

# Adaptively Boosted Time Steps Demonstrate Learning over Long Time Scales in Asynchronous Variational Integrators for a Nonconservative Model from Computational Mechanics

Timothy Tyree<sup>a,c,1</sup>

<sup>a</sup>Dept. of Physics; <sup>c</sup>University of California, San Diego

This manuscript was compiled on December 15, 2020

Variational integrators are numerical methods that discretize continuous optimization problems and have multi-disciplinary applications by solving initial value problems. Asynchronous variational integrators result from the assignment of local observation times to each finite element discretizing the body. compare to exact solution to step size field provided by (1) through a model system in uniform circular motion. By treating local observation times as dynamical variables, implicit time adaptation has been defined to explicitly conserve both energy and momentum. However, the resulting integrator is not always solvable, in particular near turning points, and reasonable attempts to approximate that integrator exhibit instabilities at long times. This motivates the investigation of explicit rules for finite elements to choose the size of their next time step and motivates the testing of such rules on discretized bodies of model systems such as an isotropic hyperelastic bodies responding to highly symmetric mechanical perturbation, as is done here.

The efficacy of explicitly controlling error tolerance is shown for these rules of time adaptation, and the improved computational efficiency is quantified directly. The algorithm is visualized that recieved a 340% slowdown by constraining to synchronous time steps.

Mean step size was found to be nondecreasing over long time scales for the trials that were considered in a systematic survey over the parameter space. These larger steps support the apparent improvement in computational efficiency while maintaining the explicitly given tolerance of error. Additionally, these increasing step sizes suggest the mesh is learning over long timescales.

Rapid learning of the ideal local step size emerged for trials with step sizes initialized to sufficiently large values, which suggests any learning over long time scales is a non-local emergence of order in a field of local step size. Rapid learning was confirmed analytically to occur in an exponentially small number of steps, which trials demonstrated to occur in the average finite element in less than twenty time steps, though typically in less than ten time steps. The runtime performance of the fourth order Dormand-Prince method suffered drastically when given a high max error tolerance. Lowering the max error tolerance produced runtimes comparable to that of the synchronous control simulation, which was much slower than the explicit Newmark method. A phenomenological model of variance of local step size emerged for numerous trials without damping ( $N_{\text{significant}} \geq 80$ ) that showed significant stationarity over long times ( $p < 0.01$ ). At short times, evidence of super-diffusion emerged in the variance of step sizes immediately following the synchronized initialization.

Overall, this report overwhelmingly supports the practical application of adaptive learning for local time step sizes used by the explicit Newmark method presented in this report. In this report, shape functions are discussed for neither space nor time.

Geometric Numerical Integration | Computational Mechanics | Structural Mechanics | Finite Element Methods | Machine Learning | Mathematical Physics

## Introduction

There exist no fixed time step numerical methods that conserves both energy and momentum (1, 2). Such conservation laws naturally define an implicit scheme for adaptive time step size (1). However, that implicit scheme is not always solvable and has been shown to be unsolvable near turning points (1). This has motivated methods for approximate solutions to these implicit equations involving energy reservoirs. However, those numerical methods were demonstrated to exhibit instabilities for long timescales (1).

The explicit Newmark method is a symplectic variational integrator can exhibit long time volume stability without the introduction of additional Lagrange multipliers in the theory (1), which is demonstrated in Fig. 4F.

**Configuration Manifolds and Deformation Maps.** In the numerical modeling of solid bodies moving in accordance to hyperelastic motion, it becomes useful to define a deformation map,  $\phi: \Omega \rightarrow \Omega'$  that smoothly transforms an unchanging reference configuration,  $\Omega$ , into a dynamical world configuration,  $\Omega'$ . The configuration can be any collection of degrees of freedom. In the event  $\Omega$  models a topological space such as smooth

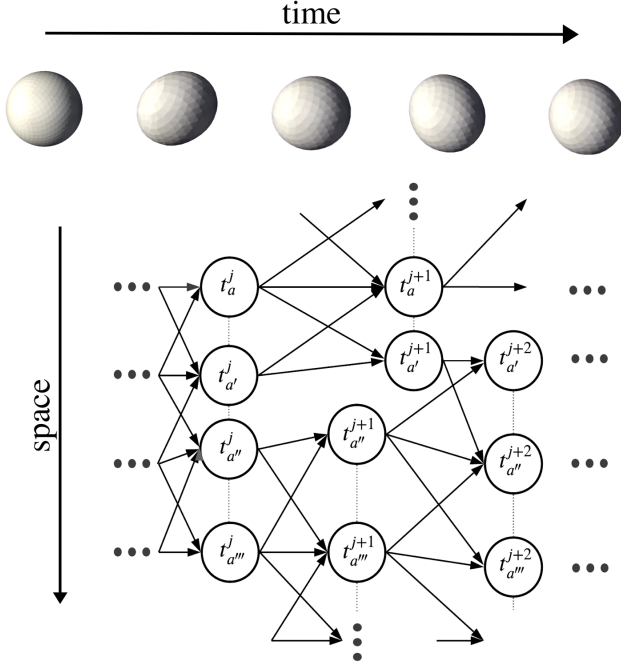
## Significance Statement

**Analogy to Machine Learning.** The error of discrete energy,  $\Delta E$  provides a savory minimization problem invoking recursive control of finite elements that goes beyond the simple modeling mechanical systems. One may consider the following loss function:

$$\mathcal{L} = \left( \alpha (\Delta E/E_0)^2 + (1 - \alpha) |\Delta E/E_0| \right)$$

where  $E_0$  has units of energy and  $\alpha \in [0, 1]$ . The vertices on a finite element can be analogous to analytically provided weights between perceptrons that may consider their recent time step history in choosing their next time step size. More discussion in text, in particular, consider "Adaptive Neural Time Steps" in "Future Directions".

<sup>1</sup>To whom correspondence should be addressed. E-mail = tyree at physics dot ucsd dot edu



**Fig. 1.** (top) four snapshots of the spherical hyperelastic body considered in this report and (bottom) a graph representation of finite elements with a local hidden variable,  $h$ , that represents the size of their next time step, which directly influences vertices, which can be viewed as "weights" between finite elements, which can be viewed as weak learners learning  $h$ . This recurrent network quickly learns a local step size that lies within the given range of given acceptable tolerances. Initializing with large step sizes results in effective adaptation in an exponentially small number of local time steps.

surface and an enclosed finite volume, one may linearize about a given  $X \in \Omega$  to find

$$x \equiv \vec{\phi}(X; t) = F(X; t) \cdot X + b(X; t), \quad [1]$$

defining the deformation gradient  $F : \Omega \rightarrow \Omega'$ , which accounts for local rotations, dilations, and general linear deformations. If the body is undeformed at  $x \in \Omega'$ , then  $F$  is an identity matrix. In this report, the reference configuration,  $\Omega$ , is taken to be the world configuration at equilibrium.

**Lagrange-d'Alembert Variational Principle.** To model systems with damping or forcing, we can attain the equations of motion from the Lagrange-d'Alembert variational principle in terms of a Lagrangian density, which we define on time and on the tangent bundle of the world configuration,  $T\Omega'$ . In other words,  $L : T\Omega' \otimes \mathbb{R} \rightarrow \mathbb{R}$ .

The physical trajectory of the deformation map satisfies

$$\delta S + \int_{t_0}^{t_f} dt \int_{\Omega} dX \vec{F}(\phi(X; t), \dot{\phi}(X; t), t) \cdot \delta \phi(X; t) = 0 \quad [2]$$

for all variations with  $\delta \phi(X, t_0) = \delta \phi(X, t_f) = 0$ ,  $\forall X \in \Omega$  and with the classical action,

$$S[\phi] = \int_{t_0}^{t_f} dt \int_{\Omega} dX L(\phi(X; t), \dot{\phi}(X; t), t) \quad [3]$$

Integrating a model for a physical systems requires that we consider the Lagrangian density,

$$L = \frac{1}{2} \dot{\phi}^T \rho(X; t) \dot{\phi} - U(\phi(X; t), t),$$

where  $\rho$  is the mass density, and  $U(x, t)$  is the potential energy density.

**Neo-Hookean Constitutive Relation.** To avoid numerical artifacts arising from the inversion of finite volume elements, a Neo-Hookean constitutive relation is used that was used previously (3). The stress-strain energy density is

$$\Psi(F) = \mu(F^T F - \log(\det(F)) - 3) + \frac{\lambda}{2} \log^2(\det(F)) \quad [4]$$

where  $\mu > 0$  and  $\lambda > 0$  are the first and second Lamé parameters, measuring resistance to stretching/shearing and volume change, respectively, and may be computed using Eqn. 13.

It has been shown that Eqn. 4 results in the following first Piola-Kirchoff stress tensor:

$$P(F) \equiv \frac{\partial \Psi(F(X))}{\partial X} = \mu(F - F^{-T}) + \lambda \log(\det(F)) F^{-T}. \quad [5]$$

where we have used some identities\*. Additionally, the variation is

$$\begin{aligned} \delta P(F; \delta F) &= \mu \delta F + (\mu - \lambda \log(\det(F))) F^{-T} \delta F^T F^{-T} \\ &\quad + \lambda \text{tr}(F^{-1} \delta F) F^{-1}. \end{aligned} \quad [6]$$

**Nonlinear Hyperelastic Forces.** Hyperelastic forces result from variations of the the hyperelastic strain energy density  $\Psi = \Psi(F; t)$ . Following Ref. (3), for a sufficiently small neighborhood  $N_\epsilon(X^*) \subset \Omega$ , one can make the approximation

$$\int_{N_\epsilon(X)} dX \Psi(F; t) \approx W \cdot \Psi(F(X^*; t)),$$

where  $W = |N_\epsilon(X^*)| > 0$  is the volume of the neighborhood. An immediate corollary is that the force at the location  $X^* \in \Omega$  is given by

$$\begin{aligned} \vec{F}_{\text{elastic}}(X^*; t) &= -\nabla(\Psi(F(X; t))) W \\ &= -W \frac{\partial \Psi}{\partial F} \cdot \frac{\partial F}{\partial X^*} \equiv -W P \cdot \partial_{X^*} F. \end{aligned} \quad [7]$$

We consider the variation,

$$\delta \vec{F}_{\text{elastic}}(X^*, t) \equiv \frac{\partial}{\partial X} \cdot \delta \phi(X; t)|_{X=X^*}.$$

**Linear Rayleigh Drag.** As shown previously (3), the Rayleigh drag force in a hyperelastic body is

$$\vec{F}_{\text{damping}}(X^*, v^*; t) = \gamma \frac{\partial}{\partial X} \vec{F}_{\text{elastic}}(X; t) \Big|_{X=X^*} \cdot v^* \quad [8]$$

Where  $v^* \equiv \frac{d\phi(X^*; t)}{dt}$  is the velocity of  $\Omega'$  at time  $t$  and at location  $x^* \equiv \phi(X^*; t)$ .  $v^*$  is a member of an item taken from the tangent bundle configuration,  $T\Omega'$ .

\*We have the identities from Ref. (3):  $\delta[F^{-1}] = -F \delta F F^{-1}$ ,  $\delta[F^{-T}] = -F \delta F^T F^{-T}$ , and  $\delta[\det F] = \text{tr}(F^{-1} \delta F) \det F$ .

**Discretization of Space.** By discretizing Eqn. 2, we obtain equations defining a numerical method for advancing in time in terms of a force density,  $F$ . Let  $q_a = (x_a, v_a) \in T\Omega$  denote a nodal position and velocity for  $a = 1, 2, 3, \dots, N_{\text{vertices}} \gg 1$  nodes.

Consider  $\Sigma_K \subset \Omega$  to be the  $K^{\text{th}}$  finite volume contained in the configuration manifold for  $K = 1, 2, 3, \dots, N_{\text{elements}}$ , such that the collection of all such finite elements cover  $\Omega$ . Supposing  $N_{\text{elements}} \gg 1$  is sufficiently large, we may consider a piece-wise linear approximation to  $\phi$ , which supports approximating  $F$  with the constant value,  $F(X; t) = F(X^*; t)$  for all  $X \in \Sigma_K$ . Consider  $K^*$  to be the element index that contains the location  $X^*$ . Suppose that  $\Omega$  and  $\Omega'$  are both embedded in three spatial dimensions. Suppose  $(a_1^K, a_2^K, a_3^K, a_4^K)$  are the four node indices that correspond to the four nodes closest to  $X^*$  that are not all coplanar. Let  $A$  be a linear map from the  $K$  element indices to the  $a$  node indices. One may return these node positions in the deformed configuration as  $(x_{a_1^K}, x_{a_2^K}, x_{a_3^K}, x_{a_4^K})$ . Similarly, there exist static node positions in the material configuration,  $(X_{a_1^K}, X_{a_2^K}, X_{a_3^K}, X_{a_4^K})$ . Substituting Eqn.1 and rewriting,  $x_{a_i^K}$  as  $x_i$  for ease of notation,

$$\begin{cases} x_1 - x_4 = F(X_1 - X_4) \\ x_2 - x_4 = F(X_2 - X_4) \\ x_3 - x_4 = F(X_3 - X_4) \end{cases}$$

which is rewritten as

$$D_s \equiv \begin{bmatrix} x_1 - x_4 \\ x_2 - x_4 \\ x_3 - x_4 \end{bmatrix}^T = F \begin{bmatrix} X_1 - X_4 \\ X_2 - X_4 \\ X_3 - X_4 \end{bmatrix}^T \equiv F D_m. \quad [9]$$

Supposing that the vertices of this tetrahedron are not coplanar, there exists a  $D_m^{-1}$ , and one may immediately compute  $F = D_s D_m^{-1}$  and the finite volume  $W = |\det(D_s)|/6$ . For an unchanging equilibrium configuration, one may precompute  $B_m \equiv D_m^{-1}$  for each finite element.

**Discretization of Time.** In considering the variational problem stated by Eqn. 2, we consider classical action,  $S : T\Omega' \rightarrow \mathbb{R}$ .

We may then compute the classical discrete action,  $S_d$  and interchange the order of the sum and the integral over time

$$\begin{aligned} S_d &= \int_{t_0}^{t_f} dt \sum_{a=1}^{N_{\text{vertices}}} \frac{1}{2} \dot{x}_a(t)^T m_a \dot{x}_a(t) \\ &\quad - \int_{t_0}^{t_f} dt \sum_{K=1}^{N_{\text{elements}}} W_K(t) \Psi(F(X_K^*, x_K^*(t))) \\ &= \sum_{a=1}^{N_{\text{vertices}}} \int_{t_0}^{t_f} dt \frac{1}{2} \dot{x}_a(t)^T m_a \dot{x}_a(t) \\ &\quad - \sum_{K=1}^{N_{\text{elements}}} \int_{t_0}^{t_f} dt W_K(t) \Psi(D_s^K(t) B_m^K) \end{aligned}$$

where  $m_a > 0$  is the mass assigned to node  $a$ . Letting  $\star(a)$  denote the set of elements that have  $a$  at one of their four vertices, barycentric masses are chosen to discretize the masses of nodes according to  $m_a = \frac{1}{4} \sum_{K \in \star(a)} \rho(X_K^*) W_K$ , which has the desirable properties such as being time invariant and summing to the same value as the sum of the masses of the tetrahedral finite volume elements.

For  $N_K - 1$  time steps, let  $\Theta_K = \{t_K^j \in \mathbb{R}^+ | t_0 \leq t_K^0 < t_K^1 < t_K^2 < \dots < t_K^{N_K-1} < t_K^{N_K} \leq t_f\}$  denote the discretization of the time interval  $[t_0, t_f]$  for the  $K^{\text{th}}$  element. Similarly for  $N_a - 1$  time steps, let  $\Theta_a = \{t_a^j \in \mathbb{R}^+ | t_0 \leq t_a^0 < t_a^1 < t_a^2 < t_a^3 < \dots < t_a^{N_a-1} < t_a^{N_a} \leq t_f\}$ . The discrete action may now be written in the computable form,

$$\begin{aligned} S_d &= \sum_{a=1}^{N_{\text{vertices}}} \sum_{j=0}^{N_{\text{steps}}^a} (t_a^{j+1} - t_a^j) \frac{1}{2} \dot{x}_a(t_a^*)^T m_a \dot{x}_a(t_a^*) \\ &\quad - \sum_{K=1}^{N_{\text{elements}}} \sum_{j=0}^{N_{\text{steps}}^K} (t_K^{j+1} - t_K^j) W_K(t_K^*) \Psi(D_s^K(t_K^*) B_m^K) \end{aligned}$$

A choice of locality has been made in discretizing time. By choosing  $t_a^* = t_K^* = t_K^{j+1}$ , Eqn. 2 exhibits a minimizing function consistent with the discrete equations of motion implemented in Alg. 1.

**Endless Possibilities for Time Steps.** There are endless possibilities for how to choose  $\Theta_K$  and  $\Theta_a$ . For numerical methods that are self-adjoint, negative time steps are allowed, and the strict *time ordering* constraint may be relaxed. Otherwise, one can use a priority queue to iterate through the time-ordered updates for elements, such that the earliest element time is always performed first. Node times, positions, and velocities are updated with the element updating its time and any other local variables hidden from passive, hyperelastic dynamics.

Treating the contents of  $\Theta_a$  and  $\Theta_K$  as dynamical variables and taking variations in them has lead to discretized equations of motion that explicitly conserve both energy in addition to momentum (4).

**Discrete Lagrange d'Alembert Principle.** Let  $D_a^i$  denote the derivative operation with respect to  $x_a^i$ , which represents node  $a$  at time  $t^i$  (1). The discrete Lagrange d'Alembert Principle that is valid for asynchronous time discretizations results from Eqn. 2 to be

$$D_a^i S_d + \sum_{K, a \in K} \vec{F}_{K,a}^{j,i} = 0 \quad [10]$$

which defines an asynchronous variational integrator that converges on the exact solution (5) in terms of the discrete force,  $\vec{F}_{K,a}^{j,i}$ , exerted by element  $K$  on node  $a$  at time  $t^i \in [t_K^j, t_K^{j+1}]$ . In this report, a discrete force has units of impulse.

**Net Forces Elements Exert on Their Nodes.** Let  $\vec{F}_e^{a_i}(t)$  be the hyperelastic force of element  $K$  acting on one of its four nodes,  $a_i \in (a_1, a_2, a_3, a_4)$  from time  $t_K^j$  to time  $t$ , which is approximated as a constant value in that interval in this report. Arranging into a row vector as in Eqn. 9, Eqn. 7 provides  $\vec{F}_{\text{elastic}}^{a_i} \equiv \vec{F}_e^{a_i}$  for  $i = 1, 2, 3$ . Conservation of momentum is enforced by  $\vec{F}_e^{a_4} = -\vec{F}_e^{a_1} - \vec{F}_e^{a_2} - \vec{F}_e^{a_3}$ .

The damping force is computed similarly (Eqns. 8, 6, 9) to compute  $\vec{F}_{\text{elastic}}^{a_i}$  for each node. The net force is then

$$\vec{F}_{\text{net}}^{a_i} = \vec{F}_{\text{elastic}}^{a_i} + \vec{F}_{\text{damping}}^{a_i}. \quad [11]$$

## Methods

**Repeatable Perturbations of the Unit Sphere.** The surface of the unit sphere embedded in three spatial dimensions was

generated and discretized as a triangular mesh that is available for download (6). The spatial discretization of this spherical body resulted from a constrained Delaunay tetrahedralization method that has certain theoretical guarantees of correctness (7). The initial velocity of all nodes were set to zero in the second and third coordinates of an orthonormal basis of  $\mathbb{R}^3$ . For the first coordinate,  $\hat{x}$ , the initial velocity of all nodes was perturbed in the inward direction by initializing the array of node velocities to

$$\vec{v}_a(t_a^0) = -v_{\text{scale}} \left( \vec{x}_a(t_a^0) \cdot \hat{x} \right) \hat{x} \quad \forall a, \quad [12]$$

where  $v_{\text{scale}}$  is a scalar parameter with units of rate.

**Explicit OneStep Method following Newmark.** This explicit method following a previous work by Newmark (8) defines a variational integrator (1). An elemental update done according to this method consists of first updating the node locations to the final time, computing the nodal forces, and then updating the node velocities to the final time using Newton's second law. We consider  $x_K, v_K, m_K$ , and  $\tau_K$  to be the collection of most recent node positions, velocities, masses, and times, respectively. Updating the  $K^{\text{th}}$  element from time  $t_K^j$  to  $t_K^{j+1}$  can be performed by an explicit Newmark method using the following algorithm:

---

**Algorithm 1** Integrate one step forward in time.

---

**Input:**  $t_K^{j+1}, t_K^j, x_K, v_K, \tau_K, B_K, m_K$   
**Output:**  $x_K, v_K, \tau_K$   
**procedure** ONESTEP  
 $x_a \leftarrow x_a + (t_K^{j+1} - t_a)v_a \quad \forall a \text{ on } K$   
 $F_a \leftarrow \text{compute net force on node } a \quad \forall a \text{ on } K$   
 $v_a \leftarrow v_a + (t_K^{j+1} - t_K^j)\vec{F}_a/m_a \quad \forall a \text{ on } K$   
 $t_a \leftarrow t_K^{j+1} \quad \forall a \text{ on } K$

---

Let  $t_a \in \tau_K$ . Let  $x_a \in x_K$ . Net force is computed according to Eqn. 11.

**Iterative Solution to the Implicit Midpoint Rule.** The Implicit Midpoint Rule defines the next configuration as the one self-consistent with forces computed from the average of the current configuration and the next configuration. The Neo-Hookean constitutive model considered in Eqn. 5 provides nonlinear nodal forces (Eqn. 7). Instead of looking for explicit solutions to nonlinear problems, an approximate solution was computed iteratively. The initial approximate of the next configuration,  $q_{(0)}^{j+1}$ , was computed by Alg. 1. The next approximate,  $q_{(k+1)}^{j+1}$  was computed using forces evaluated at  $q_{(k+1)}^* \equiv \frac{1}{2}(q^j + q_{(k)}^{j+1})$ . The result of the Implicit Midpoint Rule is consistent with the result of  $\lim_{k \rightarrow \infty} q_{(k)}^{j+1}$ . After ten iterations, the approximates appeared to converge on the apparent solution. For all results involving the Implicit Midpoint Rule, thirty iterations were used, at which point negligible changes were observed on the order  $|q_{(k+1)}^{j+1} - q_{(k)}^j| \sim 10^{-10}$ .

**Explicit OneStep Method following Dormand and Prince.** This explicit method follows the work of Dormand and Prince (9). The implementation used in this report used a Butcher tableau described previously (10). The corresponding OneStep method uses six force computations per time step, is fourth order accurate, and provides a local error estimate with no

additional force computations because of its second estimate that is fifth order accurate.

**Method of Asynchronous Integration.** An algorithm for forward asynchronous integration was initialized with nodal times that are not synchronized, which was used in recording all observations in this report. Care was taken to synchronize node positions before taking geometric measurements like volume. At a high level, this integration is performed by maintaining a time ordering queue,  $\mathcal{Q}$ , of task. Each task is equipped with the future time of the completion of that task,  $t_K^{j+1}$ . Each task, concerns a particular element,  $K$ , that is to update from the time of its most recently completed task, from  $t_K^j$  to  $t_K^{j+1}$ . After the completion of this task, the next task, denoted  $(t_K^{j+2}, K)$ , is pushed onto  $\mathcal{Q}$ . Let the function,  $\text{Pop}(\mathcal{Q})$  return the remaining task with the earliest time.

Let the final time  $t_f$  come after the latest of initial element times,  $\Theta_{\text{initial}} \equiv \{t_K^0 \mid K = 1, 2, \dots, N_{\text{elements}}\}$ , which uniquely determines the initial node times. If runtime is not a concern,  $\mathcal{Q}$  may be a simple sorted list. For reasonably fast runtimes,  $\mathcal{Q}$  is best implemented as a datatype that supports fast retrieval of the earliest time. Since  $\mathcal{Q}$  is a priority queue datatype, it was implemented using a heap data structure.

---

**Algorithm 2** Integrate the body forward to time  $t_f$ .

---

**Input:**  $t_f, x_a, m_a, x_a, v_a \forall a; B_K, m_K, h_K, t_K^0 \forall K;$   
**Output:**  $x_K, v_K, \tau_K, \forall K/;$   
**procedure** FORWARDINTEGRATE  
 $t_K^1 \leftarrow t_K^0 + h_K^1, \forall K.$   
Push  $(t_K^1)$  to  $\mathcal{Q}$  for each element,  $K$ .  
**while**  $\mathcal{Q}$  is not empty **do**  
 $(t_K^{j+1}, K) \leftarrow \text{Pop}(\mathcal{Q})$   
**if**  $t_K^{j+1} \leq t_f$  **then**  
 $x_K, v_K \leftarrow \text{OneStep}(t_K^{j+1}, t_K^j, x_K, v_K, \tau_K, B_K, m_K)$   
 $t_K^{j+2} \leftarrow$   
 $h_K \leftarrow \text{ComputeNextStep}(\dots)$   
 $t_K^{j+2} \leftarrow t_K^{j+1} + h_K$   
Push  $(t_K^{j+2}, K)$  to  $\mathcal{Q}$

---

One concern can be had for whether a sufficiently small initial step size  $h_K^1$  was chosen for all  $K$ . There are many OneStep methods that suppose small time steps. Relief may be sought from the Courant condition for hyperelastic bodies, which suggests the initial step size should be limited by  $0 < h_K^1 \leq \frac{1}{10} \frac{r}{c}$ , where  $r$  is the radius of the largest sphere that can fit inside the finite element, and where

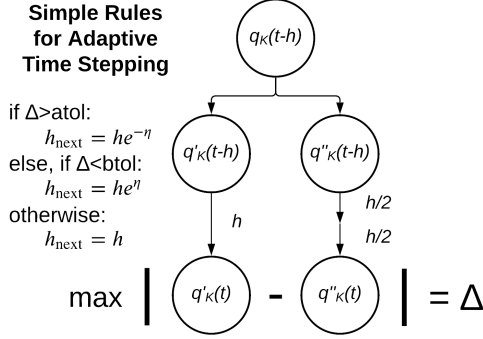
$$c = \sqrt{\frac{\mu_0 + 2\lambda_0}{\rho_0}}$$

is an estimate of the wave speed of mechanics vibrations in a hyperelastic body (1) and where  $\rho_0$  is a reasonable mass density,  $\mu_0$  is a reasonable value for Lamé's first parameter, and  $\lambda_0$  is a reasonable value for Lamé's first parameter.

**Explicit Rules for Adaptive Time Steps.** Fig. 2 shows the rules for choosing the next time step for a given element for  $\text{ComputeNextStep}$  in Alg. 2. This method for adaptive step sizes triples the number of necessary force computations to estimate the max absolute error,  $\Delta$ , between the OneStep method and the exact flow map, and is valid for any OneStep method that



is at least second order accurate for a sufficiently small step size. Let  $\text{atol}_{x,v}$  equal the maximum absolute tolerance of error allowed by the simulation for the position and velocity, respectively. Let  $\text{btol}_{x,v}$  equal the minimum absolute tolerance of error allowed by the simulation for the position and velocity, respectively. If  $\Delta > \text{atol}_{x,v}$ , then the step size is decreased by a multiplicative factor of  $e^{-\eta}$ . If  $\Delta < \text{btol}_{x,v}$ , then the step size is increased by a multiplicative factor of  $e^{+\eta}$ . These rules naturally define an explicit, local method for computing the next time step size for `ComputeNextStep` in Alg. 2 that guides adaptation of the finite element to stay within a given range of absolute tolerances of error.



**Fig. 2.** method for adaptive time stepping. (left) The rules for choosing the next time step for a given element and (right) a diagram of these rules.

As shown in Fig. 2, the input element configuration is copied into two configurations,  $q'_K(t-h)$  and  $q''_K(t-h)$ . Some OneStep method is used to compute  $q'_K(t)$ , and that same OneStep method is used to compute,  $q''_K(t-h/2)$  and then used again to compute  $q''_K(t)$ . The magnitude of the error from the exact flow map is then estimated by the measure  $\Delta = \max |q'_K(t) - q''_K(t)|$  for any OneStep method that is at least second order accurate.

**Systematic Survey.** The survey systematically varied over the parameter choices listed in Tab. 1,2. The majority of these trials had the material parameters  $\mu = 1, \lambda = 10, \gamma = 0.05$ , and  $\rho = 1$ . Throughout the systematic survey of parameters, numerous cases were considered where  $\text{atol}_{x,v} = \text{btol}_{x,v}$ , which produced results that appeared comparable to trials where  $\text{atol}_{x,v}$  and  $\text{btol}_{x,v}$  differed by at least one order of magnitude.

## Results

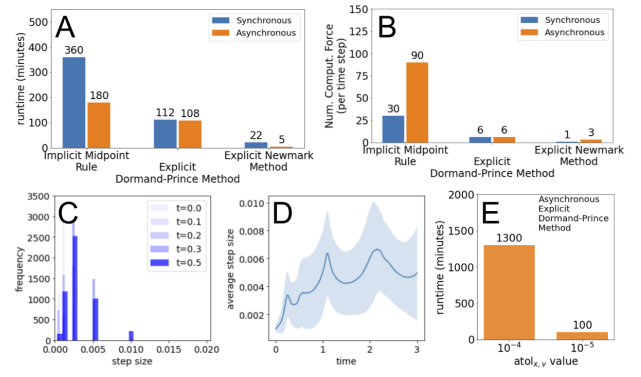
**Demonstrated Efficacy of Time Stepping.** The efficacy of the adaptive time steps resulting from Alg. 1 was demonstrated by visually comparing a trial with asynchronous time steps to a similar trial with synchronous time steps and observing visually comparable deformations over long times (see "doc/mov/" in Ref. (6)), and efficacy was quantified by comparing global measures of the simulated model versus time for various pairs of asynchronous trials and comparable synchronous trials.

**Computational Benefit of Asynchronous Integration.** The explicit procedure shown in Fig. 2 was evaluated, and the runtimes were significantly reduced for the resulting asynchronous trial when compared to the corresponding synchronous trial,

which has  $\text{ComputeNextStep}(h_K^{j+1}) = h_K^{j+2}$ . In Fig. 3A, the explicit Newmark method incurs a 340% slowdown when not using Alg. 1.

The average runtime per unit time  $t$  for the synchronous explicit Newmark method was 7 minutes, while the runtime per unit  $t$  was only 2 minutes for the asynchronous explicit Newmark method. These runtimes were computed with 2,177 finite elements. This significant improvement in run time occurred despite the additional number of force computations required by the method in Fig. 2.

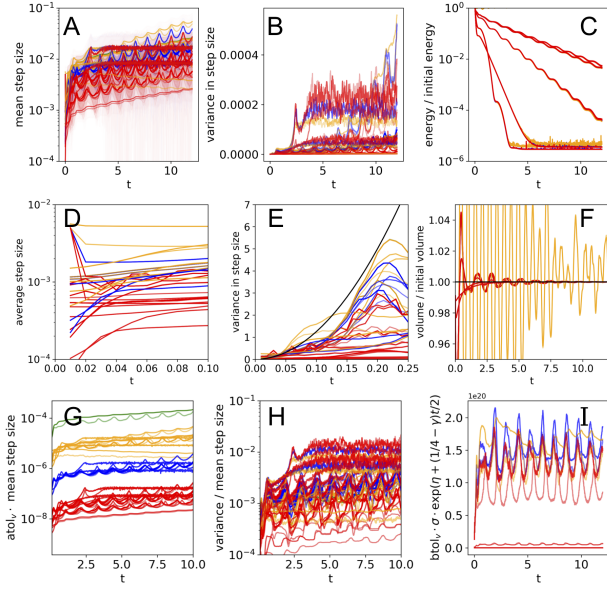
In contrast to the explicit Newmark method in Alg. 1, which uses one force computation per time step, another integrator was implemented using the Implicit Midpoint Rule, which is supported by the apparent improvement in run times in Fig. 3A. Another integrator was implemented using the explicit Dormand-Prince method, and after calibrating the parameters to its error estimate, it did not demonstrate a significant speedup comparable to that of the explicit Newmark method described in Alg. 1.



**Fig. 3.** results quantifying a significant decrease in runtime that may be attributed to the adaptive time stepping routine in Fig. 2. (A) The runtimes of the six numerical methods were compared with and without the adaptive time stepping. Significant speedups were observed for (A,left) the Implicit Midpoint Rule and for (A,right) the explicit Newmark method (Alg. 1). Time step adaptation did not appear to benefit performance for the fourth order explicit Dormand-Prince method, which considered a max absolute error that was fifth order accurate. (B) The minimum number of force computations per call the OneStep method. The Implicit Midpoint Rule considered thirty iterations, which was too slow to be of much practical use. (C) The frequency of discrete step sizes both grows and disperses shortly after a synchronous initialization at a small step size. This temporal binning may be helpful in parallel implementations. More discussion in text, in particular, consider "Parallel Adaptive Time Steps" in "Future Directions". (D) Mean step size versus time with shaded region identifying the 1 standard deviation range from the mean. Both step size plots had  $v_{\text{scale}} = 2$  for the initial conditions in Eqn. 12 with material parameters  $\mu = 1, \lambda = 10, \gamma = 0.05$ . (E) The runtime performance of the fourth order Dormand-Prince OneStep method suffered drastically at the aforementioned parameter values when  $\text{atol}_{x,v}$  was set to  $10^{-4}$ . Lowering  $\text{atol}_{x,v}$  to  $10^{-5}$  produced performance comparable to the control simulation that used synchronous time steps. Overall, the explicit Newmark method appears strictly superior to these other two numerical methods.

**Higher order method exhibits faster run times at lower tolerances.** A OneStep method equipped with a local error estimate other than the one presented here (Fig. 2) suffered unreasonably long runtimes when the magnitudes of  $\text{atol}_{x,v}$  and  $\text{btol}_{x,v}$  were chosen incorrectly (Fig. 3E). This could have occurred because if both the error took a typical value that was orders of magnitude smaller than allowed  $\text{btol}_{x,v}$ , which might have led to step sizes growing to sizes that exhibited instabilities.

**Rapid Learning of Step Size for Large Initial Step Size.** From these rules for a finite element to select its next time step size, certain patterns of the largest stable local step size emerged within two times steps. These step sizes were found to desynchronize over time in predictable ways from synchronous time steps to asynchronous time steps. In Fig. 4, the time traces are colored by the parameter  $\text{atol}_v$ , as it appeared to be the limiting parameter when compared to  $\text{atol}_x$ , which is the max absolute difference between a single time step and two half time steps, as shown in Fig. 2.



**Fig. 4.** Time traces of global measures for a systematic survey of different parameter settings ( $N_{\text{trials}} = 159$ ). Time traces are colored by the parameter,  $\text{atol}_v$ . The smallest was  $\text{atol}_v = 1\text{e-}5$  (red), then was  $\text{atol}_v = 1\text{e-}4$  (blue), then was  $\text{atol}_v = 1\text{e-}3$  (orange), and  $\text{atol}_v = 1\text{e-}2$  (green). The step sizes are from the method in Fig. 2. (A) Mean time step size versus  $t$  demonstrated rapid adaptation when using large initial step sizes. Time traces covering a one standard deviation ( $1\sigma$ ) neighborhood of the mean are visibly plotted but are faint. (B) Variance of step size versus  $t$ . Variance increases as elements desynchronize. Patterns visibly emerged for bodies not experiencing Rayleigh damping. (C) Ratio of total energy to initial energy versus  $t$ . Rayleigh damping dissipates energy at an exponential rate. (D) Mean time step size versus  $t$ . Trials initialized to larger step sizes of size  $h \sim 10^{-2}$  adapted to stable step sizes of size  $h \sim 10^{-3}$  within twenty time steps. In contrast, trials initialized to small, stable step sizes tend to adapt more slowly to reach comparable step sizes within sixty time steps. (E) Variance of step size versus  $t$ . The variance increased at an increasing rate at early times, indicating fast desynchronization, but was bounded above by (black) the function  $5^3 t^2$ . (F) Ratio of total volume to initial volume versus  $t$ . For long times, volume was stable and fluctuations in total volume were relatively small. (G) Mean step size scaled by the max absolute tolerance in velocity error,  $\text{atol}_v$ , versus  $t$ . (H) The ratio of variance to mean of step size versus time exhibited significant stationarity ( $p < 0.01$ ) in numerous trials ( $N_{\text{significant}} \geq 80$ ). (I) The transformed variance measure of a step size versus time was computed according to  $\text{btol}_v \cdot \sigma \exp(\eta + (1/4 - \gamma)t)$ . The numerical method used Algs. 1 and Alg. 2 with ComputeNextStep described by Fig. 2.

**Emergent Properties of Step Size Variance.** Recalling that one property indicative of Poisson random variables is that their ratio of variance to mean assumes a constant value on average. For the method described by Algs. 1, 2 and Fig. 2, the ratio of variance to mean size of time step exhibited a stationary time evolution (Fig. 4H) that was significant ( $p < 0.01$ ) over long time scales ( $t > 2.5$ ), for numerous trials that were effectively undamped ( $N_{\text{significant}} = 80$ ). Stationarity was confirmed by

comparable significance tests conducted on the logarithm of this ratio, which confirmed significant stationarity in the time evolution ( $p < 0.01$ ) of numerous trials ( $N_{\text{significant}} = 85$ ) that were effectively undamped. These tests for stationarity were conducted according to an augmented Dickey-Fuller unit-root test for stationary processes.

Also at long times  $t > 2.5$ , the transformed variance measure plotted in Fig. 4I also exhibited a stationary time evolution that was significant ( $p < 0.01$ ) for numerous trials ( $N_{\text{significant}} = 96$ ) trials. Comparable significance tests were carried out on the logarithm of this transformed variance measure, which was found to be significantly stationary ( $p < 0.01$ ) for numerous trials ( $N_{\text{significant}} = 121$ ) that were effectively undamped. The transformed variance model plotted in Fig. 4I is

$$b \equiv \text{btol}_v \cdot \sigma e^{\eta + (1/4 - \gamma)t/2}.$$

Additionally, a property indicative of super-diffusion emerged at short time scales (Fig. 4E). The apparent variance of step sizes appeared to scale with the square of  $t$  for  $t < 0.2$  and was bounded above by  $5^3 t^2$ .

## Discussion

**Adaptively Boosted Time Steps.** Adaptively boosted time steps can adapt in an exponentially small number of number of time steps. The key principle of adaptively boosting time stepping in mechanical systems is straightforward. Suppose a series of  $n$  successive time steps each exhibit  $\Delta > \text{atol}_{x,v}$ . For a learning rate of  $\eta = \log(2)$ , the rules in Fig. 2 dictate that the time step size decreases by a factor of  $e^{-\eta} = 1/2$  for each such time step. For  $n$  such time steps, the initial time step would have decreased by a factor of  $e^{-n\eta} = 2^{-n}$ , which is exponentially small in the number of such time steps.

Suppose a finite element has the sequence of step sizes,  $h_j$  for  $j = 1, 2, \dots$  and the first  $n \geq 0$  time steps exhibit  $\Delta > \text{atol}_{x,v}$ . The  $n^{\text{th}}$  step size will be

$$h_n = h_{n-1} e^{-\eta} = h_{n-2} e^{-2\eta} = \dots = h_1 e^{-n\eta}$$

which is exponential in the number of such time steps,  $n$ . This rapid adaptation may happen if the initial step size takes a large value, as is the case for time traces in Fig. 4D.

**A Trend Emerges.** An emergent trend appears to be universal in an overwhelmingly large subset of the  $N_{\text{trials}} = 159$  trials considered in the systematic survey. This trend is plotted in Fig. 4I. As stated in Results, this trend implies that at long times after synchronization is relaxed  $t > 2.5$ , these adaptively boosted time steps exhibit a variance satisfying

$$\sigma^2 \propto \frac{1}{\text{btol}_v^2} e^{2\eta} e^{(\gamma - \frac{1}{4})t}.$$

It is reasonable for  $1/\text{btol}_v^2$  to factor into the long term variance of step sizes provided that velocity scales of error are proportional to displacement scales of error divided by some time that is typical for these step sizes.

**Endless Possibilities for Error Estimators.** The particular loss measure considered in this report was the max absolute difference between a single OneStep call with step size  $h$  and two OneStep calls each of step size  $h/2$ .

Other measures of error could potentially be better suited to serve as  $\Delta$  in Fig. 2. One can consider many convex

functional forms that might be reasonable choices for  $\Delta > 0$ . A more general loss measure,  $\mathcal{L}$  is stated in the Significance Statement, which depends on some estimate measuring the error of energy. There may be other well characterized values that are worth consideration in some applications.

Measures that were tested with Algs. 2, 1 but were not shown to be effective or efficient included the change of local energy between time steps. Another such measure that was tested and was not shown to be effective or efficient was the local energy difference between a pair of OneStep methods—one which took one time step of size  $h$  and another which took two time steps each of size  $h/2$ . In this letter, local energy was considered to be the discrete energy of only one finite element and not its neighbors. Including neighbors via some averaging procedure could potentially result in an estimator for  $\Delta E$  that proves both efficient and effective.

A more general method of estimating error is to compare the results of two OneStep methods where one method is known to have higher precision than the other, or by comparing to a known conservation law.

**Parameter Changes that can Improve Adaptively Boosted Time Steps.** Decreasing the learning rate,  $\eta \geq 0$  could improve the stability of the simulation. A smaller learning rate results in more bins of constant step size near those initial step sizes. If the initial step size is placed in a trajectory that is an attractor to similar step sizes, then the simulation will likely become entrained in that learning trajectory, and the simulation will learn to increase the mean step size over long time scales.

A smaller learning rate results in more bins of constant step size near the initial step sizes. If the initial step size is placed near a mean step size trajectory that is an attractor, then the finite elements could exhibit larger stepsizes, and the simulation would be learning to run more efficiently.

Setting the maximum absolute tolerance at extremely small values, such as  $\text{atol}_{x,v} \leq 10^{-8}$ , excessively slowed computation. For the Dormand-Prince OneStep method, a value of  $10^{-6} \leq \text{atol}_{x,v} \leq 10^{-4}$  allowed some integrations up to time  $t = 3$  to run in less than two hours, as shown in Fig. 3A,E. The Dormand-Prince was not demonstrated to have any advantage over the Newmark method presented in Alg. 1.

## Concluding Remarks

Asynchronous variational integrators result from assigning local observation times to each finite element and updating their configurations using a variational OneStep method. This report investigates simple rules for explicit time adaptation of finite elements. These rules are tested on a model system of a spherical isotropic hyperelastic material responding to a highly symmetric mechanical perturbation, and Rayleigh damping is introduced. The efficacy of these rules for controlling tolerance demonstrated, and the improved computational efficiency is quantified directly. Rapid learning is also demonstrated when step sizes are initialized to large values.

The runtime of a fourth order method suffered drastically at a maximum error tolerance of  $10^{-4}$ , but was relatively unaffected at a maximum error tolerance of  $10^{-5}$ . A phenomenological model of step size variance emerged for trials that were not experiencing significant Rayleigh damping. This model for step size variance was significantly stationary ( $p < 0.01$ )

for numerous trials ( $N_{\text{significant}} \geq 80$ ) over long time scales ( $t > 2.5$ ).

For early times ( $t < 0.2$ ), a super-diffusive process was observed in the time series of the apparent variance of step sizes.

Overall, this report overwhelmingly supports the practical application of adaptively boosted time steps for local time steps evaluated by the explicit Newmark method described in Alg. 1. This method provides significantly faster runtimes and supports numerical accuracy comparable to existing numerical methods currently in common use for computational structural mechanics.

The adaptive time steps described in Methods can be readily be adapted to existing finite element code by using a previously implemented OneStep method as an alternative to Alg. 1.

Additionally, this report introduces several interesting extensions that are worth future study. Some such extensions are discussed in the following:

**Future Work.** Future work may include the implementation of simple electrophysiological models of cardiac arrhythmia (11) to be followed by the implementation of simple electromechanical coupling (12). Future work may also include the implementation of a marching tetrahedra algorithm (13) to detect isosurfaces both of constant transmembrane voltage and of the rate of change of transmembrane voltage. A modified marching tetrahedra algorithm has been made to detect the intersections of such isosurfaces as one dimensional strings. Such one dimensional strings are known to exist at the center of scroll waves, which have been visualized in pig ventricles exhibiting ventricular fibrillation using ultrasound (14).

The aforementioned future work describes a numerical method that provides a quantitative interface for the study of sudden cardiac death in terms of one dimensional strings.

Future work may include a comparison of the emergent step size field to the solution that achieves implicit adaptive time stepping by explicitly conserving both energy and momentum (1). This should be conducted on a model system that exhibits no turning points, such as solid pendulum exhibiting uniform circular motion.

**Future Directions.** A number of potential extensions of this report are discussed in the following, and any effort spent on them or advise on them would be appreciated.

One such extension includes adaptive tolerance. The results in this report considered static absolute tolerances. Relative tolerance is worth investigation, as well.

The nodal forces used in this report modeled a hyperelastic solid material that is homogeneous and isotropic and experiencing Rayleigh damping. It is known that myocardial tissue exhibits anisotropy through a mechanism involving the orientation of muscle fibers, which has been considered previously (11). Anisotropic material is worth consideration in a future effort.

**Endless Possibilities for Parallel Implementations of Adaptively Boosted Time Steps.** Another extension of this report could involve accelerating this method with a graphics processing unit, which could be implemented using any of several open source libraries, such as numba.cuda, pycuda, or compute.scala (15–17). Another such extension is to incorporate this numerical

method in an open source graphics design software (18), which already supports the use of graphics processing units (19).

Implementing a fully parallelized asynchronous variational integrator is nontrivial for OneStep methods that are not self-adjoint. For OneStep methods that are not self-adjoint, the order of the priority queue still must be observed. This can be done reasonably well using a parallel priority queue data type implemented using a heap bucket data structure (20).

A rigorous heap data structure may not be necessary, however, given a reasonable choice of a constant learning rate,  $\eta$ . The adaptively boosted time steps described in this report results in time steps constrained to convenient bins (see Fig. 3D). A parallel algorithm could take advantage of this if the step sizes nodes are initialized to values constrained to these bins. The choice of a large, constant learning rate resulted in a conveniently small number of bins that may update synchronously. Given the impressive state-of-the-art for synchronous parallel processors, it is worthwhile to consider some parallel priority queue that works in terms of these bins and allows for contiguous memory access.

**Adaptive Neural Time Steps.** Newton’s method of gradient descent could be applied to predict time step sizes that minimize the change to local energy to form an algorithm classified as a recurrent neural network. Such approaches should produce results consistent with the integrator resulting from the equations of explicit conservation of both energy and momentum (1, 4). However, the explicit Newmark method with adaptively boosted time steps that this report describes has demonstrated both efficacy and efficiency, and it is comparatively simple to implement.

## Acknowledgements

This report was written in completion of the curriculum for “Geometric Numerical Integration” offered under the instruction of Prof. Melvin Leok at the University of California, San Diego. The numerical methods presented in this report were implemented in NumPy (21), which was compiled to LLVM machine code just in time using Numba (15). Image rendering was performed using pyvista and tk (22), and videos were compiled and edited using ffmpeg (23). I am making the notebooks that generated these results open source for your convenience (6).

## Works Referenced

1. A Lew, J Marsden, M Ortiz, M West, Variational time integrators. *Int. J. for Num. Methods Eng.* **60**, 153–212 (2004).
2. C Kane, JE Marsden, M Ortiz, Symplectic-energy-momentum preserving variational integrators. *J. Math. Phys.* **40**, 3353–3371 (1999).
3. E Sifakis, J Barbic, Fem simulation of 3d deformable solids: a practitioner’s guide to theory, discretization and model reduction in *Acm siggraph 2012 courses*. pp. 1–50 (2012).
4. AJ Lew, Ph.D. thesis (California Institute of Technology) (2003).
5. S Müller, M Ortiz, On the  $\gamma$ -convergence of discrete dynamics and variational integrators. *J. Nonlinear Sci.* **14**, 279–296 (2004).
6. T Tyree, avi (<https://github.com/timtyree/avi>) (2020).
7. H Si, Tetgen, a delaunay-based quality tetrahedral mesh generator. *ACM Trans. Math. Softw.* **41** (2015).
8. NM Newmark, A method of computation for structural dynamics. *J. Eng. Mech. Div.* **85**, 67–94 (1959).
9. JR Dormand, PJ Prince, A family of embedded runge-kutta formulae. *J. Comp. Appl. Math.* **6**, 19–26 (1980).
10. LF Shampine, Some practical runge-kutta formulas. *Math. Comput.* **46**, 135–150 (1986).
11. F Fenton, A Karma, Vortex dynamics in three-dimensional continuous myocardium with fiber rotation: Filament instability and fibrillation. *Chaos: An Interdiscip. J. Nonlinear Sci.* **8**, 20–47 (1998).

12. CM Augustin, et al., The impact of wall thickness and curvature on wall stress in patient-specific electromechanical models of the left atrium. *Biomech. Model. Mechanobiol.*, 1–20 (2019).
13. A Doi, A Koide, An efficient method of triangulating equi-valued surfaces by using tetrahedral cells. *IEICE TRANSACTIONS on Inf. Syst.* **74**, 214–224 (1991).
14. J Christoph, et al., Electromechanical vortex filaments during cardiac fibrillation. *Nature* **555**, 667–672 (2018).
15. SK Lam, A Pitrou, S Seibert, Numba: A llvm-based python jit compiler in *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*. pp. 1–6 (2015).
16. A Klöckner, et al., Pycuda and pyopencl: A scripting-based approach to gpu run-time code generation. **38** (2012).
17. M Odersky, et al., The scala language specification (2004).
18. BO Community, *Blender - a 3D modelling and rendering package* (Blender Foundation, Stichting Blender Foundation, Amsterdam), (2018).
19. A Brito, Blender 2.8 for architecture: modeling and rendering with eevee and cycles. (2019).
20. J Iacono, B Karsin, N Sitchinava, A parallel priority queue with fast updates for gpu architectures. *arXiv preprint arXiv:1908.09378* (2019).
21. CR Harris, et al., Array programming with NumPy. *Nature* **585**, 357–362 (2020).
22. CB Sullivan, A Kaszynski, PyVista: 3d plotting and mesh analysis through a streamlined interface for the visualization toolkit (VTK). *J. Open Source Softw.* **4**, 1450 (2019).
23. F Developers, ffmpeg tool (2016).

## Supplementary Information

**Tables of Parameters.** Tab. 1 shows parameters specific to the simulation itself and Tab. 2 shows parameters specific to the material properties of the body. There is the following conversion to Lamé’s parameters from Young’s modulus,  $k$ , and Poisson’s ratio,  $\nu$ :

$$\mu = \frac{k}{2(1 + \nu)}, \quad \lambda = \frac{k\nu}{(1 + \nu)(1 - 2\nu)}. \quad [13]$$

*Nota bene:* all parameters listed in Tab. 1, 2 are non-negative real numbers computed with double floating point arithmetic.

Parameter	Name	Value(s) Considered
$\text{atol}_x$	max absolute tolerance of position error	(1e-5 – 1e-3)
$\text{atol}_v$	max absolute tolerance of velocity error	(1e-5 – 1e-3)
$\text{btol}_x$	min absolute tolerance of position error	(1e-7 – 1e-5)
$\text{btol}_v$	min absolute tolerance of velocity error	(1e-7 – 1e-5)
$h_{\text{init}}$	initial step size for all elements	(1e-3 – 1e-1)
$\eta$	learning rate for all elements	log(0.1 – 64)
$N_{\text{elements}}$	number of elements	(2e+3 – 2e+4)
$N_{\text{vertices}}$	number of elements	(5e+2 – 4e+4)
$v_{\text{scale}}$	initial velocity rate scale	(1. – 3.)

Table 1. parameters specific to the simulation

Parameter	Name	Value(s) Considered
$\rho_0$	mass density	1.
$\mu$	Lamé’s first parameter	(1. – 10.)
$\lambda$	Lamé’s second parameter	(1. – 10.)
$\gamma$	damping rate	(0. — 1.)

Table 2. parameters specific to the material properties of a hyperelastic body experiencing Rayleigh drag.