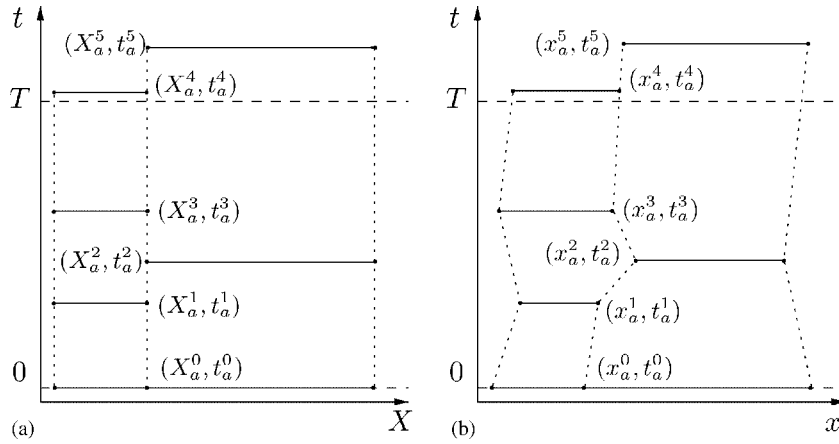
Figure 6. Spacetime diagram of the motion of a two-element, one-dimensional mesh. The set of co-ordinates and times for a single node is shown in the reference and deformed configurations. Note that the nodal co-ordinates and times are labelled according to the interaction of the node with all elements to which it belongs: (a) reference configuration; and (b) deformed configuration.

Discrete action sum: Define the discrete action sum to be

$$S_d = \sum_a \sum_{i=0}^{N_a-1} \frac{1}{2} m_a (t_a^{i+1} - t_a^i) \left\| \frac{x_a^{i+1} - x_a^i}{t_a^{i+1} - t_a^i} \right\|^2 - \sum_K \sum_{j=0}^{N_K-1} (t_K^{j+1} - t_K^j) \, V_K(x_K^{j+1}) \tag{45}$$

which approximates the continuous action over the time interval $[0, T]$.

Discrete Lagrangians: In the particular case of finite element discretizations, the above action sum can be realized as the sum of spacetime discrete Lagrangians. Let $m_{K,a}$ be the mass of node a which is due to element K, so that $m_a = \sum_K m_{K,a}$. Then the discrete action is the sum over all elements K and elemental times j of the discrete Lagrangian

$$L_K^j = \sum_{a \in K} \sum_{\{i:\, t_K^j \leqslant t_a^i < t_K^{j+1}\}} \frac{1}{2} m_{K,a} (t_a^{i+1} - t_a^i) \left\| \frac{x_a^{i+1} - x_a^i}{t_a^{i+1} - t_a^i} \right\|^2 - (t_K^{j+1} - t_K^j) \, V_K(x_K^{j+1}) \tag{46}$$

where x_K is the vector of positions of all the nodes in element K. The discrete Lagrangian L_K^j approximates the incremental action of element K over the interval $[t_K^j, t_K^{j+1}]$. In general, L_K^j depends on the nodal positions x_a^i, $a \in K$ and i such that $t_a^i \in [t_K^j, t_K^{j+1}]$.[§§] The discrete action sum S_d thus depends on all the nodal positions x_a^i for all nodes $a \in \mathcal{T}$ and for all times t_a^i, $0 \leqslant i \leqslant N_a$.

Discrete Euler–Lagrange equations: The discrete variational principle states that the discrete trajectory of the system should be a critical point of the action sum for all admissible variations

---

[§§]If adjacent elements possess coincident elemental times one must derive the correct interpretation by taking the appropriate limits. See Reference [19, Section 3.2].

of the nodal co-ordinates x_a^i. The discrete Euler–Lagrange equations are

$$D_a^i S_d = 0 \tag{47}$$

for all $a \in \mathscr{T} \setminus \partial_d \mathscr{B}$ such that $t_a^i \in (0, T)$. Here and subsequently, D_a^i denotes differentiation with respect to x_a^i. The discrete Euler–Lagrange equations (47) define the equations of motion of the discrete problem.

After introducing the definition of the discrete Lagrangian (46) under consideration into (47), a straightforward calculation gives the discrete Euler–Lagrange equations explicitly in the form

$$p_a^{i+1/2} - p_a^{i-1/2} = I_a^i \tag{48}$$

where

$$p_a^{i+1/2} \equiv m_a \frac{x_a^{i+1} - x_a^i}{t_a^{i+1} - t_a^i} \equiv m_a v_a^{i+1/2} \tag{49}$$

are discrete linear momenta and m_a are the nodal masses, i.e. $m_a = \sum_K m_{K,a}$. In addition, we define

$$I_K^j \equiv -(t_K^j - t_K^{j-1}) \frac{\partial}{\partial x_K^j} V_K(x_K^j) \tag{50}$$

which may be regarded as the impulses exerted by element K on its nodes at time t_K^j. In Equation (48) I_a^i represents the component of I_K^j corresponding to node a, with $t_a^i = t_K^j$. Equation (48) may be interpreted as describing a sequence of percussions imparted by the elements on their nodes at discrete instants of time. Thus, the element K accumulates and stores impulses I_K^j over the time interval (t_K^{j-1}, t_K^j). At the end of the interval, the element releases its stored impulses by imparting percussions on its nodes, causing the linear momentum of the nodes to be altered. The resulting nodal trajectories can be regarded as piecewise linear in time. We note that adjacent elements interact by transferring linear momentum through their common nodes. Note that the resulting algorithm is explicit of the central-difference type.

4.3. Implementation of AVIs

In this section, we turn our attention to discussing the implementation of the AVI corresponding to the discrete Lagrangian (46).

Owing to the algorithm's asynchronous nature, a suitable scheduling procedure which determines the order of operations while ensuring causality must be carefully designed. One particularly efficient implementation consists of maintaining a priority queue (see, e.g. Reference [37]) containing the elements of the triangulation.¶¶ The elements in the priority queue are ordered according to the next time at which they are to become active. Thus, the top element in the queue, and consequently the next element to be processed, is the element whose next activation time is closest to the present time.

¶¶We stress that this is only one possibility for ordering the computation, and is not imposed by the discrete Euler–Lagrange equations. We will discuss some other possibilities in Section 4.7.

Explicit AVI Algorithm

Input data: $T, \mathcal{T}, \left\{ x_a^0, \dot{x}_a^{1/2} \mid a \in \mathcal{T} \right\}$

▷ Initialization

   ▷ $q_a \leftarrow x_a^0$, $v_a \leftarrow \dot{x}_a^{1/2}$, $\tau_a \leftarrow 0$ for all $a \in \mathcal{T}$
   ▷ Do for all $K \in \mathcal{T}$

      ▷ $\tau_K \leftarrow 0$
      ▷ Compute t_K^1
      ▷ Push (t_K^1, K) into priority queue

   ▷ End Do

▷ Iterate over the elements in time

   ▷ Do until priority queue is empty

      ▷ Extract next element: Pop (t, K) from priority queue
      ▷ Update positions: $q_a \leftarrow q_a + v_a (t - \tau_a)$, for all $a \in K$
      ▷ Update node's time: $\tau_a \leftarrow t$, for all $a \in K$
      ▷ If $t < T$
         ▷ Update velocities:
            $v_a \leftarrow v_a - (t - \tau_K) \frac{\partial V_K}{\partial x_a} (q_K) / m_a$, for all $a \in K$
         ▷ Update element's time: $\tau_K \leftarrow t$
         ▷ Compute t_K^{next}
         ▷ Schedule K for next update:
            Push (t_K^{next}, K) into priority queue

   ▷ End Do

▷ End

Figure 7. Algorithm implementing the discrete Euler–Lagrange equations of the action sum given by Equation (46).

The general flow of the calculations is as follows. The priority queue is popped in order to determine the next element to be processed. The new configuration of this active element is computed from the current velocities of the nodes. Subsequently, these velocities are modified by impulses computed based on the new element configuration. Finally, the next activation time for the element is computed and the element is pushed into the queue. A flow chart of the numerical procedure is given in Figure 7. Note that the algorithm allows the time step of each element to change in time.

The use of the priority queues is particularly simple in C++, since these are provided by several freely available libraries. Also, routines implementing priority queues in C are freely available to download. Priority queues are frequently implemented through balanced binary trees [37, pp. 458].

The adaption of existing FE codes to implement the explicit AVI integrator is fairly simple. The computations at the element level remain untouched, while the driver that assembles the global internal force vector should be removed, since there is no assembly required. Instead, a driver that implements the operations in Figure 7 should be coded. Note that apart from the priority queue and two arrays to store elemental and nodal times, respectively, no extra storage nor new arrays are required over traditional explicit Newmark schemes. To plot the configuration of the continuum, a short routine computing the positions of the nodes at the time of the most advanced element is needed. In this case, each node is advanced by following a linear trajectory parallel to its velocity. It is worth noting that explicit AVIs allow for the reusage of most of the existing FE structural dynamics codes.

4.4. Momentum conservation properties

In order to derive the discrete momentum balance equations, we consider a one-parameter group of trajectories with nodal positions given by $(x_a^i)^\varepsilon = \exp(\varepsilon\Omega)x_a^i + \varepsilon v$, for any vector $v \in \mathbb{R}^3$ and skew-symmetric matrix Ω. We assume that the discrete action sum is translation and rotation invariant, so that the value of S_d for $(x_a^i)^\varepsilon$ is equal to its value for x_a^i.

Differentiating the discrete action sum of the perturbed trajectory with respect to ε gives

$$0 = \left.\frac{\partial}{\partial \varepsilon} S_d((x_a^i)^\varepsilon)\right|_{\varepsilon=0} = \sum_a \sum_{i=0}^{N_a} D_a^i S_d v + \sum_a \sum_{i=0}^{N_a} (x_a^i \times D_a^i S_d)\omega \tag{51}$$

where $\omega \in \mathbb{R}^3$ is the axial vector of Ω, which in terms of the Hodge star operator is $\omega = *\Omega$.

We now use the fact that x_a^i is a trajectory of the system and so it satisfies the discrete Euler–Lagrange equations (48). Then $D_a^i S_d = 0$ for all a and $i = 1, \ldots, (N_a - 1)$. Under these conditions, Equation (51) reduces to

$$0 = \left[\sum_a D_a^0 S_d + \sum_a D_a^{N_a} S_d\right] v + \left[\sum_a x_a^0 \times D_a^0 S_d + \sum_a x_a^{N_a} \times D_a^{N_a} S_d\right] \omega \tag{52}$$

As this must hold for all v and all ω we obtain

$$\sum_a D_a^{N_a} S_d = -\sum_a D_a^0 S_d \tag{53}$$

$$\sum_a x_a^{N_a} D_a^{N_a} S_d = -\sum_a x_a^0 D_a^0 S_d, \tag{54}$$

which furnishes precise statements of discrete linear and angular momentum conservation, respectively. Using the particular expression (45) for the discrete action sum we can evaluate (53) and (54) to give

$$\sum_a p_a^{1/2} = \sum_a p_a^{N_a - 1/2} \tag{55}$$

$$\sum_a x_a^0 \times p_a^{1/2} = \sum_a x_a^{N_a} \times p_a^{N_a - 1/2} \tag{56}$$

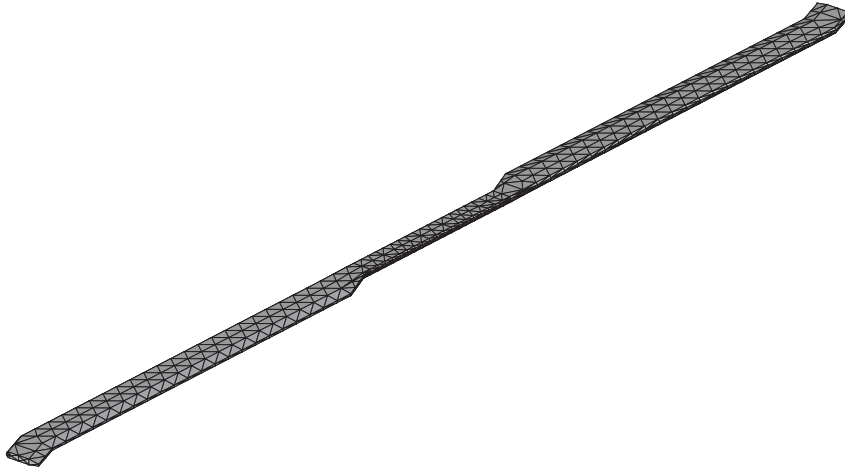which clearly shows that both momenta are conserved.

Figure 8. Mesh of the blade. It consists of 2089 ten-noded tetrahedral elements and 4420 nodes.

Momentum evolution without symmetry: We assumed above that the discrete action sum was translation and rotation invariant, which will only be the case if there are no displacement boundary conditions and no forces or torques are applied to the body. If there are such effects, then we can carry out the same calculation but with the left-hand side of (51) being non-zero. In fact, this non-zero term will exactly give the net result force and moment and so (55) and (56) will give discrete versions of the familiar statements that change in linear and angular momentum should equal the total impulse and moment of impulse imparted to the system. These expressions are worked out in general in Reference [19].

4.5. Numerical examples

In this section, we illustrate the performance of an explicit AVI for structural dynamics by simulating the motion in vacuum of a simplified model of the blades of an Apache AH-64 helicopter. The simulation of the dynamics of rotor blades and similar systems has long been a challenge and a test bed for time-integration algorithms for structural dynamics [38–43]. The slender geometry of the blades renders the system prone to dynamic instability, and the need to compute accurate solutions for long periods of time, to conserve linear and angular momentum, and to accurately distribute the energy across the frequency spectrum place stringent tests on time-integration methods.

The blade has a span of 7.2 m, a chord of 533 mm and a maximum thickness of 40 mm. The three-dimensional mesh of the blade is shown in Figure 8. It consists of 2089 ten-noded tetrahedral elements and 4420 nodes. The original geometry of the hub of the blade rotor was replaced by a simple straight joint. The blades comprise five steel spars reinforced with glass fibre. However, for this example we simply replace the hollow structure by an effective homogeneous material. The strain-energy density is given by

$$W(F, X) = \frac{\lambda_0(X)}{2}(\log J)^2 - \mu_0(X) \log J + \frac{\mu_0(X)}{2} \mathrm{tr}(F^{\mathsf{T}} F) \tag{57}$$

which describes a Neohookean solid extended to the compressible range. In this expression, $\lambda_0(X)$ and $\mu_0(X)$ are—possibly inhomogeneous—Lamé constants. The corresponding Piola stress-deformation relation follows from (41) in the form

$$P = \lambda_0 \log J F^{-\mathsf{T}} + \mu_0(F - F^{-\mathsf{T}}) \tag{58}$$

As the initial condition at time $t = 0$, we consider the blade in its undeformed configuration with a velocity field corresponding to an angular velocity $\omega = 40$ rad/s. Owing to the absence of external forces or dissipation the energy is conserved and the centre of mass does not move from its original position for all times. We classify cases according to the dimensionless angular velocity

$$\hat{\omega} = \frac{\omega L^2}{cw}$$

where ω is the nominal angular velocity of the blade, L its span, w is the chord of the blade, and

$$c = \sqrt{\frac{\lambda_0 + 2\mu_0}{\rho_0}}$$

is a nominal dilatational wave speed of the material. In all examples, the elemental time steps are determined from the Courant condition, which provides an estimate of the stability limit for explicit integration (c.f. e.g. Reference [2]). Specifically, the value of the time step for each element is set to a fraction of the Courant limit and is computed as

$$\Delta t = f \frac{h}{c} \tag{59}$$

where $f = \frac{1}{10}$ and h is the radius of the largest ball contained in the element. The time step is kept constant in each element throughout the computation. We note that the elemental time steps are not required to be integer-related, and the element trajectories are not synchronized in general.

We proceed to report three test cases which illustrate the performance and properties of the AVIs. The first case consists of a homogeneous blade with material constants $\rho_0 = 250$ kg/m^3, $\lambda_0 = 0.98$ GPa and $\mu_0 = 0.98$ GPa, and, correspondingly, $\hat{\omega} = 1.13$. For this choice of parameters the computed trajectory is close to a rigid rotation, Figure 9, and remains stable both for short (Figure 9(a)) and long (Figure 9(b)) times.

The second case consists of a homogeneous blade with material constants $\rho_0 = 2500$ kg/m^3, $\lambda_0 = 4$ GPa and $\mu_0 = 4$ GPa, and, correspondingly, $\hat{\omega} = 1.78$. Moderately large amplitude oscillations, including spanwise torsional modes, develop during the initial transient, Figure 10(a). Remarkably, after this transient the blade settles down to a fairly stable shape and rotates almost rigidly.

Finally, we consider a two-material blade. The central joint consists of a stiff material with constants $\rho_0 = 4500$ kg/m^3, $\lambda_0 = 3$ GPa and $\mu_0 = 0.75$ GPa, while the remainder of the blade is composed of a softer material with constants $\rho_0 = 250$ kg/m^3, $\lambda_0 = 0.1$ GPa and $\mu_0 = 0.025$ GPa. The contrast between the dilatational wave speeds of the two materials, and therefore between the stable time steps for elements of the same size in the two different regions, is 1.3. The value of the dimensionless angular velocity relative to the softer material is $\hat{\omega} = 5.02$. The
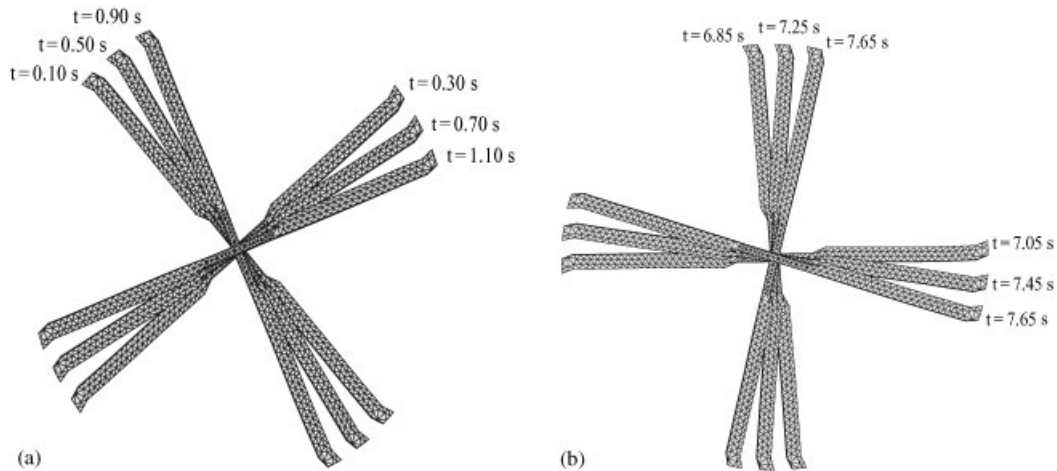
Figure 9. Evolution of the blade for the first and most rigid case. The motion of the blade is essentially that of a rigid body. The centre of mass does not move, a consequence of the discrete linear momentum conservation, and the period of the blade is very close to the one of a completely rigid blade, since the spanwise elongation is negligible. The final snapshots correspond to approximately 266 million updates of the smallest element in the mesh: (a) initial steps; and (b) final steps.
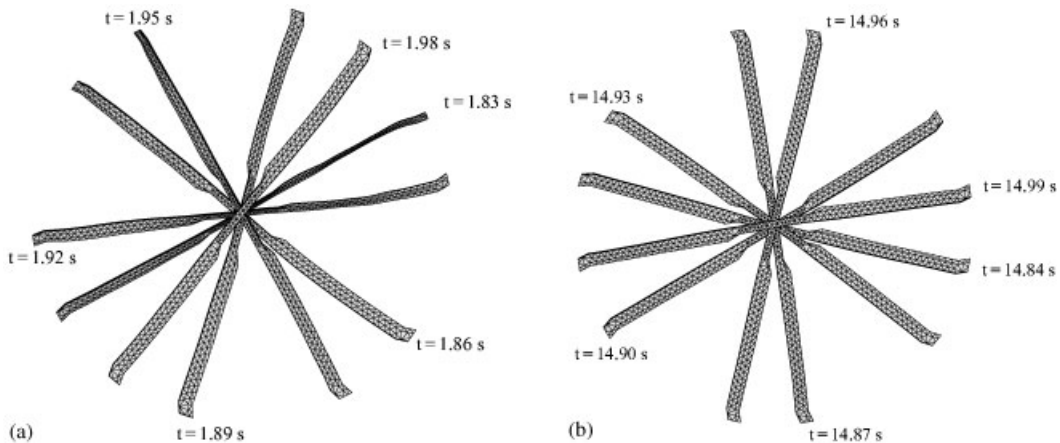

Figure 10. Evolution of the blade for the second case. During the initial phases of the motion, some fairly large deflections, including torsion along the spanwise direction, occur. However, after a relatively long time the blade rotates with an almost fixed shape. The period of rotation has changed slightly with respect to a rigid blade, since there is a non-negligible spanwise elongation inducing a change in the corresponding moment of inertia. The final snapshots correspond to approximately 325 million updates of the smallest element in the mesh: (a) initial steps; and (b) final steps.
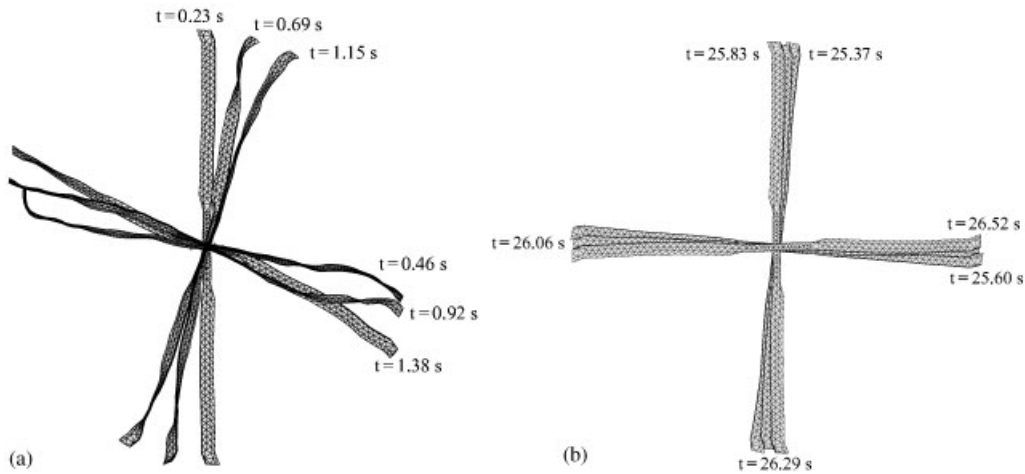
Figure 11. Evolution of the blade for the third and softest case. During the initial phases of the motion, the blade behaves as a very flexible strip. Surprisingly, after a relatively long time the blade settles down rotates with a small amplitude oscillation close to an almost fixed shape. The period of rotation with respect to a rigid blade has changed considerably, since the spanwise elongation is large. The final snapshots correspond to approximately 234 million updates of the smallest element in the mesh: (a) initial steps; and (b) final steps.

Table I. Period of rotation of the blade for long times. As the value of $\hat{\omega}$ grows, the blade deforms more increasing its span, and therefore its moment of inertia, to accommodate the centrifugal forces. Since the discrete angular momentum is conserved, the period rotation should grow accordingly.

Case	$\hat{\omega}$	Period (s)
1	1.13	0.1575 ± 0.0005
2	1.78	0.1595 ± 0.0005
3	5.02	0.1845 ± 0.0005

initial transient is characterized by very large-amplitude oscillations in the blade, Figure 11(a), while the central joint remains essentially rigid. As in the previous case, after this transient the blade settles down to a stable nearly rigid orbit.

A notable feature of the solutions is that, despite their asynchronous character, they advance smoothly in time without ostensible jerkiness or vacillation. Also, note that the centre of mass of the blade does not move, which is a consequence of the conservation of linear momentum by the algorithm. The period of rotation for long times is of particular interest. A perfectly rigid blade would rotate once every 0.157 s. However, a flexible blade increases its span under the action of the inertia forces induced by the rotary motion, and its angular velocity slows down in order to conserve the total angular momentum. This effect is indeed observed in the simulations, as shown in Table I, and is a consequence of angular-momentum conservation.
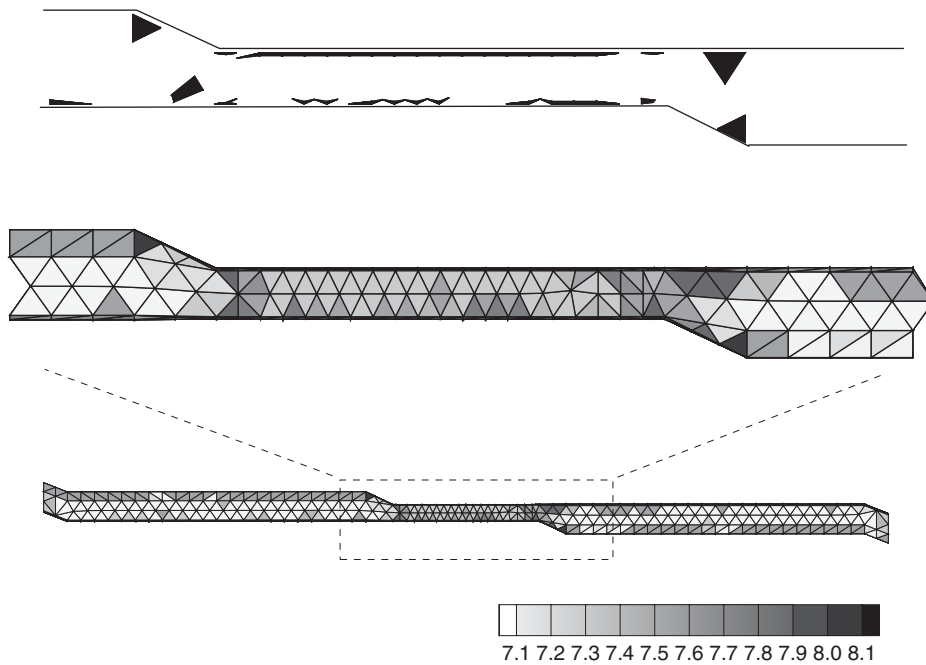
Figure 12. Contour plot of the $\log_{10}$ of the number of times each element was updated by the AVI after 27.439 s of simulation of case 3, in which inertial forces prevail. The picture in the middle shows an enlargement of the central part of the blade, which is made out of a stiffer material than the rest. The abrupt change in the number of elemental updates between the two region is noteworthy. Additionally, the picture on the top shows only those few elements updated more than 10^8 times. These elements or slivers would drive the whole computation speed down for a constant time step algorithm, while AVIs circumvents this difficulty gracefully.

A measure of performance of the AVIs is shown in Figure 12, which depicts the number of updates in each of the elements of the mesh for the third numerical example described above. As is evident from the figure, the larger elements in the mesh are updated much less frequently than the finer elements. In particular, a small number of slivers are updated very frequently, as required by their small Courant limit. Also, the elements in the central joint, made of a stiffer material, are updated more often than the more flexible elements in the blade.

Some relevant statistics are collected in Table II. Overall, in the present example the number of AVI updates is roughly 15% of the number of updates required by explicit Newmark at constant time step. It should be carefully noted, however, that in the example under consideration the vast majority of the elements have similar sizes and aspect ratios, differing only by at most one order of magnitude. It is easy to set up examples in which the update count of the constant time-step algorithm bears an arbitrarily large ratio to the update count of the AVI. A case which arises in practice with some frequency concerns a roughly uniform triangulation of the domain which contains a small number of high aspect-ratio elements. The presence of a single bad element suffices to drive down the critical time step for explicit integration to an arbitrarily small value. This problem often besets explicit dynamics, especially in three

Table II. Maximum and minimum number of elemental updates for a single element at the final time. The total column shows the sum of the number of elemental updates in the whole mesh at the final time. In contrast, traditional time-stepping algorithms would have advanced with the same number of updates on each element, which is equal to the value in the Maximum column. The ratio between the total number of updates in the whole mesh in these two cases is shown in the speed-up column, a direct measure of the cost saving features of AVI.

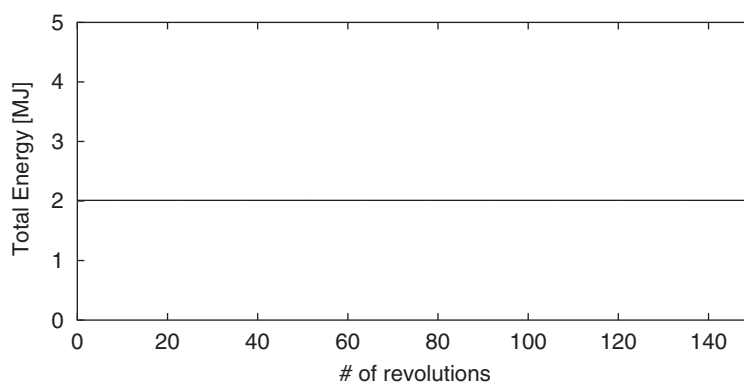| Case	$\hat{\omega}$	Final time (s)	Maximum	Minimum	Total	Speed-up
1	1.13	7.849	266×10^6	14×10^6	8.7×10^{10}	6.37
2	1.78	15.015	325×10^6	17×10^6	42.6×10^{10}	6.37
3	5.02	27.439	234×10^6	12×10^6	8.4×10^{10}	5.80

Figure 13. Evolution of the total energy in the blade as a function of the number of revolutions of the blade, for the third and softest case. Remarkably, the energy remains nearly constant even after the smallest element in the mesh has been updated more than 200 million times, at the end of the horizontal axis.

dimensions where bad elements, or *slivers*, are difficult to eliminate entirely. The AVI algorithm effectively sidesteps this difficulty, as bad elements drive down their own times steps only, and not the time steps of the remaining elements in the mesh. In this manner, the overall calculation is shielded from the tyranny of the errant few.

In Section 3.5, we alluded to the excellent energy-conservation properties of variational integrators. Our numerical tests (see also Reference [19]) suggest that the AVIs possess excellent energy-conservation properties as well. Thus, for instance, Figure 13 shows the time evolution of the total energy of the blade in the third case. It is remarkable that the energy remains nearly constant throughout the calculations, which entail 234×10^6 updates of the smallest element in the mesh, or approximately 150 revolutions of the blade, despite the dynamically unstable initial transient. The energy behaviour is equally good in the remaining numerical examples.

4.6. Complexity and convergence

The computational costs involved in the AVI algorithm can be separated into those associated with the element updates, and the overhead involved in the determination of which element to

update next. It is readily verified, however, that the latter cost is generally much smaller than the former, as shown next.

Complexity and cost estimates: We proceed to estimate the speed-up afforded by AVIs relative to explicit Newmark, or other similar explicit method, with a time step

$$\Delta t_{\min} = \min_K \Delta t_K$$

For a fixed final time T the cost C_{AVI} of the AVI method is

$$C_{\mathrm{AVI}} = \underbrace{(C_{\mathrm{EU}} + C_{\mathrm{PQ}})}_{\text{cost per timestep}} \underbrace{\sum_K \frac{T}{\Delta t_K}}_{\text{\# timesteps}} \tag{60}$$

where C_{EU} is the cost per element update and C_{PQ} is the cost of inserting and removing one element from the priority queue. The corresponding cost for explicit Newmark is

$$C_{\mathrm{NM}} = C_{\mathrm{EU}} \sum_K \frac{T}{\Delta t_{\min}} \tag{61}$$

If we neglect the priority queue cost by setting $C_{\mathrm{PQ}} = 0$ then we can calculate the maximum AVI speed-up to be

$$\lambda = \left. \frac{C_{\mathrm{NM}}}{C_{\mathrm{AVI}}} \right|_{C_{\mathrm{PQ}} = 0} = \frac{\sum_K (1/\Delta t_{\min})}{\sum_K (1/\Delta t_K)}. \tag{62}$$

We now consider the effect on the cost of the priority queue, which we assume to be implemented using a balanced binary tree. Under these conditions, the cost is $C_{\mathrm{PQ}} = \kappa \log_2(E)$, where E is the number of elements in the mesh and κ is the cost per level of the binary tree, which can be taken to be constant. Specifically, our benchmarks give $\kappa \approx 0.2 \, \mu s$. By contrast, the cost C_{EU} of an element update depends sensitively on the type of element and the constitutive model, and can be substantial for complex constitutive models. Even for relatively simple non-linear elasticity models, however, this cost is still greatly in excess of C_{PQ}. For example, for a Neohookean material and six-noded triangles, our numerical benchmarks give $C_{\mathrm{EU}} \approx 10 \, \mu s$. We can now estimate the size of the mesh for which the cost C_{PQ} begins to dominate C_{EU}. The ratio of costs is given by

$$\frac{C_{\mathrm{PQ}}}{C_{\mathrm{EU}}} = \frac{\kappa}{C_{\mathrm{EU}}} \log_2(E) = 0.02 \log_2(E)$$

and for C_{PQ} to be greater than C_{EU} would thus require $\log_2(E) > 50$, i.e. an inordinately large mesh containing on the order of 10^{15} elements. Taking the priority-queue overhead into consideration, true AVI speed-up is

$$\frac{C_{\mathrm{NM}}}{C_{\mathrm{AVI}}} = \frac{\lambda}{1 + (\kappa/C_{\mathrm{EU}}) \log_2(E)} \tag{63}$$

In the helicopter-blade calculations, the number of elements is $E = 2089$, the cost of an elemental update for a ten-node tetrahedral element is $C_{\mathrm{EU}} \approx 25 \, \mu s$ and thus

$$\frac{C_{\mathrm{NM}}}{C_{\mathrm{AVI}}} = \frac{6.4}{1 + (0.2/25) \log_2(2089)} = 5.9$$
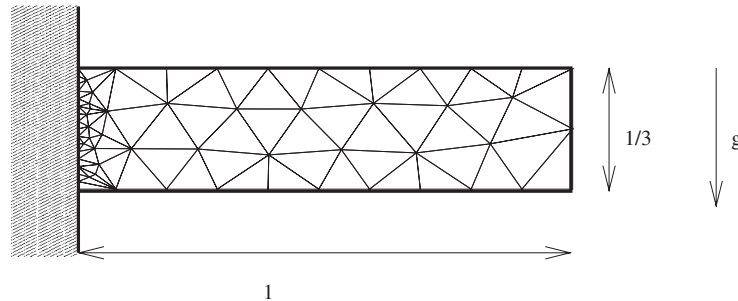
Figure 14. Schematic diagram of the geometry of the slab and the coarsest
mesh for the cost/accuracy example.

which is close to the optimal speed-up $\lambda = 6.4$. These estimates demonstrate the relatively
modest impact of the priority queue on the overall performance of the method.

Cost/accuracy analysis: We quantitatively demonstrate that asynchronous methods can sub-
stantially lower the cost of a computation without degradation in accuracy by means of the fol-
lowing numerical test. Consider a two-dimensional plane strain, non-linear elastic slab clamped
on one edge and free at the other, deflecting under the action of uniform gravity. The ma-
terial is Neohookean extended to the compressible range, Equation (58). The slab is initially
undeformed and at rest. We consider an initial non-uniform mesh composed of 88 six-node
quadratic triangles. A schematic picture of the geometry and the initial mesh are shown in
Figure 14.

The initial mesh is refined by uniform quadrisection several times over, resulting in meshes
containing 88, 352, 1408, 5632 and 22 528 elements. For each mesh, the solution is computed
up to a prescribed final time using both AVI and explicit Newmark. The explicit Newmark
calculations are started as discussed in Section 2.4. In the AVI calculations, the appropriate
initial conditions at node a are

$$x_a^0 = \varphi_0(X_a)$$

$$v_a^{1/2} = \frac{1}{2m_a} \sum_{\{K \mid a \in K\}} I_{Ka}^0$$

where $\varphi_0(X)$ is the initial deformation mapping and

$$I_{Ka}^0 = -\Delta t_K \frac{\partial V_K(x_K^0)}{\partial x_{Ka}^0}$$

Note that this initialization procedure reduces to Newmark's for a uniform time step. An explicit
Newmark solution computed with an exceedingly fine 90112-element mesh is presumed to be
ostensibly converged, and used in lieu of the exact solution in order to compute errors.

The L^2-norm displacement and deformation-gradient errors are shown in Figure 15 as a
function of the computational cost. It is evident from these plots that the computational cost
incurred in the computation of solutions of equal accuracy is substantially less for the AVIs.
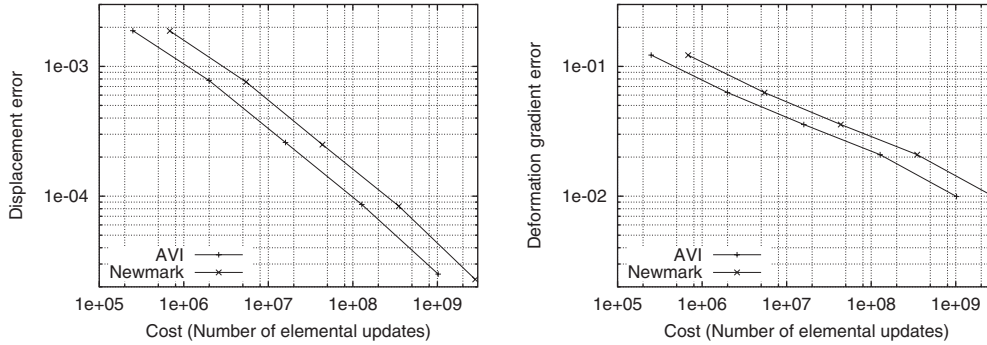
Figure 15. L^2 errors for the displacement (on the left) and the deformation gradient (on the right) as a function of the number of elemental updates for the slab problem. As is readily seen from the plots, AVIs are substantially cheaper than Newmark in computational cost for a desired error value.

Convergence of AVIs: The slope of the plots in Figure 15 is also directly related to the rate of convergence in Δt. To verify this, we note that the elemental time steps are proportional to the inner radius of the element, by virtue of the Courant condition. This implies that the rate of convergence in Δt is three times the slope of the plots (Figure 15). This gives a displacement rate of convergence of roughly 1.9, and a deformation-gradient rate of convergence of 1.1, for both methods.

The convergence of AVIs may also be established analytically. To this end, define the *maximum timestep*

$$\Delta t_{\max} = \max_K \Delta t_K \tag{64}$$

and the *maximum final time*

$$t_{\max} = \max_K N_K \Delta t_K \tag{65}$$

With these definitions we can state the following result.

Theorem 4.1
Consider a sequence of solutions obtained by the application of asynchronous variational integrators to a fixed spatial discretization with maximum time step $\Delta t_{\max} \to 0$ and maximum final time $t_{\max} \to T$. Then the final discrete configuration converges to the exact configuration of the semidiscrete problem at time T.

Proof
See Appendix A and Corollary A.8. □

Resonance instabilities in asynchronous methods: It is known that multi-time-step methods occasionally exhibit resonance instabilities for some combinations of time steps (See, e.g. References [44–46] in the case of molecular dynamics, and References [47, 48] in the case of finite element methods in elastodynamics). We have not been able to detect any such instabilities in our simulations; however, a systematic study has yet to be performed.

4.7. Extensions of AVIs

In this section, we comment on some possible extensions and ideas to be explored. The uninterested reader can skip it and continue directly to the next section.

Implicit AVI: Asynchronous algorithms do not necessarily need to be explicit. A locally implicit AVI method can be constructed by using, for instance, the discrete action sum

$$S_d = \sum_a \sum_{i=0}^{N_a-1} \frac{1}{2} m_a (t_a^{i+1} - t_a^i) \left\| \frac{x_a^{i+1} - x_a^i}{t_a^{i+1} - t_a^i} \right\|^2 - \sum_K \sum_{j=0}^{N_K-1} (t_K^{j+1} - t_K^j)\, V_K(x_K^{(j+1)*})$$

where

$$(x_K^{(j+1)*})_a = x_a^{i+1} \alpha_a + x_a^i (1 - \alpha_a)$$

and such that $t_K^{j+1} = t_a^{i+1}$, with a nodal parameter $\alpha_a \in [0, 1]$ for each node a. The corresponding discrete Euler–Lagrange equations possess q_a^{i+1} for every node $a \in K$ involved implicitly as an argument of the potential energy of the element, which needs to be solved for.

The algorithm shown in Figure 7 is modified to replace the explicit update of the positions of the nodes. It is substituted for the solution of a non-linear system of equations for the new positions of all the nodes in the element under consideration.

Probabilistic sorting for element updates: In Section 4.3, we considered solving the AVI equations (48) by maintaining a priority queue to track which element has the smallest next time in the entire mesh. This is an overly strong requirement for solving the equations, however, as we can update an element whenever all of the adjacent elements have larger next times. This methodology also provides a basis for the construction of parallel implementations of AVI methods.

One simple scheme for doing this is to maintain the array of next times for each element. To perform a time step, we select an element at random and check whether its next time is less than that of all adjacent elements. If it is, then we update it and start another time step. If not, we move to the adjacent element with the smallest next time, and proceed to check its adjacent elements, repeating the test until we find an element with next time less than its neighbours. This procedure is clearly guaranteed to terminate, and it will always find an element which we can update.

The advantage of this method over the priority queue used earlier is that the cost will be constant in the number of elements in the mesh. To implement the method efficiently requires storing adjacency information in the mesh data structures, with which the cost of this algorithm will be very small compared with the other operations necessary to perform an elemental update.

To estimate the cost of this algorithm we note that in the worst case we may need to iterate over every element in the mesh before finding an element which we can update. This is extremely unlikely, however, and so it is appropriate to ask how many steps are necessary on average. We can readily calculate an upper bound for this by assuming an infinite mesh and that each element test is independent, which is overly pessimistic. For a mesh in which each element is adjacent to r other elements, the average number of steps before this algorithm

finds an appropriate element is bounded by

$$N_{\text{step}} = \sum_{n=1}^{\infty} n \left(1 - \left(\frac{1}{2} \right)^r \right) \left(1 - \left(\frac{1}{2} \right)^{r-1} \right)^{n-1} \left(\frac{1}{2} \right)^{r-1}$$

$$= 2^{r-1} - \frac{1}{2} \tag{66}$$

For tetrahedral elements, such as those used in the numerical simulations in Section 4.5, each element has $r = 4$ adjacent elements and so the average number of steps necessary to find an element to update would be 7.5, irrespective of the size of the mesh. This algorithm would thus add a negligible cost to the AVI implementation and allow it to achieve almost the maximum speed-up factor λ over explicit Newmark.

5. SPACETIME LAGRANGIAN MECHANICS

In this section, we review the abstract formulation of spacetime Lagrangian mechanics and apply it to dynamic and static continuum mechanics and ordinary differential equations. The salient feature of the forthcoming development is that it considers variations in both the reference spacetime and the deformed configurations. As a consequence, we obtain the energy conservation and the equilibrium of configurational forces as Euler–Lagrange equations of the variational principle.

5.1. Variational principles and equations of motion

Now we derive the equations of motion for a continuum. For this, let $\mathscr{X} \in \mathbb{R}^m$ be an open spacetime reference configuration and Z^μ denote the μth co-ordinate in $\mathscr{X}$. We seek a deformation mapping $\varphi(Z)$ that takes the point Z in the reference configuration to the point $x \in \mathbb{R}^n$, in the deformed configuration, with co-ordinates x^i, $i = 1, \ldots, n$. The Lagrangian density of the continuum is given by a function $L(\varphi, \nabla_Z \varphi, Z)$, and the action functional follows as

$$S[\varphi] = \int_{\mathscr{X}} L(\varphi, \nabla_Z \varphi, Z) \, \mathrm{d}Z$$

The variational principle states that the deformation mapping should be a critical point of the action functional. The variations of the action functional are

$$\delta S[\varphi] = \int_{\partial \mathscr{X}} \frac{\partial L}{\partial \varphi^i_{,\mu}} N_\mu \delta \varphi^i \, \mathrm{d}Z - \int_{\mathscr{X}} \left[\frac{\partial L}{\partial \varphi^i} - \frac{\partial}{\partial Z^\mu} \frac{\partial L}{\partial \varphi^i_{,\mu}} \right] \delta \varphi^i \, \mathrm{d}Z \tag{67}$$

where N is the outer unit normal to $\mathscr{X}$. The corresponding Euler–Lagrange equations are obtained to be

$$\frac{\partial L}{\partial \varphi^i} - \frac{\mathrm{d}}{\mathrm{d}Z^\mu} \frac{\partial L}{\partial \varphi^i_{,\mu}} = 0 \tag{68}$$

for all $Z \in \mathscr{X}$.

Boundary conditions: The deformation mapping is subject to essential boundary conditions on part of the boundary $\partial_d \mathcal{X} \subset \partial \mathcal{X}$. Traction-like boundary conditions are imposed by requesting

$$\frac{\partial L}{\partial \varphi^i_{,\mu}} N_\mu = \tau_i \tag{69}$$

on $\partial_\tau \mathcal{X} \subset \partial \mathcal{X}$, where τ is a vector-valued field on $\partial_\tau \mathcal{X}$. We enforce the essential boundary conditions directly. On the other hand, we modify the variational principle by asking it to satisfy

$$\delta S[\varphi] = \int_{\partial_\tau \mathcal{X}} \tau_i \delta \varphi^i$$

for all admissible variations in order to impose the traction-like boundary conditions. Of course, conditions on the sets $\partial_\tau \mathcal{X}$ and $\partial_d \mathcal{X}$ are necessary to guarantee the well-posedness of the problem.

Dynamic continuum mechanics: The equations describing the dynamics of a continuum are obtained by choosing $\mathcal{X} = (0, T) \times \mathcal{B}$, where $\mathcal{B}$ is an open connected region of $\mathbb{R}^3$. In keeping with standard practice, we let $Z^0 = t$ be the time co-ordinate, and $Z^I = X^I$, $I = 1, 2, 3$, be the spatial co-ordinates, so the co-ordinates of a point Z are given by (t, X^1, X^2, X^3).

The Lagrangian density is chosen to be of the form

$$L(\varphi, \dot{\varphi}, \nabla_X \varphi, X) = \frac{\rho_0}{2} \|\dot{\varphi}\|^2 - W(\nabla_X \varphi, X) \tag{70}$$

which, as mentioned previously, describes the dynamics of hyperelastic materials. The Euler–Lagrange equations (68) translate into the infinitesimal balance of linear momentum (42). If present, the traction-like boundary conditions (69) reduce to prescribing the tractions over $(0, T) \times \partial \mathcal{B}$ or the velocities in $\mathcal{B}$ at time $t = 0$ or T.

Static continuum mechanics: The static equilibrium equations for a hyperelastic continuum are obtained by choosing $\mathcal{X} = \mathcal{B} \subset \mathbb{R}^3$ and a Lagrangian density of the form

$$L(\varphi, \nabla_X \varphi, X) = W(\nabla_X \varphi, X) \tag{71}$$

The Euler–Lagrange equations (68) reduce to

$$\nabla_X \cdot P = 0$$

Ordinary differential equations: In this case, spacetime has only one dimension, namely time, and the Euler–Lagrange (68) equations reduce to (2).

5.2. Noether's theorem and momentum conservation

Consider a one-parameter family φ^ε of deformation mappings that are a symmetry of the mechanical system, i.e. $S[\varphi] = S[\varphi^\varepsilon]$ where $\varphi^0 = \varphi$. Denote by

$$\xi = \left.\frac{\partial \varphi^\varepsilon}{\partial \varepsilon}\right|_{\varepsilon=0}$$

the infinitesimal generator of the symmetry. Therefore, we have that

$$0 = \left.\frac{\partial S[\varphi^\varepsilon]}{\partial \varepsilon}\right|_{\varepsilon=0} = \int_{\partial \mathcal{X}} \frac{\partial L}{\partial \varphi^i_{,\mu}} N_\mu \xi^i \, \mathrm{d}Z \tag{72}$$
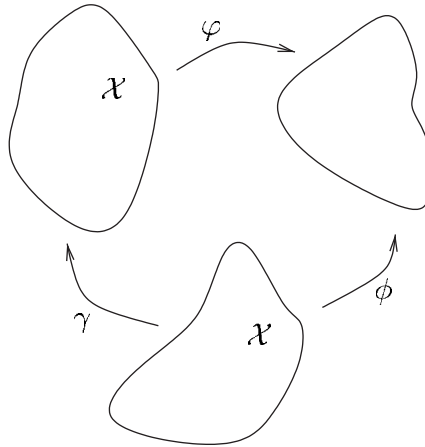
Figure 16. Schematic representation of the relation between γ, ϕ and φ. The parametric configuration $\bar{\mathscr{X}}$ and the reference configuration $\mathscr{X}$ are isomorphic.

where we have used the definition of the symmetry for the left equality and Equation (67) for the right one, assuming φ satisfies the Euler–Lagrange equations (68). This is the integral form of Noether's theorem.

In fact, this holds over any subset $\mathscr{X}' \subseteq \mathscr{X}$, as can be seen by repeating the previous calculation with the action integral restricted to $\mathscr{X}'$.

Dynamic continuum mechanics: Given $\mathscr{X} = (0, T) \times \mathscr{B}$ as before, the boundary is given by $\partial \mathscr{X} = \partial \mathscr{B} \cup \{0\} \times \mathscr{B} \cup \{T\} \times \mathscr{B}$. Expression (72) becomes

$$\int_{\mathscr{B}} \frac{\partial L}{\partial \dot{\varphi}^i} \xi^i \, \mathrm{d}X \bigg|_{t=T} - \int_{\mathscr{B}} \frac{\partial L}{\partial \dot{\varphi}^i} \xi^i \, \mathrm{d}X \bigg|_{t=0} = - \int_0^T \int_{\partial \mathscr{B}} \frac{\partial L}{\partial \varphi^i_{,I}} \xi^i N_I \, \mathrm{d}X \, \mathrm{d}t \tag{73}$$

Note that the global balance of linear and angular momentum for the Lagrangian density (70) is obtained from Equation (73) by considering $\varphi^\varepsilon = \varphi + \varepsilon v$ and $\varphi^\varepsilon = \exp(\varepsilon \Omega) \varphi$, respectively. Here v is any vector in $\mathbb{R}^3$ and Ω is any 3×3 skew-symmetric matrix.

5.3. Restated variational principles and configurational forces

The variational principle in Section 5.1 can be restated in a more general way to obtain the equilibrium equations for configurational forces, of which energy balance and the conservation of the J and L integrals are examples. Consider a smooth one-to-one and onto map γ from an open set $\bar{\mathscr{X}} \subset \mathbb{R}^m$, the parametric configuration, to the reference configuration $\mathscr{X}$. Denote by $\phi = \varphi \circ \gamma$ the composition map from the parametric configuration $\bar{\mathscr{X}}$ to the deformed configuration, as illustrated in Figure 16. The μth co-ordinate of a point in the parametric configuration is denoted $\bar{Z}^\mu$.

Next, consider the action functional of Section 5.1 as a function of ϕ and γ

$$S[\phi, \gamma] = \int_{\mathscr{X}} L(\phi \circ \gamma^{-1}, \nabla_Z(\phi \circ \gamma^{-1}), Z) \, \mathrm{d}Z = \int_{\bar{\mathscr{X}}} L(\phi, \nabla_{\bar{Z}}\phi \cdot \nabla_{\bar{Z}}\gamma^{-1}, \gamma) \det \nabla_{\bar{Z}}\gamma \, \mathrm{d}\bar{Z}$$

Varying the action with respect to ϕ gives the Euler–Lagrange equations (68). The variations of the action with respect to γ are

$$\delta S|_\phi = \int_{\mathcal{X}} \left[\frac{\partial L}{\partial Z^\nu} + \frac{\mathrm{d}}{\mathrm{d}Z^\mu} \left(\frac{\partial L}{\partial \varphi^i_{,\mu}} \varphi^i_{,\nu} - L\delta^\mu_\nu \right) \right] \delta\gamma^\nu \, \mathrm{d}Z - \int_{\partial\mathcal{X}} \left(\frac{\partial L}{\partial \varphi^i_{,\mu}} \varphi^i_{,\nu} - L\delta^\mu_\nu \right) N_\mu \delta\gamma^\nu \, \mathrm{d}Z \quad (74)$$

By restricting the variations of γ to be zero on the boundary, we obtain the corresponding Euler–Lagrange equation

$$\frac{\partial L}{\partial Z^\nu} + \frac{\mathrm{d}}{\mathrm{d}Z^\mu} \left(\frac{\partial L}{\partial \varphi^i_{,\mu}} \varphi^i_{,\nu} - L\delta^\mu_\nu \right) = 0 \quad (75)$$

Note that by not varying γ on the boundary of $\bar{\mathcal{X}}$, the composition $\phi \circ \gamma^{-1}$ is always a deformation mapping of the class considered in Section 5.1. Equations (75) then should not add any new condition on the dynamics, since the action does not depend on ϕ and γ separately, but only through φ. Therefore, a deformation mapping $\varphi = \phi \circ \gamma^{-1}$ that satisfies the Euler–Lagrange equations (68) should satisfy Equation (75) automatically. This can be checked through direct computation, as seen from replacing the identity

$$\frac{\mathrm{d}L}{\mathrm{d}Z^\mu} = \frac{\partial L}{\partial \varphi^i} \varphi^i_{,\mu} + \frac{\partial L}{\partial \varphi^i_{,\nu}} \varphi^i_{,\nu\mu} + \frac{\partial L}{\partial Z^\mu}$$

in Equation (75).

   The expression between parentheses in Equation (75) is known as Eshelby's energy-momentum tensor, i.e.

$$C^\mu_\nu = \frac{\partial L}{\partial \varphi^i_{,\mu}} \varphi^i_{,\nu} - L\delta^\mu_\nu \quad (76)$$

and Equation (75) states the balance of configurational forces.

   Noether's theorem: In addition to the conservation properties obtained from considering symmetries of the action acting on the deformed configuration, there are additional conserved quantities emanating from symmetries on the reference configuration. Consider a one parameter family of maps γ^ε that are a symmetry of the mechanical system, i.e. $S[\phi, \gamma^\varepsilon] = S[\phi, \gamma]$ for a fixed deformed configuration given by ϕ, where $\gamma^0 = \gamma$. Therefore, as in Section 5.2, we have

$$0 = \frac{\partial S}{\partial\varepsilon}\bigg|_{\varepsilon=0} = -\int_{\partial\mathcal{X}} \left(\frac{\partial L}{\partial \varphi^i_{,\mu}} \varphi^i_{,\nu} - L\delta^\mu_\nu \right) N_\mu \xi^\nu \, \mathrm{d}Z = -\int_{\partial\mathcal{X}} C^\mu_\nu N_\mu \xi^\nu \, \mathrm{d}Z \quad (77)$$

where we have used the definition of the symmetry on the left equality, and Equation (74) on the right one, assuming that γ satisfies the Euler–Lagrange equations (75). As before,

$$\xi^\nu = \frac{\partial\gamma^\varepsilon}{\partial\varepsilon}\bigg|_{\varepsilon=0}$$

is the infinitesimal generator of the symmetry.

5.4. Time symmetry and energy conservation

Energy conservation can be regarded in this context as a consequence of the translational symmetry along the time axis. We will see this in the ordinary differential equations and dynamic continuum mechanics cases separately.

Ordinary differential equations: Eshelby's energy-momentum tensor, in this case a scalar, simplifies to be the energy of the mechanical system. Consider then the family of mappings $\gamma^\varepsilon(t) = t + \varepsilon$, with infinitesimal generator $\xi = 1$. If the Lagrangian density does not depend explicitly on time so that the family of mappings is a symmetry of the mechanical system, we have from (77) that the energy at the beginning and the end of the time interval should be the same.

Dynamic continuum mechanics: The expression for Eshelby's momentum tensor is

$$
C = \begin{bmatrix} \dfrac{\rho_0}{2}\|\dot\varphi\|^2 + W & -\dot\varphi^{\mathsf{T}}P \\[2mm] \rho_0 F^{\mathsf{T}}\dot\varphi & -F^{\mathsf{T}}P + \left(W - \dfrac{\rho_0}{2}\|\dot\varphi\|^2\right)I \end{bmatrix}
$$

where I stands for the 3×3 identity matrix. Consider the family of mappings $\gamma^\varepsilon(t, X) = (t + \varepsilon, X)$, with the infinitesimal generator $\xi = (1, 0, 0, 0)$. If the Lagrangian density is not time dependent so that the family of mappings is a symmetry of the mechanical system, then by substituting the above expression in (77) we obtain

$$
\int_{\mathscr{B}} \left(\frac{\rho_0}{2}\|\dot\varphi\|^2 + W\right)\mathrm{d}X \bigg|_0^T - \int_0^T \int_{\partial\mathscr{B}} N_I P_i^I \dot\varphi^i \, \mathrm{d}X \, \mathrm{d}t = 0
$$

which is the familiar statement of energy balance between the initial and final times.

5.5. Reference symmetries and conserved path integrals

Dynamic continuum mechanics: In addition to the symmetry associated to time-axis translations, two path-independent integrals are obtained from analysing the effect of translations and rotations on the reference configuration. To do this, note that Equation (77) in Noether's theorem holds after replacing $\mathscr{X}$ by any of its subsets.

Firstly, let $\mathscr{C} \subset \mathscr{B}$ and $[t_1, t_2] \subset [0, T]$. Let $v \in \mathbb{R}^3$ be a unit vector and consider the family of mappings $\gamma^\varepsilon(t, X) = (t, X + \varepsilon v)$, with the infinitesimal generator $\xi = (0, v^1, v^2, v^3)$. If the Lagrangian density is not dependent on X, the family of mappings is a symmetry of the mechanical system. Therefore, from (77) we obtain

$$
v^J \left[\int_{\mathscr{C}} \rho_0 \dot\varphi_i F_J^i \, \mathrm{d}X \bigg|_{t_1}^{t_2} - \int_{t_1}^{t_2} \int_{\partial\mathscr{C}} \left[P_i^I F_J^i + \left(\frac{\rho_0}{2}\|\dot\varphi\|^2 - W\right)\delta_J^I \right] N_I \, \mathrm{d}X \, \mathrm{d}t \right] = 0 \tag{78}
$$

Since (78) is valid for all v, the expression in the large square brackets is identically zero, and it is valid for all subsets $\mathscr{C} \in \mathscr{B}$ and $[t_1, t_2] \subset [0, T]$. Formula (78), among other things, is the fundamental identity behind the definition of a spacetime path-independent dynamic J-integral, of utmost importance for dynamic fracture mechanics.

Secondly, consider the family of mappings $\gamma^\varepsilon(t, X) = (t, \exp(\varepsilon\Omega)X)$, where Ω is a skew-symmetric matrix. The infinitesimal generator is $\xi = (t, \omega \times X)$, where ω is axial vector

of Ω, $\omega = *\Omega$. If the Lagrangian density is independent of X, the family γ^ε is a symmetry whose conserved quantity is

$$\omega^K \left[\int_{\mathscr{C}} \varepsilon^{JKL} X_L \rho_0 \dot{\varphi}_i F_J^i \, dX \bigg|_{t_1}^{t_2} \right.$$

$$\left. - \int_{t_1}^{t_2} \int_{\partial\mathscr{C}} \varepsilon^{JKL} X_L \left[P_i^I F_J^i + \left(\frac{\rho_0}{2} \|\dot{\varphi}\|^2 - W \right) \delta_J^I \right] N_I \, dX \, dt \right] = 0 \qquad (79)$$

where ε^{JKL} is the Levi-Civita permutation tensor. Since Equation (79) is valid for all $\omega \in \mathbb{R}^3$, the expression in the large square brackets is equal to zero. Here too, it is valid for all subsets $\mathscr{C} \subset \mathscr{B}$ and $[t_1, t_2] \subset [0, T]$. The path independence of the dynamic fracture mechanics L-integral is obtained from Equation (79).

Static continuum mechanics: In this case, Eshelby's energy-momentum tensor reduces to

$$C_J^I = P_i^I F_J^i - W \delta_J^I \qquad (80)$$

By setting $\rho_0 = 0$ in Equations (78)–(79), and recalling that we are assuming no time dependence of the solution, we obtain that

$$\int_{\partial\mathscr{C}} (P_i^I F_J^i - W \delta_J^I) N_I \, dX = 0 \qquad (81)$$

and

$$\int_{\partial\mathscr{C}} \varepsilon^{JKL} X_L (P_i^I F_J^i - W \delta_J^I) N_I \, dX = 0 \qquad (82)$$

for all subsets $\mathscr{C} \subset \mathscr{B}$. Equations (81) and (82) are the path-independent integrals associated to the translational and rotational symmetries in the reference configuration. These identities are the basis for defining the fracture mechanics J- and L-integrals, respectively.

For the particular case in which the strain energy density W does not depend explicitly on X and is homogenous of degree m, the dimension of $\mathscr{B}$, an additional path-independent integral appears. Under these hypotheses, we have that $W(\lambda F) = \lambda^m W(F)$ for all $\lambda > 0$, and the potential energy is invariant under the one parameter family of transformations $\gamma^\lambda(X) = \lambda X$ of the reference configuration. To see this denote $Y = \lambda X$, and note that

$$\int_{\mathscr{C}} W(\nabla_X \varphi) \, dX = \int_{\lambda\mathscr{C}} W(\lambda \nabla_Y \varphi(Y/\lambda)) \frac{dY}{\lambda^m} = \int_{\lambda\mathscr{C}} W(\nabla_Y \varphi(Y/\lambda)) \, dY$$

The infinitesimal generator of this symmetry is $\xi = X$. After substituting this in (77) we obtain

$$\int_{\mathscr{C}} (P_i^I F_J^i - W \delta_J^I) N_I X^J \, dX = 0 \qquad (83)$$

which is the fundamental identity to derive the path independence of the M-integral.

A thorough discussion on configurational forces and conserved integrals can be found in References [49–52].

6. TIME ADAPTION AND DISCRETE PATH INDEPENDENT INTEGRALS

In perfect analogy to the continuous case, we obtain next the energy balance equation and configurational force equilibrium for the discrete system. As we shall see, this task is accomplished by identifying the discrete reference configuration $\mathcal{X}$ with the mesh used to approximate it, and by obtaining the Euler–Lagrange equations corresponding to continuously varying this mesh. We shall develop these ideas further subsequently.

6.1. Discrete energy conservation for ODEs

Consider first a synchronous, as opposed to asynchronous, discretization of an ordinary differential equation. The mesh on the reference configuration consists of a partition $0 < t_1 < \cdots < t_{N-1} < T$ of the interval $[0, T]$. The discrete action sum can be regarded as a function of both the co-ordinates $\{q_k\}$ and the set of times $\{t_k\}$. Alternatively, we may consider the discrete action sum as a function of generalized co-ordinates $\mathbf{q}_k = (t_k, q_k)$.

Next, as proposed in Reference [53], we extend the discrete variational principle to the extended configuration by regarding both the co-ordinates and the times on an equal footing. We request the discrete action sum to be stationary with respect to variations in the generalized co-ordinates $\mathbf{q}_k$, i.e. with respect to *both* the co-ordinates q_k and the times t_k. The resulting discrete Euler–Lagrange equations are

$$0 = D_2 L_d(q_{k-1}, q_k, t_k - t_{k-1}) + D_1 L_d(q_k, q_{k+1}, t_{k+1} - t_k) \tag{84}$$

$$0 = D_3 L_d(q_{k-1}, q_k, t_k - t_{k-1}) - D_3 L_d(q_k, q_{k+1}, t_{k+1} - t_k) \tag{85}$$

The first equation coincides of course with Equation (5). Of particular interest is Equation (85), which says that the quantity

$$E_k = -D_3 L_d(q_k, q_{k+1}, t_{k+1} - t_k) \tag{86}$$

should be conserved for all times, i.e. $E_k = E_{k+1}$. By analogy to the continuous case, E_k is the natural definition of discrete energy. To further motivate this definition, Equation (85) can be seen as the discrete analog to Equation (75), which expresses the conservation of Eshelby's energy-momentum tensor, the energy in the one-dimensional case, for a time-independent Lagrangian. E_k is therefore the discrete Eshelby's energy-momentum tensor.

As an example, for the discrete Lagrangian (3) the discrete energy is

$$E_d = -D_3 L_d(q_0, q_1, h) = \frac{1}{2} \left(\frac{q_1 - q_0}{h} \right)^\mathsf{T} M \left(\frac{q_1 - q_0}{h} \right) + V(q_0)$$

In contrast to the continuum case, in which the configurational forces balance equations (75) are automatically satisfied whenever the Euler–Lagrange equations (68) are, no such property holds in the discrete case. The extension of the discrete variational principle to the extended configuration adds new equations to the discrete mechanical system, namely equations (85).

The set of Equations (84) and (85) determines *both* the co-ordinates $\{q_k\}$ as well as the discrete times $\{t_k\}$. This results in *time adaption*, in as much as the time set $\{t_k\}$ is not prescribed at the outset but is determined as part of the solution instead.

Of course, the viability of the extended configuration approach relies on the solvability of the discrete Euler–Lagrange equations (84)–(85). However, Kane *et al.* [53] pointed out that it is not always possible to determine a positive time step from the discrete energy-conservation equation, especially near *turning points* where velocities are small. Kane *et al.* [53] overcame this difficulty by formulating a minimization problem that returns the exact solution whenever one exists.

The following simple example demonstrates that solvability cannot be always counted on, especially for explicit algorithms. The example concerns a simple harmonic oscillator with mass m and spring constant κ. For this system, the discrete Euler–Lagrange equations for the extended configuration corresponding to the discrete Lagrangian (3) are

$$p^{i+1/2} - p^{i-1/2} = -h^{i-1/2}\kappa q^i \tag{87}$$

$$\frac{1}{2m}(p^{i-1/2})^2 + \frac{1}{2}\kappa(q^i)^2 = \frac{1}{2m}(p^{i+1/2})^2 + \frac{1}{2}\kappa(q^{i+1})^2 = H \tag{88}$$

where

$$p^{i+1/2} = m\frac{q^{i+1} - q^i}{h^{i+1/2}} \tag{89}$$

and we write

$$h^{i+1/2} = t^{i+1} - t^i \tag{90}$$

It should be noted that Equation (87) describes a variable time-step central-difference scheme, and therefore the algorithm is *explicit*. In terms of the dimensionless variables

$$\hat{p} = \frac{p}{\sqrt{2mH}}, \quad \hat{q} = q\sqrt{\frac{\kappa}{H}}, \quad \hat{h} = \frac{h}{\sqrt{m/\kappa}} \tag{91}$$

Equations (87)–(89) may be recast in the form

$$\hat{p}^{i+1/2} - \hat{p}^{i-1/2} = -\hat{h}^{i-1/2}\hat{q}^i \tag{92}$$

$$(\hat{p}^{i-1/2})^2 + (\hat{q}^i)^2 = (\hat{p}^{i+1/2})^2 + (\hat{q}^{i+1})^2 = 1 \tag{93}$$

$$\hat{h}^{i+1/2} = \frac{\hat{q}^{i+1} - \hat{q}^i}{\hat{p}^{i+1/2}} \tag{94}$$

The problem is now to solve these equations for $(\hat{q}^{i+1}, \hat{p}^{i+1/2}, \hat{h}^{i+1/2})$, subject to the constraint $\hat{h}^{i+1/2} > 0$, given $(\hat{q}^i, \hat{p}^{i-1/2}, \hat{h}^{i-1/2})$, $\hat{h}^{i-1/2} > 0$.

This problem can readily be solved graphically in the phase plane $(\hat{q}, \hat{p}) \in \mathbb{R}^2$, Figure 17. Equation (93) defines a constant energy surface, which in the present case reduces to a circle, and Equation (92) defines the constant linear momentum surface, which here reduces to a horizontal line. The intersections of this line with the circle return two possible solutions of the system. The value of $\hat{h}^{i+1/2}$ is given by the inverse of the slope of the segment joining $(\hat{q}^i, 0)$ with $(\hat{q}^{i+1}, \hat{p}^{i+1/2})$. Valid solutions correspond to segments with positive slopes.
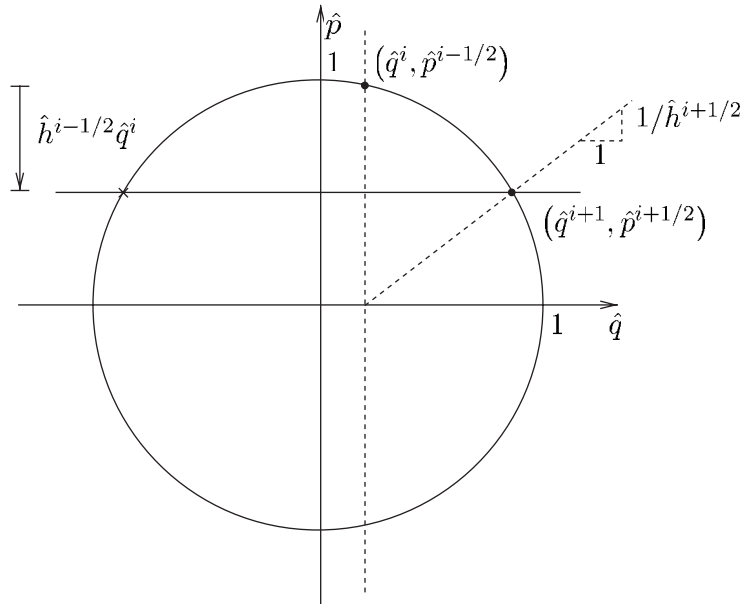
Figure 17. Graphical interpretation of the algorithm. There are two intersections of the constant energy and momentum surfaces. The cross denotes a solution rendering a negative value of $\hat{h}^{i+1/2}$, while the circle indicates the positive solution.

It is clear from this construction that solutions fail to exist for sufficiently large $|\hat{h}^{i-1/2}\hat{q}^i|$, as under such conditions the constant-linear-momentum line does not intersect the constant-energy circle. Since both $\hat{q}^i$ and $\hat{h}^{i-1/2}$ are given as initial conditions, this lack of solvability implies that the explicit algorithm may not be able to conserve energy over some time steps. It does not appear to be known at present whether it is always possible to formulate—most likely *implicit*—discrete Lagrangians such that the discrete Euler–Lagrange equations (84)–(85) are always solvable.

Time-dependent Lagrangians: When the Lagrangian depends explicitly on time, the value of the energy is not preserved but grows or decays accordingly. New terms will appear in the discrete energy balance equation (85) in this case, since it stems from taking derivatives with respect to the times of the discretization. This case is developed in detail in Reference [19].

6.2. Discrete energy conservation for AVIs

The additional freedom provided by AVIs to choose local time steps not only provides indisputable computational advantages but also sheds light into how energy is transferred between the different subsystems, or elements. In this case, we regard the discrete action sum as a function of the co-ordinates x_a^i of each particle a and the time of each subsystem t_K^j. In other words, we consider the action sum to be function of generalized co-ordinates $\mathbf{x}_a^i = (t_a^i, x_a^i)$, in which we restrict the time component t_a^i to coincide with the time t_K^j of one of the elements K. The extended discrete variational principle acting on the generalized co-ordinates results in

the discrete Euler–Lagrange equations

$$D_a^i S_d = 0 \tag{95}$$

$$D_K^j S_d = 0 \tag{96}$$

for all $t_a^i \in (0, T)$ and $t_K^j \in (0, T)$, where $D_K^j S_d$ indicates the partial derivative of the discrete action with respect to the elemental time t_K^j. The linear momentum balance equation (47) appears again as Equation (95). The second equation above, Equation (96), expresses a balance in the energy transfer between the different subsystems. The system of equations (95)–(96) should be satisfied by solving for *both* the co-ordinates x_a^i and the times t_K^j.

As an example, for the AVI algorithm described by the discrete action sum (45), equation (96) reads

$$T_K^{j-} + V_K(x_K^j) = T_K^{j+} + V_K(x_K^{j+1}) \tag{97}$$

In this expression,

$$T_K^{j-} \equiv \sum_{a \in K} \frac{1}{2} (v_{K,a}^{j-})^\mathsf{T} m_a v_{K,a}^{j-} \tag{98}$$

$$T_K^{j+} \equiv \sum_{a \in K} \frac{1}{2} (v_{K,a}^{j+})^\mathsf{T} m_a v_{K,a}^{j+} \tag{99}$$

are the kinetic energies carried by the nodes of K before and after the element strikes those nodes and changes their linear momenta according to (48), and $v_{K,a}^{j-}$ and $v_{K,a}^{j+}$ are the velocities of a node a in K also before and after t_K^j, respectively.

It is evident that Equation (97) encodes a discrete local balance of energy. It should be noted that the local energy balance equation (97) allows for energy transfer between the elements, as required. Specifically, the energy is transferred between elements by the conduit of the variations in kinetic energy. Thus, in general $T_K^{j+} \neq T_K^{(j+1)-}$, which implies that part of the kinetic energy is exchanged between neighbouring elements. In particular, $T_K^{(j+1)-} - T_K^{j+}$ equals the work done on element K between two successive elemental updates.

The relative ease with which the discrete dynamics formalism yields a complete and internally consistent set of equations of motion and balance equations, such as (48) and (97) for the discrete Lagrangian (46), can hardly be overstated. Thus, while in hindsight (48) and (97) are eminently reasonable, arriving at them without the benefit of a formal procedure would require uncommon intuition (or luck).

Energy conservation properties: The energy conservation properties of AVIs can be analysed by looking at the effects of translating the time axis on the action sum. Consider then the one-parameter family of elemental times given by $(t_K^j)^\varepsilon = t_K^j + \varepsilon$. If the discrete action sum is invariant under the action of this family we obtain that

$$0 = \frac{\partial}{\partial \varepsilon} S_d(x_K^j, (t_K^j)^\varepsilon)\Big|_{\varepsilon=0} = \sum_K \sum_{j=0}^{N_K} D_K^j S_d \tag{100}$$
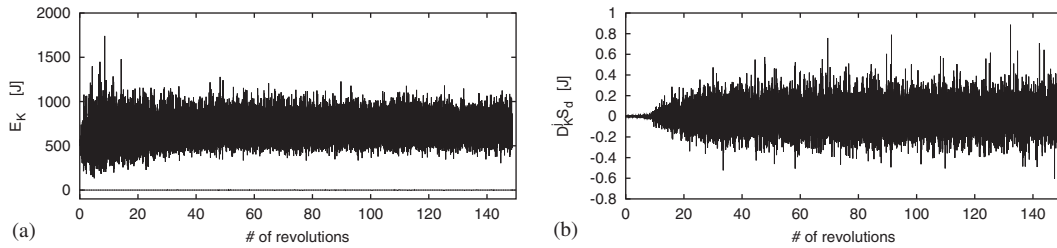
Figure 18. Evolution of the residual of the energy balance equation (96) (or energy 'imbalance') and the total energy of the same element for one element of the mesh chosen at random. Even though the residual of the energy equation is not exactly zero, the absolute size of the energy 'imbalance' is negligible when compared with the total energy in the element. The local energy balance is 'nearly' satisfied at all times, even though this equation is never enforced. This curves were obtained from case 3 of the helicopter blade example: (a) total energy (top) and residue of the energy equation (bottom) in one element; and (b) blow up of the residue of the energy equation shown in the figure on the left.

Next, note that if we use the extended variational principle, and solve for both times and co-ordinates, then $D_K^j S_d = 0$ for $j = 1, \ldots, N_K - 1$ and we obtain

$$E_d = \sum_K D_K^0 S_d = -\sum_K D_K^{N_K} S_d \tag{101}$$

Equation (101) expresses the exact conservation of the total energy E_d between times 0 and T, provided the system of equations (95)–(96) has a solution.

The parallel with the continuous system is insightful here too. After a close inspection, we note that Equation (100) is the discrete analog to Equation (74), which expresses the variations of the actions under variations of the reference configuration. While in the continuous case Equation (77) in Noether's theorem is a consequence of the Euler–Lagrange equations (68), we need to use the extended variational principle and obtain Equation (96) to recover the conserved quantity in the discrete case.

Solving for the time step is not necessary in practice: As seen from the numerical examples in Section 4.5, Figure 13, and in Reference [19], the total energy of the system oscillates around a constant value for very long times without overall growth or decay. These occurs in spite of keeping the time step in each element constant and not enforcing equation (96).

The residual of the energy equation (97), given by the left-hand side of Equation (96), is of considerable interest. The evolution of this residual in time, together with the elemental total energy as given by the left-hand side of Equation (97), is shown in Figure 18 for an element chosen at random. These curves were obtained from the third case of the helicopter blade example in Section 4.5. In this case, the time steps are chosen to satisfy the Courant condition instead of the energy balance in Equation (96). The absolute value of the energy 'imbalance' is negligible when compared with the total energy of the element, so for all practical purposes, the energy balance is 'nearly' satisfied at the local level as well.

These numerical tests suggest that both the local and global energy behaviour of the AVI algorithm are excellent even when Equation (96) is not enforced. We emphasize that these numerical results correspond to the case in which the various elemental time steps are specified at the initial time and are not changed as the simulation progresses. We have not investigated other strategies.
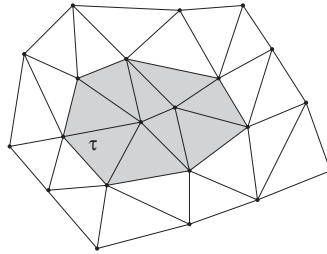
Figure 19. Example of a subset of elements τ in a finite element mesh. The elements in the shaded
region contribute with their elemental potential energy to the discrete action sum $S_{d,\tau}$.

6.3. Implementation of time-step adaption for AVIs

Even though the numerical evidence suggests that it is not necessary to solve the complete
system (95)–(96) for both the co-ordinates and the times in order to obtain energy conservation,
if one wished to do it, there is an algorithm that accomplishes this task. It is briefly described
below.

  The algorithm proceeds similarly to the one explained in Figure 7. The basic additional
idea is that after updating the positions and velocities of an element according to the linear
momentum balance equation, we should reschedule this element and all those sharing nodes
with it to satisfy the energy balance equation for each of them. In this way, the time t_K^{j+1} at
which any element K is extracted from the priority queue corresponds to the solution of the
energy balance equation $D_K^j S_d = 0$.

  Of course a strategy to overcome the occassional lack of solutions of the energy balance
equation is needed to ensure the computation of a time step, as described in Reference [53].

  The satisfaction of the energy balance equation introduces, as expected, a growth in the
computational cost of the algorithm, since the resulting equations are intrinsically non-linear in
the elemental time.

6.4. Conserved discrete path integrals

The theory of discrete mechanics provides a way to define discrete path integrals, and therefore
discrete J- and L-integrals, whose value is independent of the discrete path chosen. We begin
by identifying the finite element mesh on the undeformed shape of the continuum body as the
discrete reference configuration. The discrete parametric configuration can be identified with
several copies of the reference finite element. Then, the discrete version of the mapping γ in
Section 5.3 is constructed with the isoparametric mappings from the reference element to each
of the elements in the mesh, while the discrete analogue to the mapping ϕ is the traditional finite
element isoparametric interpolation. The variables of the isoparametric mapping are of course
the positions X_a of the nodes in the reference configuration, while those of the isoparametric
interpolation are the positions of the nodes x_a in the deformed one (Figure 19).

  Let $\tau \subset \mathcal{T}$ be a subset of elements of the finite element mesh, and let

$$S_{d,\tau} = \sum_{K \in \tau} V_K = \sum_{K \in \tau} \int_K W(\nabla \varphi_h, X)\, \mathrm{d}X$$

be the discrete action sum, in this case the potential energy, corresponding to τ. We regard now $S_{d,\tau}$ as a function of (x_a, X_a). Consider next the one-parameter family of mappings $(X_a)^\varepsilon = \exp(\Omega\varepsilon)X_a + \varepsilon v$, for a skew-symmetric matrix Ω and a vector $v \in \mathbb{R}^3$. If the strain-energy density W does not depend explicitly on X, i.e. the material is homogeneous, then the discrete action sum is invariant under the rotation or translation of the mesh with the family of mappings $(X_a)^\varepsilon$. This uses the fact that W is assumed to be material frame indifferent. Therefore,

$$0 = \left.\frac{\partial}{\partial \varepsilon} S_{d,\tau}\right|_{\varepsilon=0} = \sum_{a\in\tau} \frac{\partial S_{d,\tau}}{\partial X_a} v + \sum_{a\in\tau} X_a \times \frac{\partial S_{d,\tau}}{\partial X_a} \omega \tag{102}$$

where $\omega \in \mathbb{R}^3$ is the axial vector of Ω, which in terms of the Hodge star operator is $\omega = *\Omega$. Since this holds for all v and ω, we have that

$$0 = \sum_{a\in\tau} \frac{\partial S_{d,\tau}}{\partial X_a} \tag{103}$$

$$0 = \sum_{a\in\tau} X_a \times \frac{\partial S_{d,\tau}}{\partial X_a} \tag{104}$$

As we have seen with the energy conservation, in the discrete case we need to request the equilibrium of configurational forces, the analogue to Equation (75), separately, since it is not implied by the discrete Euler–Lagrange equations. Equilibrating the discrete configurational forces by repositioning the nodes of the mesh has originally been proposed, analysed and implemented by Thoutireddy and Ortiz [54]. Following their development, we request then

$$\frac{\partial S_d}{\partial X_a} = 0 \tag{105}$$

for all nodes a in $\mathscr{T} \setminus \partial\mathscr{B}$. Nodes on the boundary of the domain admit constrained variations only, since they should preserve the original geometry of the reference configuration. We note that S_d in Equation (105) is the action sum over the whole mesh, not only over τ. For a node a in the interior of τ, we have that $\partial S_d/\partial X_a = \partial S_{d,\tau}/\partial X_a$. This is not the case for a node a on the boundary of τ, since S_d may contain contributions of elements having a as a node which are not in τ. By eliminating the corresponding terms in Equations (103) and (104) we are left with

$$0 = \sum_{a\in\partial\tau} \frac{\partial S_{d,\tau}}{\partial X_a} \tag{106}$$

$$0 = \sum_{a\in\partial\tau} X_a \times \frac{\partial S_{d,\tau}}{\partial X_a} \tag{107}$$

Since Equations (106) and (107) are valid independently of the choice of τ, we obtain the discrete analogues to Equations (81), (82), respectively. The definition of the discrete path-independent J- and L-integral follows directly from here.
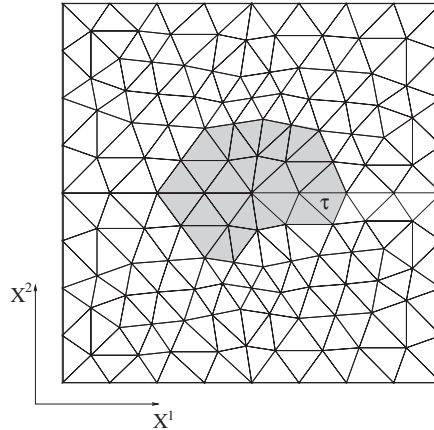
Figure 20. Example finite element mesh for a two-dimensional fracture mechanics simulation. If the configurational forces are in equilibrium, then expression (106) evaluated on $\partial\tau$ is equal to the J-integral. The value of the J-integral is the same when computed on the boundary of any submesh τ that contains the node at the crack tip.

As an application of (106), we discuss next the computation of the value of the J-integral on a two-dimensional crack tip (see Figure 20). As proposed in Reference [54], the value of the J-integral can be computed, from a finite element elastostatics simulation, as the configurational force conjugate to varying the node at the crack tip in the direction of crack advance, in this case the X^1-direction. We consider then a submesh τ containing the node at the crack tip, which we denote b. From (106) we have that

$$J = \frac{\partial S_{d,\tau}}{\partial X_b^1} = -\sum_{a\in\partial\tau\setminus b}\frac{\partial S_{d,\tau}}{\partial X_a^1}$$

The sum on the right-hand side includes null contributions from those nodes that belong to the crack faces. These nodes can be varied on the X^1 direction, since this preserves the geometry, and therefore the X^1-component of their conjugate configurational force should be zero. Only those nodes on $\partial\tau$ that do not belong to the crack face will contribute to the right-hand side, in an exact analogy to the continuous J-integral.

APPENDIX A: PROOF OF AVI CONVERGENCE

In this section, we prove that the asynchronous variational integrators discussed in Section 4 converge as the time steps go to zero. We only consider convergence of the time stepping, and we take the spatial discretization to be fixed. This analysis applies equally well if AVI methods are applied directly to an ODE problem. In this section, we denote time steps with the symbol h.

There are three key ideas needed for the proof:

1. The generalization of AVI methods to *asynchronous splitting methods*, which provide a simple framework to consider convergence of the time stepping. This explicitly discards all information about the spatial discretization, focussing attention on the time stepping.
2. The formulation of a reasonable *proxy system*, to which convergence can be proven. For standard integration techniques, after N steps of size h one proves that the integrator is close to the true flow map for time $T = Nh$. For asynchronous methods this is no longer correct, and in these notes we introduce Equation (A10) below as the proxy.
3. The condition under which convergence occurs. For standard time-stepping methods we consider convergence as the time step goes to zero. For asynchronous methods, however, it is possible for all time steps to go to zero but for the method to not converge to the true flow. We introduce the concept of *maximum asynchronicity* and we prove convergence as this tends to zero. For AVI methods this translates into a proof of convergence as the maximum time step goes to zero.

The technical assumptions made in this section are relatively standard. We work on unbounded Euclidean spaces $\mathbb{R}^n$ and assume that all vector fields are globally Lipschitz. Only autonomous systems are considered, as time dependancy can be included by adding a dimension with uniform flow for time. We only consider non-negative time steps here, and leave the general case for future work.

A.1. Asynchronous splitting methods (ASMs)

We will be concerned with systems on $\mathbb{R}^n$ of the form

$$\dot{x}(t) = f(x(t)) = \sum_{i=1}^{M} f_i(x(t)) \tag{A1a}$$

$$x(0) = x_0 \tag{A1b}$$

where the vector field f is the sum of M component vector fields f_i. We will denote by $\Phi^t : \mathbb{R}^n \to \mathbb{R}^n$ the flow map of f and by $\Phi_i^t : \mathbb{R}^n \to \mathbb{R}^n$ the flow map of f_i.

We assume that the component vector fields f_i are Lipschitz with constant L, so that

$$\| f_i(x) - f_i(y) \| \leqslant L \| x - y \| \quad \text{for } i = 1, \dots, M \tag{A2}$$

for all $x, y \in \mathbb{R}^n$. This implies that f is Lipschitz with constant ML. The flow maps Φ_i and Φ thus exist and are continuously differentiable for all t (Reference [55]).

An asynchronous splitting method for system (108) consists of M integrators $\Psi_i^h : \mathbb{R}^n \to \mathbb{R}^n$ for the component vector fields f_i together with a sequence of time steps $h_k \geqslant 0$ and indices i_k for $k = 1, \dots, N$. The method is then defined by

$$y_k = \Psi_{i_k}^{h_k}(y_{k-1}) \quad \text{for } k = 1, \dots, N \tag{A3a}$$

$$y_0 = x_0 \tag{A3b}$$

We define the cumulative time for component i to be

$$t_k^i = \sum_{j=1}^{k} \delta_{i_j}^i h_j \tag{A4}$$

and the global minimum time as

$$t_k^{\min} = \min_{i=1,\dots,M} t_k^i \tag{A5}$$

Observe that

$$t_{k+1}^{\min} - t_k^{\min} \leqslant h_{k+1} \quad \text{for } k = 1,\dots,N-1 \tag{A6}$$

The *maximum asynchronicity* is the smallest h such that

$$t_k^i - t_k^{\min} \leqslant h \quad \text{for } i = 1,\dots,M, \quad k = 1,\dots,N \tag{A7a}$$

$$h_k \leqslant h \quad \text{for } k = 1,\dots,N \tag{A7b}$$

Note that this implies that

$$t_{k+1}^i - t_k^{\min} \leqslant h \quad \text{for } i = 1,\dots,M, \quad k = 1,\dots,N-1 \tag{A8}$$

We will assume that each integrator Ψ_i is consistent, so that there is a Lipschitz function $C_\Psi : \mathbb{R}^n \to \mathbb{R}$ with constant L_Ψ, and $h^* \in \mathbb{R}$ such that

$$\|\Psi_i^t(x) - \Phi_i^t(x)\| \leqslant t^2 C_\Psi(x) \quad \text{for all } t \leqslant h^* \tag{A9}$$

for each $i = 1,\dots,M$ and all $x \in \mathbb{R}^n$.

In common with standard synchronous splitting methods, asynchronous splitting methods are geometric in that conservation properties of the component Ψ_i are inherited by the overall integrator, as it is simply a composition. For example, if each Ψ_i is a symplectic map or if each Ψ_i preserves a quantity $C(x)$ then the complete method will also be symplectic or C preserving, respectively.

The *proxy system* to which we will show convergence is defined by

$$z_k = \Phi^{t_k^{\min}}(x_0) + \sum_{i=1}^{M} (t_k^i - t_k^{\min}) f_i(\Phi^{t_k^{\min}}(x_0)) \tag{A10a}$$

$$z_0 = x_0 \tag{A10b}$$

Observe that as the maximum asynchronicity h tends to zero, the proxy system will converge to the true flow map Φ^t.

A.2. AVIs as ASMs

AVIs are a special case of asynchronous splitting methods (ASMs). Here we use the notation from Section 4, extended to allow the possibility that the elemental time steps vary over time.

Let Θ be the set of all elemental times, indexed sequentially so that $\Theta = \{t_a\}_{a=0}^{N_\Theta}$, and modified so that the initial time $t_0 = 0$ only appears once. Thus, $t_a \geqslant t_{a-1}$ for all $a = 1, \ldots, N_\Theta$. Let $\{K_a\}_{a=1}^{N_\Theta}$ be the sequence of element indices and $\{\Delta t_a\}_{a=1}^{N_\Theta}$ be the sequence of element time steps, so that for each $a = 1, \ldots, N_\Theta$ there is a j with $1 \leqslant j \leqslant N_K$ such that $t_a = t_{K_a}^j$ and $\Delta t_a = t_{K_a}^j - t_{K_a}^{j-1}$.

The spatially discretized system for AVIs is given by

$$\begin{pmatrix} \dot{x} \\ \dot{p} \end{pmatrix} = f(x, p) = \begin{pmatrix} M^{-1}p \\ -\sum_{K \in \mathcal{T}} \dfrac{\partial V_K}{\partial x} \end{pmatrix} \tag{A11}$$

We consider the decomposition of f into $M = N_{\mathcal{T}} + 1$ vector fields f_i given by

$$f_i(x, p) = \begin{pmatrix} 0 \\ -\dfrac{\partial V_i}{\partial x} \end{pmatrix} \quad \text{for } i = 1, \ldots, N_{\mathcal{T}} \tag{A12a}$$

$$f_{N_{\mathcal{T}}+1}(x, p) = \begin{pmatrix} M^{-1}p \\ 0 \end{pmatrix} \tag{A12b}$$

where $N_{\mathcal{T}}$ is the number of elements in $\mathcal{T}$. A sufficient condition for the vector fields f_i to be Lipschitz is that $D^2 V$ is uniformly bounded. This is satisfied in the linear elastic case for example. We now consider $N = 2N_\Theta$ time steps and indices specified by

$$h_{k-1} = t_a - t_{a-1}, \quad h_k = \Delta t_a \tag{A13a}$$

$$i_{k-1} = N_{\mathcal{T}} + 1, \quad i_k = K_a \tag{A13b}$$

where $k = 2a$ for $a = 1, \ldots, N_\Theta$. The method thus consists of alternate global 'drift' and local 'kick' steps.

The fact that the time sequence Θ is increasing means that the maximum asynchronicity h is equal to the maximum time step

$$h = \max_{K \in \mathcal{T}} \max_{j=1, \ldots, N_K} \left(t_K^j - t_K^{j-1} \right) \tag{A14}$$

The maximum asynchronicity will thus tend to zero whenever the maximum elemental time step tends to zero.

A.3. Convergence proof

We first recall variants of Gronwall's inequality in discrete and continuous time.

Lemma A.1
Consider a sequence $\{e_k\}_{k=0}^N$ of real numbers satisfying

$$e_k \leqslant A_k e_{k-1} + B_k \quad \text{for } k = 1, \ldots, N$$

where $e_0 \geqslant 0$, $A_k \geqslant 1$ and $B_k \geqslant 0$ for all $k = 1, \ldots, N$. Then

$$e_k \leqslant \left(e_0 + \sum_{i=1}^k B_i \right) \prod_{i=1}^k A_i$$

Lemma A.2
Consider a continuous real valued function $e(t)$ which satisfies

$$e(t) \leqslant B + \int_0^t A e(\tau) \, \mathrm{d}\tau \quad \text{for } 0 \leqslant t \leqslant T$$

for some non-negative constants A, B, T. Then

$$e(t) \leqslant B \mathrm{e}^{At} \quad \text{for } 0 \leqslant t \leqslant T$$

Now we recall some standard bounds on flow maps of Lipschitz vector fields.

Lemma A.3
Given a vector field f on $\mathbb{R}^n$ which is globally Lipschitz with constant L and which has flow map Φ, then

$$\|\Phi^t(x) - x\| \leqslant t \|f(x)\| \mathrm{e}^{Lt} \tag{A15}$$

$$\|\Phi^t(x) - \Phi^t(y)\| \leqslant \|x - y\| \mathrm{e}^{Lt} \tag{A16}$$

$$\|\Phi^t(x) - [x + tf(x)]\| \leqslant Lt^2 \|f(x)\| \mathrm{e}^{Lt} \tag{A17}$$

for all $x, y \in \mathbb{R}^n$ and all $t \geqslant 0$.

Next we make precise the sense in which the proxy system is close to the true flow.

Lemma A.4
The proxy system satisfies

$$\|\Phi^{t_k^{\min}}(x_0) - x_0\| \leqslant t_k^{\min} \mathrm{e}^{MLt_k^{\min}} \sum_{i=1}^M \|f_i(x_0)\| \tag{A18a}$$

$$\|z_k - \Phi^{t_k^{\min}}(x_0)\| \leqslant h(1 + MLt_k^{\min} \mathrm{e}^{MLt_k^{\min}}) \sum_{i=1}^M \|f_i(x_0)\| \tag{A18b}$$

Proof
The bound (A18a) follows directly from (A15), which together with the definition (A10a) then gives (A18b). $\square$
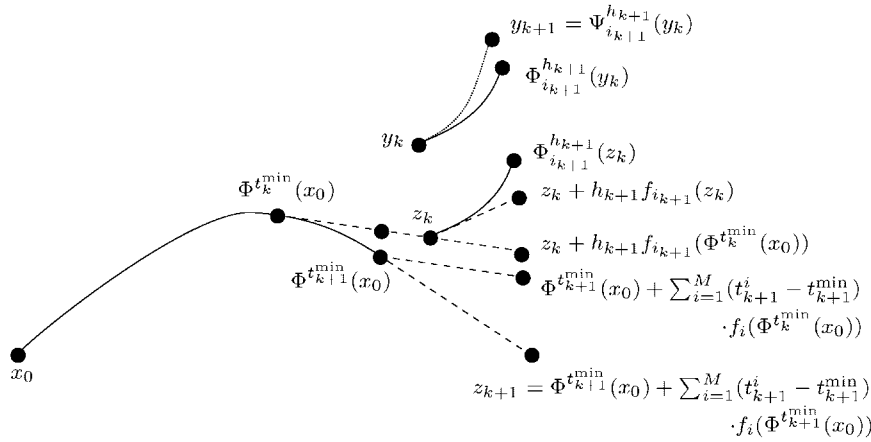
Figure A1. Decomposition of the error used in Lemma A.5.

Now we are ready to prove convergence of asynchronous splitting methods. We derive a bound on the incremental error, using the decomposition illustrated in Figure A1.

Lemma A.5
The difference between the computed solution and the proxy system at time step $k+1$ satisfies

$$\|y_{k+1} - z_{k+1}\| \leqslant e^{(L+2L_\Psi)h_{k+1}} \|y_k - z_k\|$$

$$+ h_{k+1}h \left[C_\Psi(x_0) + C_B(L, L_\Psi, M, t_k^{\min}) \sum_{i=1}^{M} \|f_i(x_0)\| \right] \quad (A19)$$

where C_B is a smooth function which remains bounded as its arguments tend to zero.

Proof
Using the definitions of y_{k+1} and z_{k+1} we decompose the error at step $k+1$ as

$$\|y_{k+1} - z_{k+1}\| \leqslant \left\| \Psi_{i_{k+1}}^{h_{k+1}}(y_k) - \Phi_{i_{k+1}}^{h_{k+1}}(y_k) \right\| \quad (A20a)$$

$$+ \left\| \Phi_{i_{k+1}}^{h_{k+1}}(y_k) - \Phi_{i_{k+1}}^{h_{k+1}}(z_k) \right\| \quad (A20b)$$

$$+ \left\| \Phi_{i_{k+1}}^{h_{k+1}}(z_k) - [z_k + h_{k+1}f_{i_{k+1}}(z_k)] \right\| \quad (A20c)$$

$$+ \left\| [z_k + h_{k+1}f_{i_{k+1}}(z_k)] \right.$$

$$\left. - [z_k + h_{k+1}f_{i_{k+1}}(\Phi^{t_k^{\min}}(x_0))] \right\| \quad (A20d)$$

$$+ \left\| [z_k + h_{k+1} f_{i_{k+1}}(\Phi^{t_k^{\min}}(x_0))] \right.$$

$$\left. - \left[\Phi^{t_{k+1}^{\min}}(x_0) + \sum_{i=1}^{M} (t_{k+1}^i - t_{k+1}^{\min}) f_i(\Phi^{t_k^{\min}}(x_0)) \right] \right\| \qquad \text{(A20e)}$$

$$+ \left\| \left[\Phi^{t_{k+1}^{\min}}(x_0) + \sum_{i=1}^{M} (t_{k+1}^i - t_{k+1}^{\min}) f_i(\Phi^{t_k^{\min}}(x_0)) \right] \right.$$

$$\left. - \left[\Phi^{t_{k+1}^{\min}}(x_0) + \sum_{i=1}^{M} (t_{k+1}^i - t_{k+1}^{\min}) f_i(\Phi^{t_{k+1}^{\min}}(x_0)) \right] \right\| \qquad \text{(A20f)}$$

Now (A20e) is equal to

$$\| [\Phi^{t_k^{\min}}(x_0) + (t_{k+1}^{\min} - t_k^{\min}) f(\Phi^{t_k^{\min}}(x_0))] - \Phi^{(t_{k+1}^{\min} - t_k^{\min})}(\Phi^{t_k^{\min}}(x_0)) \|$$

and (A20f) is equal to

$$\left\| \sum_{i=1}^{M} (t_{k+1}^i - t_{k+1}^{\min})[f_i(\Phi^{t_{k+1}^{\min}}(x_0)) - f_i(\Phi^{t_k^{\min}}(x_0))] \right\|$$

and so using (A9), (A16), (A17), (A2), (A7a) and (A6) we obtain

$$\| y_{k+1} - z_{k+1} \| \leqslant h_{k+1}^2 C_{\Psi}(y_k) \qquad \text{(A21a)}$$

$$+ \| y_k - z_k \| \mathrm{e}^{Lh_{k+1}} \qquad \text{(A21b)}$$

$$+ L h_{k+1}^2 \mathrm{e}^{Lh_{k+1}} \| f_{i_{k+1}}(z_k) \| \qquad \text{(A21c)}$$

$$+ h_{k+1} L \| z_k - \Phi^{t_k^{\min}}(x_0) \| \qquad \text{(A21d)}$$

$$+ M L h h_{k+1} \mathrm{e}^{MLh_{k+1}} \| f(\Phi^{t_k^{\min}}(x_0)) \| \qquad \text{(A21e)}$$

$$+ h h_{k+1} \mathrm{e}^{Lh_{k+1}} \sum_{i=1}^{M} \| f_i(\Phi^{t_k^{\min}}(x_0)) \| \qquad \text{(A21f)}$$

where (A21a)–(A21f) are bounds for (A20a)–(A20f), respectively.

We now recall that for any Lipschitz function $a : \mathbb{R}^n \to \mathbb{R}^m$ with constant L_a, we have $\| a(y) \| \leqslant \| a(x) \| + L_a \| y - x \|$ for all $x, y \in \mathbb{R}^n$. Using this we can substitute the bounds in Lemma A.4 into (A21) to compute

$$\| y_{k+1} - z_{k+1} \| \leqslant (\mathrm{e}^{Lh_{k+1}} + h_{k+1}^2 L_{\Psi}) \| y_k - z_k \|$$

$$+ h_{k+1} h \left(C_{\Psi}(x_0) + [L_{\Psi} h + L \mathrm{e}^{Lh_{k+1}} + L^2 h \mathrm{e}^{Lh_{k+1}} + \mathrm{e}^{Lh_{k+1}} \right.$$

$$+ MLe^{MLh_{k+1}} + L + (L\Psi(hML + 1) + ML^2$$

$$+ L^2e^{Lh_{k+1}}(hML + 1) + M^2L^2e^{MLh_{k+1}}$$

$$+ MLe^{MLh_{k+1}})t_k^{\min}e^{MLt_k^{\min}}]\sum_{i=1}^{M}\|f_i(x_0)\|\Big)$$

Finally, we note that $e^{Lh_{k+1}} + h_{k+1}^2 L\Psi \leqslant e^{(L+2L\Psi)h_{k+1}}$ and we use the restrictions $h \leqslant 1/L$ and $h \leqslant 1/L\Psi$ to see that the function C_B depends only on L, $L\Psi$, M and $t_k^{\min}$ and remains bounded as its arguments tend to zero. This gives the desired result. $\qquad\square$

Lemma A.6
The difference between the computed solution and the proxy system at time step k satisfies

$$\|y_k - z_k\| \leqslant h(t_k^{\min} + h)Me^{(L+2L\Psi)M(t_k^{\min}+h)}\left[C\Psi(x_0) + C_B(L, L\Psi, M, t_k^{\min})\sum_{i=1}^{M}\|f_i(x_0)\|\right] \quad \text{(A22)}$$

Proof
This follows from applying Lemma A.1 to the estimate in Lemma A.5 and observing that from (A4) and (A7) we have

$$\sum_{j=1}^{k} h_j = \sum_{i=1}^{M} t_k^i \leqslant \sum_{i=1}^{M} (t_k^{\min} + h) = M(t_k^{\min} + h)$$

$\qquad\square$

Theorem A.7
Take a sequence of index and time step specifications $\{i_k^\alpha, h_k^\alpha\}_{k=1}^{N^\alpha}$ for $\alpha \in \mathbb{Z}^+$ with asynchronicity bounds h^α. Assume that $h^\alpha \to 0$ and $t_N^{\min} \to T$ as $\alpha \to \infty$. Let $\{y_k\}_{k=1}^{N^\alpha}$ be the sequence generated by the asynchronous splitting method with initial condition $y_0 = x_0$. Then

$$\lim_{\alpha \to \infty} \|y_{N^\alpha} - x(T)\| = 0$$

Proof
We decompose the global error to give

$$\|y_{N^\alpha} - x(T)\| \leqslant \|y_{N^\alpha} - z_{N^\alpha}\| + \|z_{N^\alpha} - \Phi^{t_{N^\alpha}^{\min}}(x_0)\| + \|\Phi^{t_{N^\alpha}^{\min}}(x_0) - \Phi^T(x_0)\|$$

The three components each tend to zero as $\alpha \to \infty$, as can be seen from the bounds in Lemmas A.4 and A.6, thus giving the result. $\qquad\square$

Corollary A.8 (Restatement of Theorem 4.1)
Consider a sequence of asynchronous variational integrators for the same spatial discretization with maximum time step $h \to 0$ and maximum final time $t_{N_\Theta} \to T$. Then the final configuration converges to the exact solution $\Phi^T(x_0, p_0)$.

Proof
We have that $t_{N_\Theta} - h \leqslant t_{N_\Theta}^{\min} \leqslant t_{N_\Theta}$ and so $t_{N_\Theta}^{\min} \to T$. The result then follows from Theorem A.7. $\qquad\square$

REFERENCES

1. Johnson C. *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Cambridge University Press: Cambridge, MA, 1987.
2. Hughes TJR. *The Finite Element Method*: *Linear Static and Dynamic Finite Element Analysis*. Prentice-Hall: Englewood Cliffs, NJ, 1987.
3. Channell PJ, Scovel C. Symplectic integration of Hamiltonian systems. *Nonlinearity* 1990; **3**(2):231–259.
4. Rowley CW, Marsden JE. Variational integrators for point vortices. *Proceedings of the CDC*, vol. 40, Las Vegas, NV, 2002.
5. Hairer E, Lubich C. Long-time energy conservation of numerical methods for oscillatory differential equations. *SIAM Journal of Numerical Analysis* 2000; **38**(2):414–441 (electronic). ISSN 1095-7170.
6. Marsden JE, West M. Discrete mechanics and variational integrators. In *Acta Numerica*, vol. 10. Cambridge University Press: Cambridge, MA, 2001.
7. Ge Z, Marsden JM. Lie–Poisson integrators and Lie–Poisson Hamilton–Jacobi theory. *Physics Letters A* 1988; **133**:134–139.
8. Gonzalez O, Simo JC. On the stability of symplectic and energy-momentum algorithms for non-linear Hamiltonian systems with symmetry. *Computer Methods in Applied Mechanics and Engineering* 1996; **134**(3–4):197–222.
9. Hairer E, Lubich C, Wanner G. *Geometric Numerical Integration*: *Structure-Preserving Algorithms for Ordinary Differential Equations*. Springer: Berlin, 2002.
10. MacKay R. Some aspects of the dynamics of Hamiltonian systems. In *The Dynamics of Numerics and the Numerics of Dynamics*, Broomhead DS, Iserles A (eds). Clarendon Press: Oxford, 1992; 137–193.
11. Sanz-Serna JM. The numerical integration of Hamiltonian systems. In *Computational Ordinary Differential Equations*, Cash JR, Gladwell I (eds). Clarendon Press: Oxford, 1992; 437–449.
12. Ryckaert J, Ciccotti G, Berendsen H. Numerical integration of the cartesian equations of motion of a system with constraints: Molecular dynamics of n-alkanes. *Journal of Computational Physics* 1977; **23**:327–341.
13. Anderson H. Rattle: a velocity version of the shake algorithm for molecular dynamics calculations. *Journal of Computational Physics* 1983; **52**:24–34.
14. Leimkuhler BJ, Skeel RD. Symplectic numerical integrators in constrained Hamiltonian-systems. *Journal of Computational Physics* 1994; **112**(1):117–125.
15. Kane C, Repetto EA, Ortiz M, Marsden JE. Finite element analysis of nonsmooth contact. *Computer Methods in Applied Mechanics and Engineering* 1999; **180**:1–26.
16. Pandolfi A, Kane C, Marsden JE, Ortiz M. Time-discretized variational formulation of non-smooth frictional contact. *International Journal for Numerical Methods in Engineering* 2002; **53**:1801–1829.
17. Fetecau R, Marsden JE, Ortiz M, West M. Nonsmooth Lagrangian mechanics, 2002, preprint.
18. Marsden JE, Patrick GW, Shkoller S. Multisymplectic geometry, variational integrators, and nonlinear PDEs. *Communications in Mathematical Physics* 1998; **199**(2):351–395.
19. Lew A, Marsden JE, Ortiz M, West M. Asynchronous variational integrators. *Archive for Rational Mechanics and Analysis* 2003; **2**:85–146.
20. Neal MO, Belytschko T. Explicit–explicit subcycling with non-integer time step ratios for structural dynamic systems. *Computers and Structures* 1989; **6**:871–880.
21. Belytschko T, Mullen R. Mesh partitions of explicit-implicit time integrators. In *Formulations and Computational Algorithms in Finite Element Analysis*, Bathe K-J, Oden JT, Wunderlich W (eds). MIT Press: Cambridge, MA, 1976; 673–690.
22. Belytschko T. Partitioned and adaptive algorithms for explicit time integration. In *Nonlinear Finite Element Analysis in Structural Mechanics*, Wunderlich W, Stein E, Bathe K-J (eds). Springer: Berlin, 1981; 572–584.
23. Smolinski P, Wu Y-S. An implicit multi-time step integration method for structural dynamics problems. *Computational Mechanics* 1998; **22**:337–343.

24. Hughes TJR, Liu WK. Implicit-explicit finite elements in transient analysis: stability theory. *Journal of Applied Mechanics* 1978; **78**:371–374.
25. Hughes TJR, Pister KS, Taylor RL. Implicit-explicit finite elements in nonlinear transient analysis. *Computer Methods In Applied Mechanics And Engineering* 1979; **17/18**:159–182.
26. Grubmüller H, Heller H, Windemuth A, Schulten K. Generalized Verlet algorithm for efficient molecular dynamics simulations with long-range interactions. *Molecular Simulation* 1991; **6**:121–142.
27. Tuckerman M, Berne BJ, Martyna GJ. Reversible multiple time scale molecular dynamics. *The Journal of Chemical Physics* 1992; **97**:1990–2001.
28. Kane C, Marsden JE, Ortiz M, West M. Variational integrators and the Newmark algorithm for conservative and dissipative mechanical systems. *International Journal for Numerical Methods in Engineering* 2000; **49**(10):1295–1325.
29. Newmark N. A method of computation for structural dynamics. *ASCE Journal of the Engineering Mechanics Division* 1959; **85**(EM 3):67–94.
30. Simo JC, Tarnow N, Wong KK. Exact energy-momentum conserving algorithms and symplectic schemes for nonlinear dynamics. *Computer Methods in Applied Mechanics and Engineering* 1992; **100**(1):63–116.
31. Skeel RD, Zhang GH, Schlick T. A family of symplectic integrators: Stability, accuracy, and molecular dynamics applications. *SIAM Journal on Scientific Computing* 1997; **18**(1):203–222.
32. Skeel RD, Srinivas K. Nonlinear stability analysis of area-preserving integrators. *SIAM Journal on Numerical Analysis* 2000; **38**(1):129–148.
33. Bottasso CL. A new look at finite elements in time: a variational interpretation of Runge–Kutta methods. *Applied Numerical Mathematics* 1997; **25**(4):355–368.
34. Marsden JE, Ratiu T. *Introduction to Mechanics and Symmetry*, *vol. 17 of Texts in Applied Mathematics*. Springer: Berlin, Second Edition, 1999.
35. Reich S. Backward error analysis for numerical integrators. *SIAM Journal on Numerical Analysis* 1999; **36**(5):1549–1570.
36. Marsden JE, Hughes TJR. *Mathematical Foundations of Elasticity*. Dover Publications: New York, 1994.
37. Knuth D. *The Art of Computer Programming*. Addison-Wesley: Reading, MA, 1998.
38. Borri M. Helicopter rotor dynamics by finite-element time approximation. *Computers and Mathematics with Applications—Part A* 1986; **12**(1):149–160.
39. Friedmann PP. Numerical-methods for the treatment of periodic systems with applications to structural dynamics and helicopter rotor dynamics. *Computers and Structures* 1990; **35**(4):329–347.
40. Sheng G, Fung TC, Fan SC. Parametrized formulations of Hamilton's law for numerical solutions of dynamic problems: Part II. Time finite element approximation. *Computational Mechanics* 1998; **21**(6):449–460.
41. Armero F, Romero I. 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. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**:2603–2649.
42. Armero F, Romero I. On the formulation of high-frequency dissipative time-stepping algorithms for nonlinear dynamics. Part II: second-order methods. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**:6783–6824.
43. Bottasso CL, Bauchau OA. Multibody modeling of engage and disengage operations of helicopter rotors. *Journal of the American Helicopter Society* 2002; **46**:290–300.
44. Izaguirre JA, Reich S, Skeel RD. Longer time steps for molecular dynamics. *Journal of Chemical Physics* 1999; **110**(20):9853–9864.
45. Bishop TC, Skeel RD, Schulten K. Difficulties with multiple time stepping and fast multipole algorithm in molecular dynamics. *Journal of Computational Chemistry* 1997; **18**(14):1785–1791.
46. Biesiadecki JJ, Skeel RD. Dangers of multiple time steps methods. *Journal of Computational Physics* 1993; **109**:318–328.
47. Daniel WJT. Analysis and implementation of a new constant acceleration subcycling algorithm. *International Journal for Numerical Methods in Engineering* 1997; **40**:2841–2855.
48. Daniel WJT. The subcycled newmark algorithm. *Computational Mechanics* 1997; **20**:272–281.
49. Gurtin ME. *Configurational Forces as Basic Concepts of Continuum Physics*. Springer: Berlin, 2000.
50. Knowles JK, Sternberg E. On a class of conservation laws in linearized and finite elastostatics. *Archive for Rational Mechanics and Analysis* 1972; **44**(3):187–210.

51. Rice JR. Conserved integrals and energetic forces. In *Fundamentals of Deformation and Fracture* (*Eshelby Memorial Symposium*), Miller KJ, Bilby BA, Willis JR (eds). Cambridge University Press: Cambridge, MA, 1985; 33–56.
52. Eshelby JD. Energy relations and the energy-momentum tensor in continuum mechanics. In *Inelastic Behavior of Solids*, Kanninen MF *et al.* (eds). McGraw-Hill: New York, 1970; 77–115.
53. Kane C, Marsden JE, Ortiz M. Symplectic energy-momentum integrators. *Journal of Mathematical Physics* 1999; **40**:3353–3371.
54. Thoutireddy P, Ortiz M. A variational arbitrary Lagrangian Eulerian (VALE) method of mesh adaption and optimization. 2002, in preparation.
55. Abraham R, Marsden JE, Ratiu T. *Manifolds*, *Tensor Analysis*, *and Applications* (2nd edn). Springer: Berlin, 1988.