Preprint (2006)
http://www.geometry.caltech.edu

Geometric, Variational Integrators for Computer Animation

L. Kharevych Weiwei Y. Tong E. Kanso J. E. Marsden P. Schröder M. Desbrun
Caltech

Abstract

We present a general-purpose numerical scheme for time integration of Lagrangian dynamical systems—an important computational tool at the core of most physics-based animation techniques. Several features make this particular time integrator highly desirable for computer animation. It numerically preserves important invariants, such as linear and angular momenta. The symplectic nature of the integrator also guarantees a correct energy behavior, even when dissipation and external forces are added. Holonomic constraints can also be enforced quite simply. Finally, our simple design methodology allows for higher-order accurate schemes if needed. Two key properties set the method apart from earlier approaches. First, the nonlinear equations that must be solved during an update step are replaced by a minimization of a novel energy-like function, speeding up time stepping by more than a factor of two in practice. Second, the formulation introduces additional variables that provide key flexibility in the implementation of the method. These properties are achieved using a discrete form of a general variational principle called the Pontryagin-Hamilton principle, analog in spirit to geometric modeling techniques to design smooth curves or surfaces. We show its application to a simulation in non-linear elasticity with implementation details.

1. Introduction

Mathematical models of the evolution in time of dynamical systems (whether in biology, economics, or computer animation) generally involve systems of differential equations. *Solving* a physical system means figuring out how to move the system forward in time from a set of initial conditions, allowing the computation of, for instance, the trajectory of a ball (*i.e.*, its position as a function of time) thrown up in the air. Although this example can easily be solved analytically, direct solutions of the differential equations governing a system are generally hard or impossible—we need to resort to numerical techniques to find a discrete temporal description of a motion. Consequently, there has been a significant amount of research in applied mathematics on how to deal with some of the most useful systems of equations, leading to a plethora of numerical schemes with various properties, orders of accuracy, and levels of complexity of implementation [PFTV92]. In Computer Animation, these integrators are crucial computational tools at the core of most physics-based animation techniques, and classical methods (such as fourth-order Runge-Kutta, implicit Euler, and more recently the Newmark scheme) have been methods of choice in practice [Par01]. Surprisingly, developing better (*i.e.*, faster and/or more reliable) integrators received very little attention in our community, even if the few papers dedicated to this goal showed encouraging results [HES03].

In this paper, we follow a *geometric*—instead of a traditional numerical-analytic—approach to the problem of time integration. Geometry at its most abstract is the study of symmetries and their associated invariants. Variational approaches based on such notions are commonly used in geometric modeling and discrete differential geometry. Here we will treat mechanics in a similar way. Indeed, the very essence of a mechanical system is characterized by its *symmetries* and *invariants*. Thus preserving these symmetries and invariants (*e.g.*, certain momenta) into the discrete computational setting is of paramount importance if one wants discrete time integration to properly capture the underlying continuous motion. Motivated by the success of discrete variational approaches in geometry processing and the well-known variational nature of most dynamical systems, we advocate the use of *discrete variational principles as a way to derive simple, robust, and accurate time integrators*. In particular, we derive a novel, simple geometric integrator based on the very general Hamilton-Pontryagin principle.

1.1. Background

Dynamics as a Variational Problem Considering mechanics from a variational point of view goes back to Euler, Lagrange and Hamilton. The form of the variational principle most important for continuous mechanics is due to Hamilton, and is often called *Hamilton's principle* or the *least action principle* (as we will see later, this is a bit of a misnomer: "stationary action principle" would be more correct): it states that a dynamical system always finds an optimal course from one position to another. A more formal definition will be presented in Section 2, but one consequence is that we can recast the traditional way of thinking about an object accelerating in response to applied forces, into a geometric viewpoint. There the path followed by the object has *optimal geometric properties*—analog to the notion of geodesics on curved surfaces. This point of view is equivalent to Newton's laws in the context of classical mechanics, but is broad enough to encompass areas ranging to E&M and quantum mechanics.

Geometric Integrators are a class of numerical time-stepping methods that exploit the geometric structure of mechanical systems [HLW02]. Of particular interest within this class, *variational integrators* [MW01] discretize the variational formulation of mechanics we mentioned above, providing a solution for most ordinary and partial differential equations that arise in mechanics. While the idea of discretizing variational formulations of mechanics is standard for elliptic problems using Galerkin Finite Element methods for instance, only recently did it get used to derive variational time-stepping algorithms for mechanical systems. This approach allows the construction of integrators with any order of accuracy [Wes03, Lew03], and can handle constraints as well as external forcing. Results have been shown to be superior to all other types of integrators for simulations of physical phenomena [KMOW00]. This discrete-geometric framework is thus versatile, powerful, and general. For example, the well-known symplectic variant of the Newmark scheme (velocity Verlet) can best be elucidated by writing it as a variational integrator [Wes03]. Of particular interest in computer animation, the simplest variational integrator can be implemented by taking two consecutive positions $q_0 = q(t_0)$ and $q_1 = q(t_0 + dt)$ of the system to compute the next position q_2. Repeating this process calculates an entire discrete (in time) trajectory.

Accurate vs. Qualitative Integrators While it is unavoidable to make approximations in numerical algorithms (*i.e.*, to differ from the continuous equivalent), the matter becomes whether the numerics can provide satisfactory results. *Qualitative* reproduction of phenomena is often favored in computer animation over absolute *accuracy*. We argue in the following that one does not have to ask for *either* plausibility *or* accuracy. In fact, we seek a simple method robust enough to provide good, qualitative simulations that can *also* be easily rendered arbitrarily accurate. The simplectic character of variational integrators provides good foundations for the design of robust algorithms: this property guarantees *good statistical predictability* through accurate preservation of the *geometric* properties of the exact flow of the differential equations. As a consequence, symplecticity offers long-time energy preservation—a crucial property since large energy increase is often synonymous with numerical divergence while a large decrease dampens the motion, decreasing visual plausibility. A well-known example where this property is crucial is the simple pendulum (particularly relevant in robotic applications for articulated figures), for which even high-order integrators can fail in keeping the periodicity of the motion [Wik]. With this in mind, we will pursue numerical schemes which offer qualitatively-correct *as well as* arbitrarily accurate solutions.

1.2. Contributions

We address the problem of discrete time integration as a *discrete geometric problem* where the dynamics is obtained from a (stationary action) *Hamilton-Pontryagin principle*, *i.e.*, as the stationary point of a discrete action. Using the Hamilton-Pontryagin principle provides conceptual and algorithmic simplicity even for dissipative systems and in the presence of constraints. Computationally, our novel approach is more efficient (better than a factor of two improvement) since we can replace the usual non-linear multi-dimensional root finding time stepping techniques by a simpler *minimization* procedure (generalizing the idea of "minimum principle" [RO99]). The resulting new family of variational symplectic integrators also inherits key numerical properties: guaranteed momenta preservation and correct energy behavior. We demonstrate the robustness, simplicity, and efficiency of our time integration schemes by applying them to nonlinear elasticity and additionally describe a novel dissipation model.

2. Overview of Continuous Lagrangian Dynamics

Before presenting our contributions, we first give a description of the continuous Lagrangian principles of dynamical systems as they relate to the development of the discrete Hamilton-Pontryagin principle.

Consider a finite-dimensional dynamical system parameterized by the state variable q (*i.e.*, the vector containing all degrees of freedom). The Lagrangian function of the system is given as a function of q and $\dot{q}$. In the more restrictive case of basic elasticity, this Lagrangian function L is defined as the kinetic energy K minus the potential energy W of the system:

$$L(q,\dot{q}) = K(\dot{q}) - W(q).$$

The *action functional* is the integral of L along a path $q(t)$, over time $t \in [0,T]$. *Hamilton's principle* now states that *the correct path of motion of a dynamical system is such that its action has a stationary value*, *i.e.*, the integral along the correct path has the same value to within first-order infinitesimal perturbations. As an "integral principle" this description encompasses the entire motion of a system between two fixed times.

Computing variations of the action induced by variations δq of the path $q(t)$ results in:

$$\delta S(q) = \delta \int_0^T L(q(t),\dot{q}(t))\, dt = \int_0^T \left[\frac{\partial L}{\partial q}\cdot\delta q + \frac{\partial L}{\partial \dot{q}}\cdot\delta\dot{q}\right] dt$$
$$= \int_0^T \left[\frac{\partial L}{\partial q} - \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right)\right]\delta q\, dt \;+\; \left[\frac{\partial L}{\partial \dot{q}}\cdot\delta q\right]_0^T,$$

where integration by parts is used in the last equality. When the endpoints of $q(t)$ are held fixed with respect to all variations $\delta q(t)$ (*i.e.*, $\delta q(0) = \delta q(T) = 0$), the rightmost term in the above equation vanishes. Therefore, the condition of stationary action for arbitrary variations δq with fixed endpoints stated in Hamilton's principle directly indicates that the remaining integrand in the previous equation must be zero for all time t, yielding the well-known Euler-Lagrange equations:

$$\frac{\partial L}{\partial q} - \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) = 0. \tag{1}$$

Standard Example Let $K = \frac{1}{2}\dot{q}^T M \dot{q}$, where M is the mass Matrix. Then (1) simply states Newton's law: $M\ddot{q} = -\nabla W(q)$, *i.e.*, mass times acceleration equals force. Here, the force is conservative (no damping occurs) since it is derived from a potential function.

Forced Systems To account for non-conservative forces F (typically: dissipation), the least action principle is modified:

$$\delta \int_0^T L(q(t), \dot{q}(t))\, dt + \int_0^T F(q(t), \dot{q}(t)) \cdot \delta q\, dt = 0,$$

which is known as the *Lagrange-d'Alembert principle*.

Lagrangian vs. Hamiltonian Mechanics Lagrangian mechanics is not the only existing formalism available. In fact, Hamiltonian mechanics provides an alternative, closely related formulation. For later use we point out that Hamiltonian mechanics is described in *phase space*, *i.e.*, the current state of a dynamical system is given as a pair (q, p), where q is the state variable, while p is the momentum, defined as $p = \partial L/\partial \dot{q}$.

Discrete Lagrangian Mechanics The least action principle stated above can be used as a guiding principle to derive discrete integrators. In fact, West [Wes03] proposed a direct discretization of the *integral* of the Lagrangian to construct a proper and simple *discrete* action function. In this approach the integrals are replaced with quadrature rules, *i.e.*, linear combinations of discrete evaluations of the Lagrangian, over each elementary time step. Time stepping is then realized by taking the variation of the *discrete action* between two positions $q(t+dt)$ and $q(t+2dt)$ of a dynamical system. In this way this class of approaches respects the variational nature of time evolution in the discrete realm. The resulting *discrete Euler-Lagrange (DEL) equations* provide the update rule to advance in time: given two consecutive (in time) states of the system, the next state (at the end of the current time step) can be computed through a non-linear solve of the DEL equations.

3. Fully Variational Integrators

We will now present a novel family of variational integrators based on a more general principle known as the Hamilton-Pontryagin principle (a.k.a. Livins' principle). In this approach the velocity v is, a priori, an *additional* free variable. We will show how a discrete version of this principle will lead to integrators sharing the exact same numerical benefits as the best integrators known so far *and* allow us to express time-stepping as a simple *minimization* instead of a computationally more expensive multi-dimensional root finding problem.

3.1. Continuous Hamilton-Pontryagin Principle

The *Hamilton-Pontryagin principle* (deeply rooted in the control of dynamical systems) states that the equations of mechanics are given by the critical points of the *Hamilton-Pontryagin action*:

$$\delta \int_0^T \Big[p(\dot{q} - v) + L(q, v) \Big]\, dt = 0,$$

where the configuration variable q, the velocity v and the momentum p are all viewed as *independent* variables. (See [YM06] for an exposition and history.) That is, $q(t)$, $v(t)$, $p(t)$ are varied independently (with end-point conditions on $q(t)$). Notice the similarity with Hamilton's principle: p can be interpreted as a *Lagrange multiplier* to enforce the equality between $\dot{q}$ and v. The Hamilton-Pontryagin principle yields equations equivalent to the Euler-Lagrange equations (1), since, for the respective variations $\delta p(t)$, $\delta q(t)$ and $\delta v(t)$ over the three independent variables, we get:

$$v = \dot{q}, \qquad \frac{dp}{dt} = \frac{\partial L(q,v)}{\partial q}, \qquad p = \frac{\partial L(q,v)}{\partial v}. \tag{2}$$

We stress the important feature this different variational approach brings and that points to the generality of this principle: with the addition of the new variables, these equations can be understood from a Lagrangian *and* Hamiltonian point of view since the formulation involves both phase-space variables q and p within the action.

3.2. Set-Up and Discrete Formulation

Time Discretization A motion $q(t)$, for $t \in [0, T]$, of the mesh is replaced by a discrete sequence of poses q_k, with $k = 0, \ldots, N \in \mathbb{N}$, at discrete times: $\{t_0 = 0, \ldots, t_{k-1}, t_k, t_{k+1}, \ldots, t_N = T\}$. We will call h_k the time step between time t_k and t_{k+1}. Note that the time step can be adjusted throughout the computation based on standard time step control ideas if necessary. We similarly discretize $v(t)$ and $p(t)$ by the sets $\{v_k\}_{k=0}^N$ and $\{p_k\}_{k=0}^N$. Velocities v_k and momenta p_k are viewed as approximations *within* the interval $[t_{k-1}, t_k]$, *i.e.*, staggered with respect to the positions q_k.

Quadrature-based Discrete Action We will remain agnostic as to the Lagrangian used in this section: the case of non-linear elasticity will be addressed in Section 5, but our explanations are valid for any continuous Lagrangian $L(q, \dot{q})$. For a given choice of Lagrangian, one can easily derive a discrete action through *quadrature*. Computationally very attractive are one-point quadrature rules to turn the continuous action (*i.e.*, the integral in time of the Lagrangian) into a discrete Lagrangian $L^d(q_k, v_{k+1})$ through:

$$L^d(q_k, v_{k+1}) = L(q_k + \alpha v_{k+1}, v_{k+1})\, h_k \simeq \int_{t_k}^{t_{k+1}} L(q, \dot{q}) dt. \tag{3}$$

L^d is a time integral of the Lagrangian that we refer to as a discrete Lagrangian. This is not unlike the use of the term "discrete curvature" in CG which refers to a small, local *integral* of a continuous curvature. Notice that this quadrature has quadratic accuracy for $\alpha = 1/2$ and linear accuracy for all other $\alpha \in [0,1]$. More accurate quadrature rules (be they of Newton-Cotes or Gaussian type [PFTV92], for example) can be employed to increase the approximation order if necessary. Without loss of generality, we will solely use Eq. (3) in the remainder of this paper for simplicity.

3.3. Discrete Hamilton-Pontryagin Principle

Once a discrete Lagrangian is given, a ***discrete Hamilton-Pontryagin principle*** can be expressed through:

$$\delta \sum_{k=0}^{N} \left[p_{k+1}(\frac{q_{k+1}-q_k}{h_k} - v_{k+1})h_k + L^d(q_k, v_{k+1}) \right] = 0.$$

Discrete Variational Equations The discrete Hamilton-Pontryagin principle yields, upon taking discrete variations with respect to each state variable with fixed endpoints:

$$\delta p: \qquad q_{k+1} - q_k = h_k v_{k+1} \tag{4}$$

$$\delta q: \qquad p_{k+1} - p_k = D_1 L^d(q_k, v_{k+1}) \tag{5}$$

$$\delta v: \qquad h_k\, p_{k+1} = D_2 L^d(q_k, v_{k+1}) \tag{6}$$

where D_1 and D_2 denote the differentiation with respect to the first (q_k) and second (v_{k+1}) arguments of L^d.

Natural Update Procedure Given a point in the discrete Pontryagin-state space (q_k, v_k, p_k), the above equations are to be solved for $(q_{k+1}, v_{k+1}, p_{k+1})$ in the following way:

- *Plug* (5) *into* (6) so that p_{k+1} is replaced by a function of p_k and $D_1 L^d(q_k, v_{k+1})$.
- The resulting equation:

$$D_2 L^d(q_k, v_{k+1}) - h_k p_k - h_k D_1 L^d(q_k, v_{k+1}) = 0 \tag{7}$$

  can now be solved for v_{k+1} with any non-linear solver.
- q_{k+1} and p_{k+1} are found with (4) and (5) respectively.

Equivalence with DEL Equations One can readily verify (using the chain rule) that the integration procedure (4-6) obtained from the discrete Hamilton-Pontryagin principle is mathematically *equivalent* to the variational integrator described in [Wes03]. Thus, both schemes share the same numerical benefits such as the conservation of discrete energy and momenta, as we will discuss further in Section 4.3.

3.4. Discrete Pontryagin-d'Alembert Principle

For non-conservative systems, the (continuous) Pontryagin-d'Alembert principle is given by:

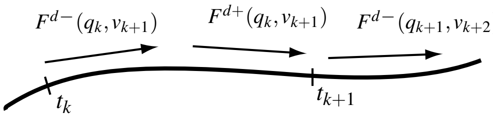
$$\delta \int_0^T [L(q,v) + p(\dot{q} - v)]\, dt + \int_0^T F_\nu(q,v) \cdot \delta q\, dt = 0$$

where $F(q,v)$ is an arbitrary (external) non-conservative force function. The ***discrete Pontryagin-d'Alembert principle*** can thus be defined as:

$$\delta(\sum_{k=0}^{N} p_{k+1}(q_{k+1} - q_k - h_k v_{k+1}) + L^d(q_k, v_{k+1})) +$$
$$\sum_{k=0}^{N} \left(F^{d-}(q_k, v_{k+1}) \cdot \delta q_k + F^{d+}(q_k, v_{k+1}) \cdot \delta q_{k+1} \right) = 0,$$

where F^{d-} and F^{d+} are defined analogously to F_d^- and F_d^+ in [Wes03] (see schematic figure below), *i.e.*, such that they approximate the total forcing over a time step:

$$F^{d-}(q_k, v_{k+1})\delta q_k + F^{d+}(q_k, v_{k+1})\delta q_{k+1} \simeq \int_{t_k}^{t_{k+1}} F(q, \dot{q})\delta q\, dt.$$



This yields, upon taking discrete variations, the following *forced discrete variational equations*:

$$q_{k+1} - q_k = h_k v_{k+1}$$
$$p_{k+1} - p_k = D_1 L^d(q_k, v_{k+1}) + F^{d-}(q_k, v_{k+1}) + F^{d+}(q_{k-1}, v_k)$$
$$h_k\, p_{k+1} = D_2 L^d(q_k, v_{k+1}).$$

3.5. Integration With Constraints

Our integration scheme can also accommodate holonomic constraints, *i.e.*, constraints that can be described by $g(q) = 0$. One just need to write the Hamilton-Pontryagin principle in terms of the variables q while using Lagrange multipliers λ to impose $g(q) = 0$:

$$\delta \int_0^T [L(q,v) + p(\dot{q} - v)]\, dt + \lambda g(q) = 0.$$

The discrete counterpart is then given by:

$$\sum_{k=0}^{N} p_{k+1}(q_{k+1} - q_k - h_k v_{k+1}) + L^d(q_k, v_{k+1}) + h_k\, \lambda_{k+1}\, g(q_{k+1}) = 0,$$

which yields the following *constrained discrete Hamilton-Pontryagin equations*:

$$q_{k+1} - q_k = h_k v_{k+1}$$
$$p_{k+1} - p_k = D_1 L^d(q_k, q_{k+1}) + h_k\, \lambda_k \nabla g(q_k)$$
$$h_k\, p_{k+1} = D_2 L^d(q_k, q_{k+1})$$
$$g(q_{k+1}) = 0.$$

These equations can be solved with any non-linear solver to derive new positions in time satisfying the holonomic constraints.

4. Faster Update through Minimization

The numerical properties of geometric integrators follow from the fact that the equations of motions on which they are based are found through the use of a discrete variational principle. Once the discrete update rules are established, a non-linear solver needs to be used in order to advance in time. In this section, we ask: can we turn this non-linear solution procedure for time update into a simpler and faster numerical procedure?

4.1. Discussion on Numerics

Current variational integrators resort to non-linear (root finding) solvers to find the next position so that one satisfies the

DEL equations (typically using an algorithm such as Newton's method [PFTV92]). Our novel integration scheme is, so far, no different: Eq. 7 needs to be solved similarly. Although seemingly related to an *energy minimization*, solving a set of non-linear equations can be far more delicate. The reason is quite simple: while the notion of "downhill" for a scalar field is easy and well defined, it does not translate directly to the case of multidimensional fields where there are conflicting downhill directions in each dimension. To circumvent this issue, solvers use the notion of "merit function" (the squared norm of the residual) to monitor the progress made towards reaching the zero [NW99]. Significant computational gain could thus be achieved by having a scalar function to minimize instead, with lower order and complexity than the merit function. In fact, this idea is very much responsible for the success of the well-known Conjugate Gradient method to solve a linear system like $Ax = b$. Its foundations come from a minimization technique applied to the function $f(x) = \frac{1}{2}x^t Ax - bx$. If one were to use the residual $\|Ax - b\|^2$ instead, the "merit function" has a term in $x^t(A^tA)x$, resulting in a much worse condition number. When non-linear equations are to be solved, the gain can be even greater. Thus, we proposes a more general derivation of variational integrators, and in particular, of our discrete Pontryagin-Hamilton integrator, for which the time stepping is performed through a minimization—hence our choice of the term *fully-variational integrators*: the variational principle is not only used for the derivation of the integrator, but also for numerical computations.

4.2. Variational Update

The time integrator that is based on (4-6) can be replaced by a variational update procedure done via *minimization of an energy-like function* given that the dynamical system satisfies certain integrability conditions as discussed below. This technique extends an idea of Radivitzky and Ortiz [RO99], where Verlet's integrator was shown to satisfy a *minimum principle*—a surprising fact given the extremum nature of Hamilton's principle. Our construction extends this property to a whole family of arbitrarily high order integration schemes.

We consider the class of dynamical systems whose discrete Lagrangian L^d has the property:

$$D_1 L^d(q_k, v_{k+1}) = \frac{\partial}{\partial v_{k+1}} P(q_k, v_{k+1}) \tag{8}$$

for some function $P(q_k, v_{k+1})$. The property (8) will be referred to as the ***variational integrability property***. We note here that this condition is strictly equivalent to another formulation given in Section 2.8 of [Lew03]. We will show however that this particular property is not as restrictive as indicated in this reference: in fact, most current models used in Computer Animation satisfy it.

Now, start again with the variational equations (4-6). Clearly, (6) can be rewritten as:

$$\frac{\partial}{\partial v_{k+1}}\left[-h_k p_{k+1} v_{k+1} + L^d(q_k, v_{k+1})\right] = -h_k p_{k+1} + \frac{\partial}{\partial v_{k+1}}\left[L^d(q_k, v_{k+1})\right] = 0$$

We can substitute (5) in the above equation to get:

$$-h_k p_k - h_k D_1 L^d(q_k, v_{k+1}) + \frac{\partial}{\partial v_{k+1}}\left[L^d(q_k, v_{k+1})\right] = 0$$

Thanks to the variational integrability property, this last equation can be rewritten as

$$\frac{\partial}{\partial v_{k+1}}\left[-h_k p_k v_{k+1} - h_k P(q_k, v_{k+1}) + L^d(q_k, v_{k+1})\right] = 0. \tag{9}$$

The quantity inside the bracket is an energy-like function of q_k, p_k and v_{k+1} and will be referred to hereafter as the ***Lilyan*** function $\mathcal{E}$:

$$\mathcal{E}(v_{k+1}) = -h_k p_k v_{k+1} - h_k P(q_k, v_{k+1}) + L^d(q_k, v_{k+1}). \tag{10}$$

The value of v_{k+1} can then be found as a *critical point* of the Lilyan. A closer look shows that if h_k is small, the Lilyan $\mathcal{E}$ is quadratic dominant (in v_{k+1}), since the terms depending on the potential energy are of order h_k^2, leaving only $p_k v_{k+1}$ and the kinetic energy as terms of order h_k. Thus, for small enough time steps, one can always find v_{k+1} as the value that *minimizes* the Lilyan for the current values of q_k and p_k.

We can now state the following result:

Suppose that the variational integrability property (8) *holds. Given the triplet* (q_k, p_k, v_k), *we can find* v_{k+1} *by minimizing the Lilyan defined by* (10), *while* q_{k+1} *and* p_{k+1} *are then explicitly computed using* (4) *and* (5). *The resulting triplet* $(q_{k+1}, p_{k+1}, v_{k+1})$ *satisfies* (4), (5), *and* (6), *giving us a fully variational integration scheme. In particular, this procedure defines a (symplectic) update map* $(q_k, p_k) \mapsto (q_{k+1}, p_{k+1})$.

Proof: Of course (4) and (5) are satisfied by construction. We need to check that (6) holds when minimizing (10) with respect to v_{k+1}. However, this is a simple calculation:

$$\begin{aligned} h_k p_{k+1} &= h_k p_k + h_k D_1 L^d(q_k, v_{k+1}) && \text{definition of } p_{k+1} \\ &= h_k p_k + h_k \frac{\partial}{\partial v_{k+1}} P(q_k, v_{k+1}) && \text{eq. (8)} \\ &= \frac{\partial}{\partial v_{k+1}}\left[h_k p_k v_{k+1} + h_k P(q_k, v_{k+1})\right] && \text{obvious} \\ &= \frac{\partial}{\partial v_{k+1}} L^d(q_k, v_{k+1}) && \text{assumed eq. (9)} \\ &= D_2 L^d(q_k, v_{k+1}), \end{aligned}$$

which is the desired equation (6). The last statement of our claim holds because this update map is equivalent to the position momentum form of the DEL equation mentioned in [Wes03]. ■

Generality of the Variational Integrability One can view the variational integrability property (8) as a design criterion that some (exceptionally nice) variational integrators might have. In fact, this property is valid for any quadrature-based discretization of a Lagrangian describing a basic elastic model (we will provide a concrete example of discrete Lagrangian for non-linear elasticity in Section 5). Thus, our construction is quite general for Computer Animation, and can easily be used to design higher-order accurate schemes (through higher order quadrature rules which map continuous integrals to discrete sums, see [MW01]) still satisfying this integrability criterion.

4.3. Numerical Advantages

Accuracy Our particular choice of one point-quadrature for the discrete Lagrangian renders the accuracy of integration linear (for $\alpha \neq 1/2$) or quadratic (for $\alpha = 1/2$). Although this level of accuracy is enough for most applications in graphics, one can devise higher-order schemes by providing more accurate quadratures, at the price of a higher computational cost. As we will detail in Section 5, the scheme we introduced is also quite versatile, as the value $\alpha = 0$ provides a fully explicit integration, which is very efficient, still linear accurate *and* continues to preserve momenta. Note also that the time step sizes h_k can be adjusted locally to control accuracy.

Conservation Laws A nice feature of our discrete variational framework is that the relationship between symmetry and conserved quantities matches the continuous theory of mechanics. More precisely, the invariance of the (continuous) Lagrangian under a given set of transformations of its variables defines its symmetries. Clearly these leave the action integral invariant as well. Thus symmetries give rise to *conserved quantities*, as stated in Noether's theorem. For example, the invariance of $L(q(t), \dot{q}(t))$ under translations and rotations results in the *conservation of linear and angular momenta*, respectively. One of the most attractive features of the variational integrators is that they conserve discrete quantities associated with discrete symmetries of the discrete Lagrangian [Lew03,Wes03]. We argued in Sections 3.2 and 4.2 that the variational scheme in (4-5) and (9) is mathematically equivalent to existing discrete Lagrangian-based integrators under certain integrability conditions and, hence, share the same numerical conservation properties (see Figure 1): momenta associated with symmetries of the Lagrangian are preserved exactly and automatically, for any order of accuracy. Note that the resulting update rules are not more complicated than standard integrators: we simply enforce this conservation laws at no extra cost by a proper discretization of the basic principle behind the underlying dynamics.

Energy The symplectic nature of our scheme also guarantees a good energy behavior. For conservative systems, the integration shows a nice energy preservation as demonstrated in Figure 1. The proper treatment of forced systems handles energy dissipation gracefully as well (see Figure 2). Note that the energy dissipation in more traditional integrators is often a mix of user-prescribed damping *and* uncontrollable numerical viscosity (depending on the time step size). In sharp contrast, our algorithm allows a precise control of the amount of damping introduced in the simulation independent of the time step used for simulation—a particularly desirable property to better control the behavior of physics-based models such as clothes where adaptive timestepping is often necessary.

A Word of Caution The reader may be misled into thinking that our scheme does not require the typical Courant-Friedrichs-Levy (CFL) condition (or equivalent) on the time step size. This is, of course, untrue: the same theoretical limitations in the explicit case ($\alpha = 0$) are still valid for our scheme. Other values of α—leading to implicit schemes—do not share this particular limitation, generally allowing for much larger time steps. Of course in the non-linear setting, time step sizes are often constrained by the non-linearity of the system. This too is no different in our setting with the notable exception that numerical energy minimizers (applied to the Lilyan) are notably less sensitive to this constraint then multi-dimensional root finders.

5. Application to Non-Linear Elasticity

In this section we put our theory to work by applying it to the simulation of the motion of an elastic body under the influence of external forces.

5.1. Set-Up

An elastic body $\mathcal{B}$ can undergo reversible deformations (changes in shape) due to applied forces. These may be *body forces* per unit volume or *surface traction* per unit area. Deformation typically depends on the material, size and geometry of the body as well as the applied forces. A motion is a one-parameter (time) family of deformations and can be described by $x(X,t)$, where X denotes the position of a material particle of $\mathcal{B}$ in the reference configuration and t is time. That is, x is the particle position in the deformed or *current* configuration.

The kinetic energy of the body is given by:

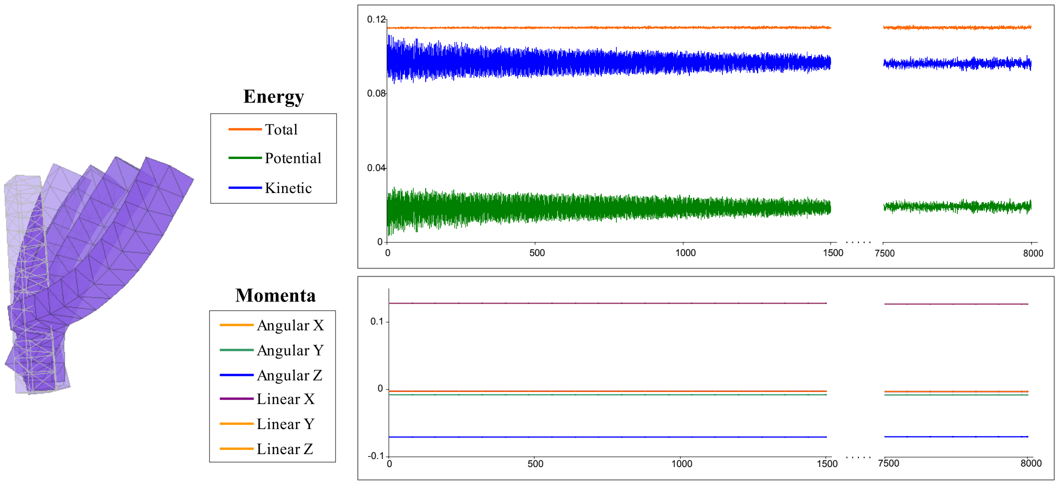
$$K = \frac{1}{2}\int_{\mathcal{B}} \rho\, v \cdot v \,\mathrm{d}V,$$

where ρ is the mass density, v is the velocity (function of material particle X and time t), and $\mathrm{d}V$ is a volume element. Further, in the pure mechanical theory of elasticity, there exists a *strain (or stored) energy density function* w per unit volume whose change represents the change in the internal energy due to mechanical deformations, which means the potential energy (excluding gravity) is written as

$$W = \int_{\mathcal{B}} w \,\mathrm{dV}.$$

The functional dependence of the internal energy w on the deformation is through the *Cauchy strain* C, defined as:

$$C = (\frac{\partial x}{\partial X})^T (\frac{\partial x}{\partial X}).$$



1: *Momenta and energy behavior for Explicit Integration over* 2 *million time steps: non-linear elasticity with explicit integration (see Section 5.3) is used to simulate an elastic rod (*160 *tets), given a non-zero initial position-momentum. No damping or external forces are used. Notice that the energy remains stable and the momenta are exactly preserved, even after* 8000 *seconds of simulation with a time step of* 0.004*s.*

More specifically, w can only depend on the three invariants I_1, I_2 and I_3 of the tensor C:

$$I_1 = \operatorname{tr}(C), \quad I_2 = \operatorname{tr}(C^2) - \operatorname{tr}^2(C), \quad I_3 = \det(C).$$

The function $w(I_1, I_2, I_3)$ varies depending on the material type, for instance, for Mooney-Rivlin materials, $w = a_1(I_1 - 3) + b_1(I_2 - 3)$, and for neo-Hookean materials $w = a_1(I_1 - 3)$. We will use the neo-Hookean model, but any other model results in a similar implementation.

Space Discretization We discretize space from the onset by approximating the elastic body using a finite dimensional simplicial mesh as routinely done in linear Finite Element methods, *i.e.*, using linear basis functions $\bar{N}$ associated to each vertex of the mesh. The position of the mesh vertices is described by the state variable q, and a motion of the mesh is represented by a time-dependent function $q(t)$. The spatially-discrete kinetic energy is formulated as

$$K_d = \frac{1}{2}\dot{q}^T M \dot{q},$$

where M is the lumped mass matrix, *i.e.*, M_{kk} is the mass inside the Voronoi cell of vertex k and $M_{kl} = 0$, while the discrete potential energy associated with the total stored energy (*i.e.*, excluding gravity) is denoted by W.

Time Discretization Using the same discrete setup as in Section 3.2, the time-discrete Lagrangian L^d can now be written as:

$$L^d(q_k, v_{k+1}, h_k) = h_k\left[\frac{1}{2}v_{k+1}^T M v_{k+1} - W(q_k + \alpha h_k v_{k+1})\right],$$

where we used a one point-quadrature (with $\alpha \in [0,1]$, midpoint for $\alpha = 1/2$) for the integration of W. Consequently, the partial derivatives are easily expressed as:

$$D_1 L^d(q_k, v_{k+1}, h_k) = -h_k \nabla W(q_k + \alpha h_k v_{k+1}),$$

$$D_2 L^d(q_k, v_{k+1}, h_k) = h_k\left[M v_{k+1} - \alpha h_k \nabla W(q_k + \alpha h_k v_{k+1})\right].$$

5.2. Damping

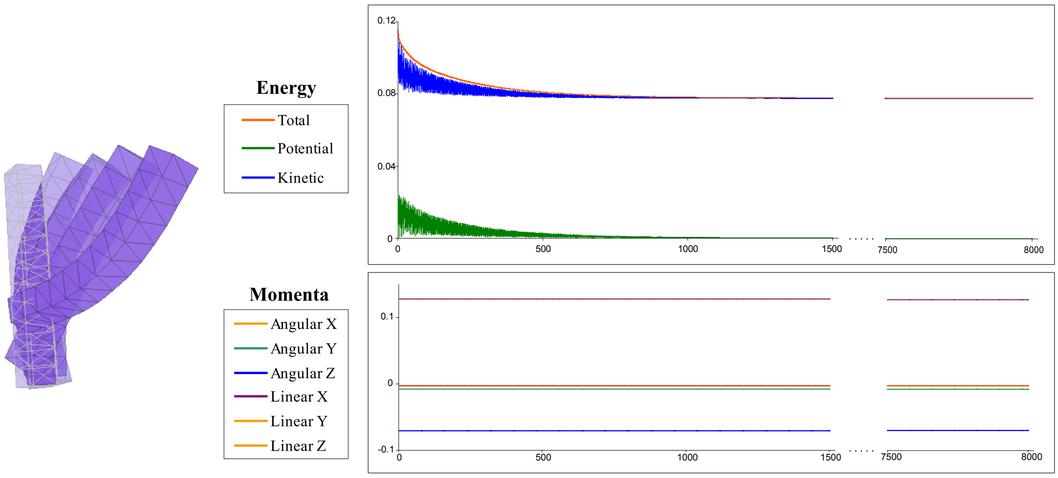
For damping, we propose an extension of the constraint-based damping model of Baraff and Witkin [BW98]. Our idea is to use the strain energy function to "measure" and damp the amount of deformation happening in one step, tantamount to a generalized Rayleigh damping. As discussed in the previous paragraph, the strain energy W is a function of the Cauchy tensor C, which itself is a function of the initial configuration $\bar{q}$ and the deformed configuration q: $W = W(C(\bar{q}, q))$. Thus we propose to simply compute the discrete damping force term as

$$F^d_{\text{damp}}(q_k, v_{k+1}) = -k_d \nabla W(C(q_k, q_k + h_k v_{k+1})).$$

Implementation of these damping forces is simple: for explicit integration, damping is added to F^{d+}, while for implicit integration, it is added to F^{d-} (improving the conditioning of the non-linear problem as it dampens the dynamics). Notice that the strain energy function depends on the gradient of the deformation field, so our damping model depends on the tensor ∇v . In particular, when the stored energy function of a spring is used, our model boils down to the traditional $-k_D k \dot{x}$ force. Similarly, for quadratic constraints, it becomes equivalent to the model proposed in [BW98]. Numerical experiments demonstrating the quality of this damping model (in particular, the fact that it does not reduce either linear or angular momentum) are described in Figure 2.

5.3. Numerical Integration

The particular choice of quadrature rule we have made thus far was designed purposely, for two distinct reasons. First, this quadrature allows fast time integration since finding the next position of a system only uses the state variables of the previous position. Second, despite its simplicity, the resulting scheme allows first and second order accuracy, the typical type of accuracy used in graphics. Finally, it also allows a choice for the user to go with a fast, explicit integration, or an implicit integration. We now describe the distinct integra-



2: *Damping is added to the same setup as in Fig. 1. The energy plot shows a smooth decrease over time, while momenta are* still *exactly preserved, even after* 2 *million time steps (explicit integration was used, with a constant time step of* $0.004s$*).*

tion schemes obtained depending on the value of α when an elastic object is simulated with external forces F_{ext} and using our damping model.

Explicit Time Integration The choice $\alpha = 0$ leads to a fully explicit, linear-accurate integration scheme: no minimization is needed. In particular, one can bootstrap the integration by setting $q_0 = \bar{q}$ (initial position), $p_0 = v_0 = 0$ (object at rest), then performing the following updates:

$$\begin{aligned} v_{k+1} &= M^{-1}\left[p_k - h_k \nabla W(q_k) + h_k F_{\text{ext}}(q_k)\right] \\ p_{k+1} &= M v_{k+1} - h_k k_d \nabla W(C(q_k, q_k + h_k v_{k+1})) \\ q_{k+1} &= q_k + h_k v_{k+1}. \end{aligned}$$

Notice that we handled the dissipating term in an explicit manner to keep the overall procedure fully explicit.

Implicit time integration For all other $\alpha \in (0,1]$, our integrator starts by finding v_{k+1} that minimizes the Lilyan $\mathcal{E}$:

$$\begin{aligned} \mathcal{E}(v_{k+1}) = &\frac{h_k}{2} v_{k+1}^T M v_{k+1} + h_k \frac{(1-\alpha)}{\alpha} W(q_k + \alpha h_k v_{k+1}) \\ &- h_k \frac{(1-\alpha)}{\alpha} E_{\text{ext}}(q_k + \alpha h_k v_{k+1}) \\ &+ h_k k_D W(C(q_k, q_k + h_k v_{k+1})) - h_k p_k v_{k+1}. \end{aligned}$$

where E_{ext} is the integral of the external force F_{ext} with respect to v_{k+1}. When non-integrable external forces are applied, the forced terms mentioned in Section 3.4 can be used instead. Other variables are then updated directly via the following rules:

$$\begin{aligned} p_{k+1} = &M v_{k+1} - h_k \alpha \nabla W(q_k + \alpha h_k v_{k+1}) \\ &+ h_k \alpha F_{\text{ext}}(q_k + \alpha h_k v_{k+1}), \\ q_{k+1} = &q_k + h_k v_{k+1}. \end{aligned}$$

Here, note that we included the dissipative terms directly inside the Lilyan function as it does not change the implicit nature of this choice of integrator. Note finally that this scheme is linear accurate, except for $\alpha = 1/2$ where the quadrature becomes quadratic accurate—thus, so is the scheme. This scheme was used to produce the animation of the bunny model in Figure 3 (see also the video).

5.4. Comparisons of Numerical Methods

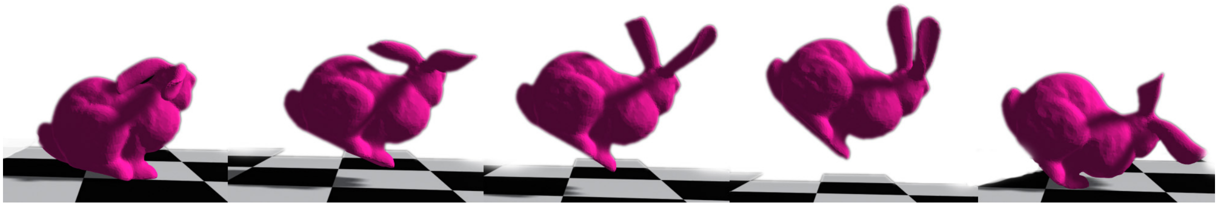
In order to assess the computational gain that our update via minimization confers, we ran the test presented in Fig. 1 with fixed timesteps at $h = 0.01s$, and at various spatial resolutions. We employed the widely-used TAO/PETSc solvers as neutral numerical tools instead of relying on our in-house solvers. All our tests were run on a 3GHZ XeonHT PC with 2.50GB of RAM. For a very low-res bar ($2K$ tets, 330 vertices), the speed-up of minimization vs. non-linear root finding is only 20%. However, as soon as the number of nodes increases, results show a clear superiority of the minimization procedure; for a bar with $12.5K$ tets (2000 vertices), the speed-up brought by our minimization update is already 2.6, while the same bar with now $24K$ tets (3784 vertices) yields a speed-up of 3.

We also experimented with larger simulations to test both the robustness and practicality of our family of integrators. We concluded that the correct energy behavior and momenta preservation with and without damping (demonstrated in Figs. 1 and 2) are indeed important qualities that most all other integrators (non-symplectic Newmark, implicit Euler, etc) do not have. In particular, being able to define damping such as the behavior of a physically-based object does not heavily depend on the time step size is a significant advantage when trying to design a particular animation: one can run well-predictive coarse simulations before running the final sequence.

6. Conclusions

We have presented an approach to derive general-purpose, fully variational time integrators for a wide class of mechanical systems using a discrete Hamilton-Pontryagin principle. Our approach has the following salient features:

- a minimization procedure replaces the traditional update rules which otherwise require computation-intensive multidimensional root-finding;



3: *Hopping Bunny: the fully-variational implicit time integration scheme presented in Section 5 is used to animate a bunny (*12.6*K vertices). Non-linear elasticity with a neo-hookean material model is used here.*

- the updates in time can be done explicitly or implicitly, and we demonstrated linear and quadratic accuracy;
- the time integrator is symplectic, and therefore preserves fundamental invariants while demonstrating excellent energy behavior;
- non-conservative forces or mechanical systems with (possibly non-linear) constraints can be handled easily and robustly;
- a novel damping model is easily added to our scheme;

The design of time integrators has not received much attention in our community despite their widespread use. Given the importance of qualitatively correct behavior in computer animation the geometric view is particularly pertinent as it ensures conservation of important quantities even for lower accuracy/higher speed simulations. Because of the general nature of our approach it no less admits high accuracy simulations when called for. An innovative aspect of our work is the introduction of the variational integrability condition which allows us to solve the non-linear problem at each time step (when using implicit integration) through a minimization. Together with the use of velocity/momentum/position variables it promises to play an important role in *motion control*. For instance, we believe that the optimization scheme proposed in [JMOB05] where the constraints are based on the discrete Hamilton variational principle could significantly benefit from our minimization-based integrators, as it can render the global optimization more scalable. Furthermore, we believe that the discrete Hamilton-Pontryagin principle that we introduce here and the ability to control v, q, and p should provide fertile grounds for various control tools (both optimal and trajectory control) as one can alter these quantities during integration to influence the motion accordingly. Finally, we wish to study whether model reduction [KLM00, BJ05] can benefit from the discrete variational integrator framework.

Acknowledgments. We thank Matt West, Hiroaki Yoshimura, and Michael Ortiz for helpful comments. This research was partially supported by NSF (ACI-0204932, DMS-0453145, CCF-0503786) and DOE (DE-FG02-04ER25657).

References

[BJ05] Jernej Barbič and Doug James. Real-Time Subspace Integration for St. Venant-Kirchhoff Deformable Models. *ACM Transactions on Graphics*, 24(3):982–990, August 2005.

[BW98] David Baraff and Andrew Witkin. Large steps in cloth simulation. In *Proceedings of ACM SIGGRAPH*, pages 43–54, 1998.

[HES03] Michael Hauth, Olaf Etzmuß, and Wolfgang Straßer. Analysis of Numerical Methods for the Simulation of Deformable Models. *The Visual Computer*, 19(7-8):581–600, 2003.

[HLW02] Ernst Hairer, Christian Lubich, and Gerhard Wanner. *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*. Springer, 2002.

[JMOB05] O. Junge, J.E. Marsden, and S. Ober-Blöbaum. Discrete mechanics and optimal control. In *Proceedings of the 16th IFAC World Congress*, pages Paper We–M14–TO/3, Prague, 2005.

[KLM00] P. Krysl, S. Lall, and J.E. Marsden. Dimensional Model Reduction in Non-linear Finite Element Dynamics of Solids and Structures. *Int. J. Numer. Meth. Engng*, 51:479–504, 2000.

[KMOW00] Couro Kane, Jerrold E. Marsden, Michael Ortiz, and Matt West. Variational Integrators and the Newmark Algorithm for Conservative and Dissipative Mechanical Systems. *Int. J. Numer. Methods Engrg.*, 49:1295–1325, 2000.

[Lew03] Adrian Lew. *Variational Time Integrators in Computational Solid Mechanics*. Phd thesis, California Institute of Technology, May 2003.

[MW01] Jerrold E. Marsden and Matt West. Discrete Mechanics and Variational Integrators. *Acta Numerica*, pages 357–515, 2001.

[NW99] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Series in Operations Research. Springer, 1999.

[Par01] Richard Parent. *Computer Animation: Algorithms and Techniques*. Series in Computer Graphics. Morgan Kaufmann, 2001.

[PFTV92] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 2nd edition, 1992.

[RO99] Raul Radovitzky and Michael Ortiz. Error Estimation and Adaptive Meshing in Strongly Nonlinear Dynamic Problems. *Comput. Method. Appl. M*, 172(1-4):203–240, 1999.

[Wes03] Matthew West. *Variational Integrators*. Phd thesis, California Institute of Technology, June 2003.

[Wik] Wikipedia. Geometric Integration Theory. http://en.wikipedia.org/wiki/Geometric_integration_theory.

[YM06] H. Yoshimura and J. E. Marsden. Dirac Structures and Lagrangian Mechanics. *J. Geom. and Physics (to appear)*, 2006.