

Asynchronous Contact Mechanics

David Harmon
Columbia University

Etienne Vouga
Columbia University

Breannan Smith
Columbia University

Rasmus Tamstorf
Walt Disney Animation Studios

Eitan Grinspun
Columbia University

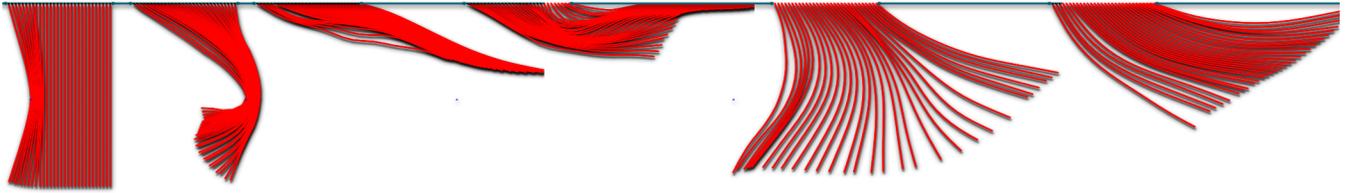


Figure 1. A prescribed particle slowly moves through a set of curtains, then impulsively shifts to a very high velocity. The slow and fast phases highlight the method’s ability to handle smooth resting and sliding with deep stacking, and arbitrarily fast penetration-free movements in which collisions are treated when (as opposed to well before or after) they occur. The curtains continue to swing for a long time, even as controlled internal dissipation damps high frequencies.

Abstract

We develop a method for reliable simulation of elastica in complex contact scenarios. Our focus is on firmly establishing three parameter-independent guarantees: that simulations of well-posed problems (a) have no interpenetrations, (b) obey causality, momentum- and energy-conservation laws, and (c) complete in finite time. We achieve these guarantees through a novel synthesis of asynchronous variational integrators, kinetic data structures, and a discretization of the contact barrier potential by an infinite sum of nested quadratic potentials. In a series of two- and three-dimensional examples, we illustrate that this method more easily handles challenging problems involving complex contact geometries, sharp features, and sliding during extremely tight contact.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

Keywords: variational, symplectic, contact, collision, simulation

1 Motivation

Even as computer hardware benefits from Moore’s Law, our ability to program, debug, and maintain software advances at a humbler pace. This observation shapes our priorities as we develop physical simulation tools for computer graphics. While making choices that yield up-front simplicity and blazing performance is important today, we prefer that these choices do not obstruct our long-term goals of extending functionality and improving realism. Laying aside ad-hoc models in favor of physical approaches might require a deeper initial investment, but it promises to pay off handsomely in predictability, controllability, and extensibility. From this vantage point, we propose to revisit the long-studied problem of simulating deformable objects in complex contact scenarios.

Safety, correctness, progress Robust simulation of complex contact scenarios is critical to applications spanning graphics (training, virtual worlds, entertainment) and engineering (product design, safety analysis, experimental validation). Challenging scenarios involve dynamics with frequent and distributed points of contact, interaction with sharp boundaries, resting and sliding contact, and combinations thereof. Useful resolution of these scenarios requires consideration of the fundamental issues of geometric *safety*, physical *correctness*, and computational *progress*, with the respective meanings that (a) for well-posed problems the simulation does not enter an invalid (interpenetrating) state, (b) collision response obeys physical laws of causality and conservation (of mass, momentum, energy, *etc.*), and (c) the algorithm completes a simulation in finite, preferably short, time.

An ideal algorithm offers provable guarantees of safety, correctness, and progress. A safety guarantee eliminates the need to iterate through the animation-design process because of unsightly penetration artifacts; such a guarantee should not fall on an overburdened user lapped in tunable parameters. A correctness guarantee is a prerequisite for physical behavior that is consistent under rediscretization of space and time. Respect for causality is critical to capturing chain reactions and phenomena such as waves and stacking; discrete conservation laws allow for the development of tunable dissipation that does not “cross-talk” with parasitic numerical damping. If, however, these two guarantees are not accompanied by guaranteed progress, the simulation may never complete, no matter how fast or parallel the hardware.

Shortcomings of synchrony Most time integration methods are synchronous, moving the entire configuration forward in lock-step from one instant in time to the next. Such synchrony is fundamentally at odds with safety, correctness, and progress: the first two goals are assured by attending to collisions in order of causality, which can require arbitrarily small time steps. The number of possible impact events in a single “reasonable” time step can be enormous: in their analysis of contact, Cirak and West [2005] present a counting argument and conclude that synchronous “contact simulation algorithms cannot attempt to exactly compute the sequence and timing of all impacts,” as this would preclude reasonable progress.

The graphics community’s prevailing emphasis on *progress* has motivated many efforts to find, *retroactively*, a physically plausible collision response to a set of collisions that occurred over a preceding time interval [Provot 1997; Bridson et al. 2002]. Such methods typically have adjustable parameters that must be carefully chosen

to balance safety and progress; other methods discard causality in favor of progress [Milenkovic and Schmidl 2001].

The principled, faithful simulation of complex collisions for deformable objects remains an open, challenging, and important problem.

Asynchrony We propose to place safety and correctness on an equal footing with progress. To overcome the fundamental opposition between these requirements, we turn to *asynchronous integration*, which integrates each geometric element at its own pace, *not* in lockstep with the entire object. Asynchrony offers compelling long-term advantages for simulations of deformable objects in complex contact—advantages that remain unexplored, in particular in terms of safety, correctness, and progress.

For scenarios involving sharp boundaries or dispersed points of contact, asynchrony renders non-interpenetration and momentum conservation tractable. Because elements advance at their own pace, those not entangled in collisions can proceed at large time steps. As shown in Figure 2, the median time step of an asynchronous method can be moderate even when tight collisions force some elements to proceed at small time steps.

Asynchronous integration As a point of departure we consider *asynchronous variational integrators* (AVIs) [Lew et al. 2003], which belong to a larger class of integrators that exactly conserve both momentum and symplecticity (loosely related to areas in phase space). The well-known Verlet (“leapfrog”) integrator is symplectic; such integrators are highly regarded because of their provable approximate conservation of energy over long spans of simulated time. AVIs were previously demonstrated to enjoy these correctness properties while simultaneously allowing for efficient treatment of spatially non-uniform discretizations; however, a correct contact model remains unexplored.

Asynchronous collision detection To ensure safety, we require an equally principled approach to collision detection. This is a heavily studied problem; alas, the many reported successes are specific to the synchronous context, and as a group current methods can be intractably slow if naively applied after each local asynchronous step. This motivates our interest in *kinetic data structures* (KDSs) [Basch et al. 1999]: a KDS algorithm maintains a data structure governed by formal invariants describing some discrete attribute (such as absence of collisions), in response to the continuous movement of geometric elements. Many existing collision detection methods can be reformulated from a KDS perspective. KDSs seem destined for asynchronous applications, because their focus on fast, minimal, “output-sensitive” data-structure updates makes them ideally suited for the small, local changes effected by each AVI step. And yet, while KDSs are the perfect suitor for AVIs with their safety complementing AVIs’ correctness, no such matrimony has yet been considered.

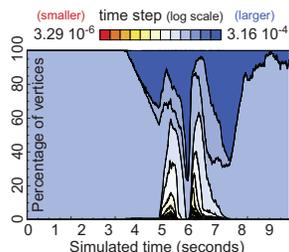


Figure 2. Asynchrony in the curtain simulation, depicted by the time-evolving distribution of vertex time step sizes, enables adaptive allocation of computational resources in spacetime.

Contributions These observations motivate our interest in approaching contact mechanics for both graphics and mechanics applications from a new direction. In particular, (a) we formulate a contact model that is *safe* independent of user parameters; (b) we *correctly* discretize time, using asynchrony to preserve the model’s safety and to respect causality, and using a symplectic-momentum integrator to exactly conserve momentum and approximately conserve energy over long run times. (c) We lay out the basic foundations for the union of AVIs with KDSs, making tractable the safe, correct integration of complex contact for highly deformable objects. Finally, we (d) expose a simple model of dissipation and friction that preserves symmetries of immersion and behaves consistently across changes to temporal discretization.

Method in brief Throughout our exposition, we will refer to line numbers of Algorithm 1, which summarizes the event-driven simulation loop. Rather than keeping the entire configuration synchronized in time, each vertex i stores its “most recently seen” position \mathbf{x}_i and velocity $\dot{\mathbf{x}}_i$ as recorded at time t_i . Events, each embodying some simple local atomic action, are drawn and processed in causal order (see algorithm LINE 2). The state of all vertices in the *stencil* of this drawn event must be advanced to the current time (LINES 3–7). When a force event is drawn, we apply impulses to the local stencil of vertices (see LINE 9 and §3); since the impulses affect the vertices’ future trajectories, we must update the continuous-time collision-detection data structures (see LINES 11–14 and §5). Some events embody data structure *certificate* updates but do not affect the trajectory (see LINES 15–18 and §5).

```

1: loop
2:   ( $E, V, h, t$ )  $\leftarrow Q.pop$            // Pop event  $E$  with potential  $V$ , time step  $h$ ,
                                       // and scheduled time  $t$ , from time-ordered queue  $Q$ 
3:    $\xi := stencil(E)$                    // global indices of the local stencil
4:   for  $i \in \xi$  do
5:      $\mathbf{x}_i \leftarrow \mathbf{x}_i + (t - t_i)\dot{\mathbf{x}}_i$  // advance vertex to current time (see §3)
6:      $t_i \leftarrow t$                    // update vertex's clock
7:   end for
8:   if  $E$  is a (external, internal, contact) force event then
9:      $\dot{\mathbf{q}}_\xi \leftarrow \dot{\mathbf{q}}_\xi - hM_\xi^{-1}\partial V/\partial \mathbf{q}_\xi$  // local impulses, local mass (see §3)
10:     $Q.push(E, V, h, t + h)$  // Return the event to the queue, with new time
11:    for  $j \in \bigcup_{i \in \xi} contingent(i)$  do
12:       $s \leftarrow failureTime(E_j)$  // compute new event time (see §5.1)
13:       $Q.update(E_j, s)$  // reschedule the contingent event (see §5.2)
14:    end for
15:    else if  $E$  is certificate failure then
16:      update KDS certificate, reschedule in  $Q$  // see §5.1 and §5.3
17:      (de)activate penalty forces // see §4
18:    end if
19:  end loop

```

2 Related work

Computational Contact Mechanics is a well-studied problem [Wriggers and Laursen 2007; Johnson 2008] of constraint enforcement: a physical trajectory travels only through the *admissible region*—the subspace of collision-free configurations (see Fig. 3). Framing collision response as an instance of constraint enforcement enables future generalizations of our method to other constraints (*e.g.*, inextensibility enforcement in §7).

To enforce constraints, engineers turn to *penalty* forces. As noted by Wriggers and Panagiotopoulos [1999], analysis begins with the impulsive penalty force, an infinite spike where bodies are in contact and zero elsewhere. The spike is impossible to model with a conservative force, necessitating approxima-

tion with quadratic or higher-order *penalty potentials*. In deviating from true impulses, penalty potentials permit visible penetration; stiffening the force helps, but it also induces smaller time steps. On the other hand, a low stiffness leads to disastrous *tunneling* through the inadmissible region (see Fig. 3). These drawbacks motivate adoption of *Lagrange multipliers* and unilateral contact laws [Pfeiffer and Glocker 2000; Eck et al. 2005], where constraint-enforcing balance constraint-violating forces. Multiple simultaneous contacts induce linear complementarity problems (LCPs) [Wriggers and Laursen 2007], with their attendant complexity and numerical pitfalls.

Graphics and robotics have embraced these developments, extending them with an eye to simplicity and efficiency. Terzopoulos et al. [1987] used penalty methods to treat contact between elastic bodies. Hahn [1988], Mirtich and Canny [1995] used impulses, viewing contact as micro-collisions, while Baraff [1989; 1994], Stewart and Trinkle [1996] presented LCP treatments for multiple simultaneous contacts with friction. Specifically targeting complex cloth collisions, Bridson et al. [2002] present a velocity filter that combines the advantages of penalty and impulsive methods, and relies on a geometric approximation for difficult impact zones [Provot 1997; Harmon et al. 2008]; geometric approaches are also instrumental in resolving pinching and other challenging configurations [Baraff et al. 2003; Volino and Magnenat-Thalmann 2006; Sifakis et al. 2008]. Recently, Guendelman et al. [2003] and Kaufman et al. [2005; 2008] treated complex stacking and friction for rigid bodies. An attempt to directly incorporate these collision algorithms into AVIs faces two challenges: many methods amortize cost by assuming temporal synchronization; a straightforward interleaving of contact-response and symplectic integration algorithms breaks the latter’s good momentum and energy behavior (see §3).

Several works consider asynchronous handling of contact. Lubachevsky [1991] used an event-driven priority-queue algorithm to simulate billiard balls, Celes [1998] handled contact between multiple mass-spring bodies, and such approaches extend to granular materials [Pöschel and Schwager 2005]. Mirtich [2000] enabled aggressive advancement of rigid body simulations with provably-correct partial-state rollback to fix missed collisions. Debonne et al. [2001] considered multirate time integration for simulation of visco-elastica. Dequidt et al. [2004] reframed asynchrony from an autonomous agent perspective. Thomaszewski et al. [2008] applied AVIs to cloth simulation, using a three-pass approach that aims to efficiently resolve collisions.

What sets our work apart is the focal triad of safety, correctness, and progress. Methods that prioritize progress by relaxing correctness can have downstream costs of simulation setup, feature development, and artifact resolution. For example, many popular methods for cloth simulation justifiably assume a zero coefficient of restitution (COR). These assumptions can be so deeply ingrained that allowing adjustment of CORs is impossible without a major overhaul or painstaking parameter-tuning. As another example, local (“Gauss-Seidel” or “Jacobi”) iterative techniques essentially optimize for the case of light collisions, resorting to (unphysical) “fail-safes” when the going gets tough.

In summary, the mechanics literature describes physical models for contact, but lacks many of the sophisticated algorithms considered by computer scientists; meanwhile, the trend in graphics has been to start with a fast but approximate solution, and then to chip away

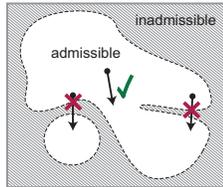


Figure 3. Trajectories (arrows) must remain in the admissible (white) region.

at the unphysical artifacts and the lack of scalability. By contrast, we begin with a more costly, but geometrically safe and physically conservative method, and build up efficiency using tools such as asynchrony and persistence.

3 Asynchronous variational integrators

Consider a mechanical system with a time-varying configuration $\mathbf{q}(t)$ in the space \mathbf{Q} of all configurations; concretely, for a mesh with vertices $\mathbf{x}_1, \dots, \mathbf{x}_n$ in 3D we represent $\mathbf{Q} = \mathbb{R}^{3n}$ by a vector of all the vertices’ Cartesian coordinates. We use a dot to denote differentiation in time, so that $\dot{\mathbf{q}}(t)$ is the configurational velocity. Let M be the mass matrix, so that $\mathbf{p} = M\dot{\mathbf{q}}$ is the momentum.

The Verlet integrator evolves a sequence of positions $\mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2 \dots$ and momenta $\mathbf{p}^0, \mathbf{p}^1, \mathbf{p}^2 \dots$ via the update rules

$$\mathbf{q}_k - \mathbf{q}_{k-1} = hM^{-1}\mathbf{p}^{k-1}, \quad \mathbf{p}^k - \mathbf{p}^{k-1} = hF(\mathbf{q}_k), \quad t_k - t_{k-1} = h,$$

where h is the time step and $F(\mathbf{q})$ is the force. The sub/superscripted indices allude to the method’s alias, *leapfrog*, reminding us that positions and velocities are staggered in time, with t_k associated to \mathbf{q}_k , and (t_k, t_{k+1}) associated to \mathbf{p}^k . In effect, Verlet first updates the position at t_k using the constant momentum associated to the preceding interval (t_{k-1}, t_k) (Algorithm LINE 5), and then impulsively “kicks,” obtaining a new momentum for the following interval (t_k, t_{k+1}) (Algorithm LINE 9), yielding a piecewise linear (p.l.) trajectory over the intervals (t_k, t_{k+1}) . A *geometric integrator* [Hairer et al. 2002; Kharevych et al. 2006], Verlet tracks conservation laws (e.g., mass, momentum, energy) and adiabatic invariants (e.g., temperature) over long run times, and offers more consistency and qualitatively predictable behavior across a range of time step sizes.

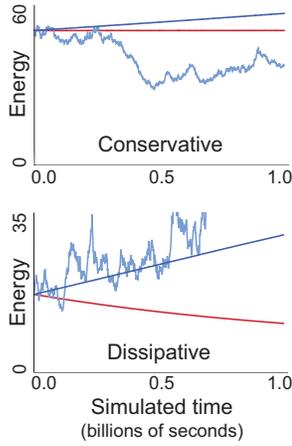
AVIs naturally extend Verlet. Each force receives an independent, regular (fixed-rate) clock, fixed a priori by stability requirements. While impulses of a force are regularly spaced in time, the superposition of forces yields events irregular in time. As with Verlet, the trajectory is p.l., interrupted by “kicks.” When their clocks are nested—as quarter notes are nested in half notes—AVIs reduce to an instance of multisteping methods [Hairer et al. 2002]. Our developments apply to this family of methods.

For example, Lew et al. [2003] assign an elastic potential to each mesh element. Irregular meshes have spatially-varying element shapes and corresponding time step stability restrictions; with AVIs each element advances at its own pace. Since an elemental potential depends only on a local mesh neighborhood, each integration event is *local*, affecting the position and velocity of a small number of *stencil* vertices. Correspondingly, Algorithm LINE 9 uses the local forces and mass matrix.

To schedule the interrupts to the p.l. trajectory, AVIs use a priority queue, conceptually populated with all event times until eternity. In practice it suffices to schedule only the next tick for each clock, since that event can schedule the subsequent tock (LINE 10).

Ensuring correctness A more complete analysis leading to the geometric and conservation properties of AVIs invokes ideas from discrete mechanics and variational integration [Marsden et al. 1998; Lew et al. 2003]. Here we stress a key outcome: Lew et al. conjecture that AVIs’ remarkable properties are due to its *multisymplecticity* (a property we further develop in a technical report [Vouga et al. 2009]); the derivation requires each force to have a regular (constant-rate, ever-ticking) clock. Playing with this clock—accelerating or pausing—is strictly forbidden. Interrupting the p.l. trajectory with other mechanisms (e.g., interleaving a velocity filter) breaks multisymplecticity.

We demonstrate the perils of tampering with the clock. A free spring of unit stiffness, rest length, and endpoint masses is placed one unit above a ground plane, in a vertical “pogo stick” orientation, and allowed to bounce repeatedly on the ground under gravity. We simulate thrice: with ordinary AVIs for gravity, the spring, and contact penalty (red horizontal line); as before, but restarting the contact penalty clock at the instant of contact (dark blue sloped line); and with AVIs for gravity and the spring, but resolving collisions with a reflective impulse at the contact instant (light blue jagged curve). The time-evolution of total energy reveals that first approach has no evident energy drift, whereas the second systematically injects energy and the third takes a random walk. Good energy behavior is equally important for dissipative systems. We add a small dashpot and repeat the experiment. Only the regularly clocked penalty force yields the expected, controllable energy dissipation.



In large-scale simulations, we observe that tampering with the clock leads to instabilities and inconsistent behavior across mesh resolutions. Supporting the observed difficulties, Zhong and Marsden [1988] prove that symplectic-momentum-energy preserving methods of regular time step do not exist (except for certain integrable systems); one cannot hope to interleave an energy-momentum collision integration with a symplectic-momentum force integration and retain either set of properties.

AVIs and contact To the best of our knowledge, the problem of extending AVIs to handle contact mechanics remains open. The conservation properties of AVIs rely on preservation of the multisymplectic form [Marsden et al. 1998; Marsden et al. 2001], and are easily broken by naively incorporating existing contact-resolution methods. A principled treatment must consider a multisymplectic formulation of contact mechanics, and an asynchronous computation of collision detection and response.

4 Discrete penalty layers

Consider a simple penalty method that penalizes proximity between bodies. For a given surface thickness η , the gap function

$$g_\eta(\mathbf{q}) = \|\mathbf{x}_b - \mathbf{x}_a\| - \eta$$

tracks signed proximity between moving points \mathbf{x}_a and \mathbf{x}_b . When $g < 0$, the points are said to be *proximate*. We can express the contact (or “interaction”) potential and force in terms of g

$$V_\eta^r(g(\mathbf{q})) = \begin{cases} \frac{1}{2}rg^2 & \text{if } g \leq 0 \\ 0 & \text{if } g > 0, \end{cases} \quad \mathbf{F} = \begin{cases} -rg\nabla g & \text{if } g \leq 0 \\ 0 & \text{if } g > 0, \end{cases}$$

respectively, where r is the contact *stiffness*. Choosing a penalty stiffness is the most criticized problem of the penalty method [Baraff 1989]. For any fixed stiffness r , there exists a sufficiently large approach velocity such that the contact potential will be overcome by the momentum, allowing the configuration to tunnel illegally through an inadmissible region (see Fig. 3).

The *barrier method* replaces the above contact potential by a function that grows unbounded as the configuration nears the boundary $g(\mathbf{q}) = 0$, eliminating the possibility of tunneling. However,

such a function must also have unbounded second derivative, ruling out stable fixed-step time integration for *any* choice of step size [Hairer et al. 2002].

To alleviate these concerns, we propose a construction consisting of an infinite family of *nested potentials*

$$V_{\eta(l)}^{r(l)}, \quad l = 1, 2, \dots,$$

where $\eta(l)$ is a monotonically decreasing proximity (or “thickness”) for the l -th potential, and $r(l)$ is a monotonically increasing penalty stiffness. For these nested potentials to be a barrier, the cumulative energy of these potentials must diverge as the distance between two primitives vanishes:

$$\sum_l r(l)\eta(l)^2 \rightarrow \infty.$$

We use $r(l) = r(1)l^3$ and $\eta(l) = \eta(1)l^{-1/4}$ [Vouga et al. 2009], where $r(1)$ and $\eta(1)$ are a simulation-dependent base stiffness and thickness for the outermost layer.

We call the region $\eta(n+1) \leq g(\mathbf{q}) \leq \eta(n)$, where exactly n of the potentials are nonzero, the n -th *discrete penalty layer* (see Fig. 4). The nested potentials’ respective maximal stable time steps form a decaying sequence, and therefore this construction *requires* an adaptive or asynchronous time stepping algorithm. Each interaction potential has its own integration clock, and has the opportunity to apply an impulse when its clock ticks. The question is how to time step such an infinite sequence.

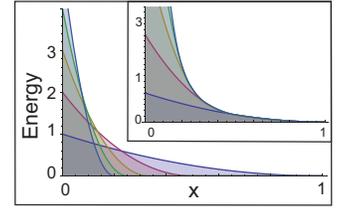


Figure 4. *Discrete penalty layers.* Potential energy of layer n plotted against proximity; *Inset:* total potential energy contributed by all layers $\leq n$. The potential energy diverges as x_a approaches x_b , guaranteeing that constraint enforcement is robust.

As we are about to see, the above construction transforms a seemingly intractable problem in Computational Mechanics—establishing a multisymplectic treatment of contact mechanics with *guaranteed* absence of tunneling—into a challenging but addressable problem in Computer Science: efficient bookkeeping on a conceptually infinite set of interaction potentials.

Central observation. During any time interval, while conceptually the (infinite number of) clocks continue to tick, and the totality of the clock ticks is dense in time, only a *finite, sparse* set of clock ticks apply (non-zero) impulses. In particular, the index of the discrete penalty layer indicates the number of *active* potentials; the rest, while conceptually present, do not influence the trajectory, and can be culled without approximation. What is needed is efficient bookkeeping to track which interaction potentials are active; each status change corresponds to a transition between penalty layers—a *discrete* change in state due to motion along a *continuous* trajectory. This is a problem that KDSs were born to solve.

5 Kinetic Data Structures for AVIs

Guibas [1998] gives an overview of kinetic data structures. Our culling of inactive forces uses an implementation of kinetic separating slabs, closely related to those used by Guibas et al. [2001b] in the context of rigid polytopes.

5.1 Kinetic separating slabs

As an illustrative example, consider a single particle falling toward a fixed floor. Conceptually, the clock for the first penalty layer is always ticking; however, it is active (exerting a nonzero impulse) only when the particle drops below height $\eta(1)$, say at time t . We must “activate the clock,” placing it on the priority queue for explicit consideration, no later than time t . Activating too late introduces error (misses impulses), while activating too early is correct, albeit overly conservative (some null events are not culled).

Suppose that calculating t is expensive. A conservative optimization uses an $\eta(1)$ -slab—a line extruded to thickness $\eta(1)$ —separating the particle from the floor. The separating $\eta(1)$ -slab serves as a proof, or *certificate*, that the particle and floor are at least $\eta(1)$ apart. This guarantee remains valid until either the floor or the particle enters the slab, at which point the certificate *fails*: we can try to find a new slab, or if doing so is costly or impossible, activate the first penalty force.

Concretely, simulation begins with identifying an $\eta(1)$ -slab (see Fig. 5); from the initial vertex state, and assuming a straight line trajectory, we compute the time t when the particle enters the slab (LINE 12), and schedule this certificate *failure event* on the priority queue (LINE 13). If t is hard to compute, any earlier time $t_1^{cf} < t$ is correct but conservative.

At time t_1^{cf} , the failure event pops off of the queue (LINE 2). We check the separation distance; suppose it exceeds $\eta(1)$. We identify a new $\eta(1)$ -slab (LINE 16), and schedule a new failure event, say at time t_2^{cf} (see Fig. 6).

Suppose that the next event, at time $t^g < t_2^{cf}$, corresponds to integration of gravity. We integrate the particle position, based on its last-known state and the elapsed time (LINE 5); we integrate the particle velocity based on the gravitational force (LINE 9). Failure time t_2^{cf} was computed assuming a constant velocity, an assumption now broken; t_2^{cf} might no longer be conservative, so the failure event must be rescheduled to guarantee safety (LINES 11–14).

The simulation continues in this manner. As the particle approaches the floor, the benefits of culling clock ticks of penalty layer one are eventually outweighed by the increasing frequency of $\eta(1)$ -slab events. Our implementation considers the trade-off to occur when the separation distance is below $\frac{11}{10}\eta(1)$; the decision of how to flag this trade-off affects performance but not safety or correctness.

When the use of $\eta(1)$ -slabs is no longer considered profitable, we activate the layer-one penalty clock (LINE 17) and forgo $\eta(1)$ -slabs (see Fig. 7). We cull clock ticks of only deeper layers. We certify inactivity by identifying an $\eta(2)$ -slab, computing and processing failure, reconstruction, and rescheduling as described above.

With the layer-one clock active, we soon encounter a layer-one penalty force integration event. This event is treated in the same manner as the gravity event or any force event. Furthermore, this

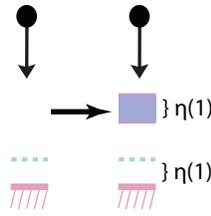


Figure 5. A separating slab KDS is created as proof of no contact.

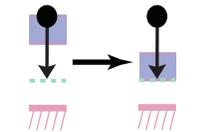


Figure 6. A new separating slab is scheduled.

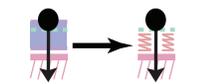


Figure 7. No efficient slab exists, so a penalty force is activated.

Event	Supporting vertices	Stencil vertices
Gravity		Entire mesh
Stretching force [Lew et al. 2003]		Triangle
Bending force [Grinspun et al. 2003]		Hinge
Penalty force (§4)		Pair of primitives
Separation slab (§5.1)	Pair of primitives	
k -DOP overlap (§5.5)	Those in k -DOP	
Render frame		

Table 1. Events and their associated supports and stencils.

event serves as an opportunity to check whether the particle is transitioning to a shallower penalty layer: if (a) the penalty impulse is null, *i.e.*, separation distance exceeds $\eta(1)$, and (b) the relative velocity is separating rather than approaching, then we de-activate the penalty force, transitioning to the next-shallower layer, and adjusting the certificates accordingly. This lazy approach to deactivation is safe by clause (a) alone; clause (b) aids in efficiency, avoiding rapid toggling of penalty layers.

5.2 Stencils, supports, and scheduling dependencies

With a basic depiction of a KDS in place, we proceed to discuss efficiency and optimization, after laying out the requisite terminology. Consider the execution of an event at its scheduled time. The set of vertices whose velocities are altered is the *stencil* of the event. The set of vertices whose trajectory was used to schedule this time is the *support* of that event. Building on the notions of stencil and support, an event *depends*, or is *contingent*, on another if the support of the former overlaps the stencil of the latter; vice versa, an event *supports* another if the stencil of the former overlaps the support of the latter. Table 1 shows the support and stencils for a set of typical events.

To our knowledge, KDSs were previously applied only to synchronous simulations, where the velocities of all primitives are updated at the same instant, *i.e.*, the stencil of *the* force-integration event contains the set of *all* vertices. By contrast, in an AVI simulation, force-integration events typically bear small stencils.

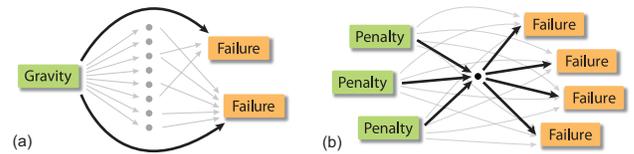


Figure 8. Directed graphs depicting events (boxes), vertices (dots), and dependencies (directed edges). Integration events (left green boxes) alter vertex trajectories, forcing rescheduling of dependent events (right orange boxes). (a) If an integration event has a large stencil, we store event-event dependencies. (b) If a vertex belongs to multiple stencil and support relations, we store event-vertex-event dependencies.

Having executed a supporting event, we must reschedule all dependent events before proceeding (Algorithm LINES 11–14). This is a problem of executing partially ordered instructions with dependencies, and it is thoroughly studied in the computer systems literature [Korneev and Kiselev 2004].

Our implementation maintains a directed graph, where edges from events to vertices and vice versa denote stencil and support relations, respectively. When an event executes, the two-neighborhood of outgoing edges yields the set of events to reschedule. The graph abstraction reveals that events with large stencils, such as gravity, should cache a list of contingent events, while events with small stencils should construct the list of contingent events on-the-fly; refer to Figs. 8a and 8b, respectively.

5.3 Thinning out certificate rescheduling

Every velocity update requires the rescheduling of dependent events. This rescheduling tends to be too costly and so frequent that it becomes intractable; these drawbacks are recognized in the KDS literature [Guibas et al. 2001a; Guibas et al. 2004]. We introduce the notion of vague trajectories to safely reduce the frequency of rescheduling.

Certificates are rescheduled when a supporting trajectory is altered. Because we are using KDSs specifically in the context of contact mechanics, we can bring into play physical insights that would otherwise not be at our disposal. As an illustrative example, consider Newton’s apple, which after being tossed into the air follows a parabolic trajectory before hitting the ground. We now split the apple and connect the two halves with a stiff spring. When we toss the apple once more, what happens? Since the two halves quickly oscillate against each other, the trajectory of each half has many wiggles—changes in velocity. Even so, the trajectory of the center of mass is exactly parabolic and, ignoring the high-frequency wiggles, the trajectory of each half is “overall” parabolic. Most importantly, unless the half-apple is very close to the floor, the parabola serves as an excellent predictor of the collision time with the floor, while the velocity associated to the rapid oscillations is noisy. This noise is twice detrimental: it impoverishes the collision time estimate, and, worse, it causes frequent rescheduling.



To harness this insight, we consider trajectories with bounded uncertainty. In place of precise linear trajectories, we consider “tubes” wide enough to encompass the noisy oscillations. On the one hand, this requires us to compute certificate expiration times that are conservative in the sense that they are valid for any precise trajectory that fits in the tube. On the other hand, the certificate will remain valid, despite noisy changes to the future trajectory, or *flightplan*, so long as the current trajectory remains inside the tube. If the predicted tube is not too thick, and if the actual trajectory remains inside the predicted tube for sufficient time, we could potentially reap a (safe, correct) dramatic reduction in rescheduling.

We pursue a simple implementation motivated by this idea. Recall our scheduling approach for the simple separating slab KDS. After creating a new certificate (say at time $t = t_0$), we scheduled a certificate failure time by solving for the time at which the particle enters the slab *assuming a constant velocity*. Because of this restrictive assumption, even a small impulse necessitated event rescheduling.

To introduce vagueness, we weaken the assumption to allow for a time-varying velocity. We therefore let the velocity of the particle $\dot{\mathbf{x}}(t) = \dot{\mathbf{x}}(t_0) + \mathbf{u}(t)$, where $\mathbf{u}(t)$ is a time-varying vector of bounded length $\|\mathbf{u}(t)\| \leq \varepsilon$. The relaxed assumption has two implications. First, it is now possible for many impulse events to affect the particle without necessitating a certificate rescheduling, so long as each impulse keeps $\|\dot{\mathbf{x}}(t) - \dot{\mathbf{x}}(t_0)\| \leq \varepsilon$. Indeed, for $\varepsilon < \|\dot{\mathbf{x}}(t_0)\|$, there is a *cone* of trajectories that avoid rescheduling. Second, the computation of the failure time must be conservative over all future trajectories satisfying the relaxed assumption, *i.e.*, we must compute the earliest possible failure time. For the separating slab, the trajectory producing the earliest failure “worst case” failure is the one maintaining $\|\mathbf{u}(t)\| = \varepsilon$ with $\mathbf{u}(t)$ in the direction of the slab.

Increasing ε reduces rescheduling frequency, since it widens the cone of covered trajectories; unfortunately, it also increases the frequency of certificate failures, since the worst-case trajectory reaches the slab sooner; these two considerations must be balanced.

Fortunately, any choice of ε keeps the system safe—the choice of ε cannot alter the actual simulated trajectory.

5.4 Broad phase

Our implementation begins with the simple separating slab KDS described above, modified so that slabs have constant (rather than zero) normal velocity. We consider this the “narrow phase.”

While formally correct, the simple KDS used on its own will not scale efficiently to large scenes. Various sophisticated KDSs track proximity, offer better “broad-phase” scaling, and could be easily adapted to the bookkeeping of the DPL index [Basch et al. 1997; Erickson et al. 1999; Guibas et al. 2001a; Agarwal et al. 2002; Gao et al. 2003; Agarwal et al. 2004; Gao et al. 2005]. We have not implemented all the available methods, thus rather than advocating for one candidate, we dedicate our exposition (recall §5.2–5.3 and see §5.6 below) to those concepts particular to the synthesis of AVIs with KDSs, independent of the chosen KDS. For completeness we briefly describe our implementation of a broad-phase KDS, then return to cross-cutting concepts.

5.5 Kinetic k -DOP hierarchy

A k -discrete oriented polytope (k -DOP) is a bounding volume (BV) described by $k/2$ real intervals $S_i = [\alpha_i, \beta_i]$, $1 \leq i \leq k/2$, each describing an object’s extent (or “support”) along some predetermined supporting axis \mathbf{d}_i [Konečný and Zikan 1997]. For $k = 6$ and orthogonal axes, k -DOPs reduce to axis-aligned bounding boxes (AABBs). For $k \rightarrow \infty$, k -DOPs approximate convex hulls.

Like most BVs, k -DOPs work best in a hierarchy whose leaves bound primitives and progressively coarser levels bound aggregates [Klosowski et al. 1998]. In synchronous simulations, a collision step updates (“rebuilds”) and traverses (“broad phase detection”) the entire hierarchy; the cost is amortized over the consideration of all pairwise collisions in the scene. This economy of scale does not immediately carry over to AVIs, where each integration step updates only a handful of primitives.

The KDS amortizes not over space but over time: as the position of primitives continuously evolve, we identify discrete transitions in execution flow of the broad-phase traversal algorithm. Such a *kinetization* of the hierarchy traversal is described in detail by Weller and Zachmann [2006] in the context of AABBs and synchronous simulations, but the general idea of incrementally updating a collision-detection tree traversal is known well beyond the KDS literature [Ericson 2004]. In retrospect, the approach seems to fit most naturally as a component of an asynchronous simulation, yet we are not aware of prior work harnessing the natural affinity of KDSs and AVIs.

For implementation details, we refer the reader to Weller and Zachmann’s exposition [2006]. Here we describe only the k -DOP and AVI-specific concepts that the former work did not explore.

Since a hierarchical BV algorithm decides whether to recurse by testing the overlap of two k -DOPs, the associated kinetic proof uses overlap and non-overlap certificates; the failure times thus correspond to instants at which two k -DOPs become (non-)overlapping.

Consider a certificate guaranteeing non-overlap of k -DOPs a and b . The simplest proof identifies a single axis \mathbf{d}_i with disjoint extents. This proof is valid during time intervals where $S_i^a \cap S_i^b = \emptyset$, *i.e.*, $\alpha_i^b - \beta_i^a > 0$ or $\alpha_i^a - \beta_i^b > 0$. Following our didactic example (§5.1), assume that the vertex trajectories are linear in time. Since a k -DOP contains multiple vertices of differing positions and velocities, the upper extent $\beta_i^a(t)$ is convex piecewise linear in time,

where kinks correspond to a new “leader” overtaking the extremal vertex (see Fig. 9); likewise, $\alpha_i^a(t)$, $\alpha_i^b(t)$, $\beta_i^b(t)$ are convex p.l. Therefore, computing the failure time reduces to finding the first instant at which a p.l. function becomes negative.

A more ambitious certificate uses the k -DOP’s *multiple* axes, and the observation that the failure of one axis need not bring down the whole certificate. Each axis \mathbf{d}_i yields a set of time intervals where the a and b are separated; the certificate is valid over the union of all positive intervals associated to all $k/2$ axes, *i.e.*, it fails at the instant where all $k/2$ axes have negative p.l. functions.

5.6 Fast certificate scheduling

The Achilles’ heel of KDSs is the frequency and cost of rescheduling. We reduced rescheduling frequency using vague trajectories; now we explore reducing rescheduling cost.

Solving for a certificate failure time requires fast root finding techniques [Guibas et al. 2004]. Even when vertex trajectories are linear in time, the algebraic function represented by the certificate can have non-trivial algebraic complexity. We avoid these numerical issues by using only certificates whose failure times are roots of a univariate p.l. polynomial, requiring identification of the (“piecewise”) segment followed by a subtraction and a division for the (“linear”) solve.

The effort of computing a certificate failure time goes in vain when a supporting event executes in the interlude. The more distant the failure time, the more likely the wasted effort. Can we avoid precise scheduling without compromising safety or correctness?

The general answer is to quickly compute a *safe approximate time* guaranteed not to exceed the actual certificate failure time [Guibas et al. 2001a]. The associated *reconfirmation* event no longer implies a certificate failure; instead it reconfirms the current proof by attempting to find a future reconfirmation time. When a reconfirmation event fails to find a safe future time, we can either schedule a true failure time, or treat the failed reconfirmation as a (premature) certificate failure. Either approach is safe, but the latter (our choice) fully eliminates the typically more complex implementation of precise failure time computation. We implemented two conservative approximations:

Linear envelope on k -DOP extent: A p.l. function $f(t)$ over $t \geq 0$ can be bounded from above by a linear function $\hat{f}(t) = f(0) + mt$, with slope m the maximum over the slopes of the pieces of f (see Fig. 9); a bound from below follows similarly. We use this fact to find conservative (non-)overlap times for k -DOP extents.

Adaptively short-circuit to the most useful k -DOP axes: Not all k -DOP axes are created equal. Depending on the configuration, some excel while others fail in establishing separation. Can we process only the the useful axes, taking the intersection of their bounds, and thus reducing by a constant factor the $O(nd)$ computation of extremal velocities and positions? We achieve this in two steps: we first assume that the k axes are already (nearly) sorted from most-to least-useful, and we progressively improve our bound by incorporating an additional axis, until an axis fails to improve the bound; in the second step, we improve the sorting (for next time) by attempting to incorporate one random unused axis, and promoting

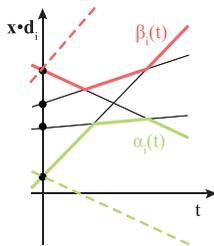


Figure 9. One axis of a k -DOP bounding 4 vertices over time. The bounds $\alpha_i(t)$ and $\beta_i(t)$ are linear functions. The dashed lines show the linear envelope which is a conservative bound for all $t > 0$.

this axis to the front of the list if it did improve the bound. For surface meshes, where k -DOPs have high aspect ratios described by a couple of axes, this approach is very effective. This idea can be understood in the language of *coresets* [Agarwal et al. 2005]; we dynamically update the coreset constituency as the system evolves.

6 Dissipative Forces

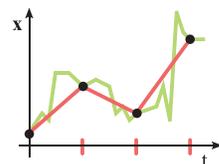
In using a geometric integrator, our approach exhibits energy near-conservation for long run times. Controlled dissipation, in the form of friction, impact coefficient of restitution, or viscous damping of high-frequency modes, is often desired in practical simulations. Our emphasis here is not on advanced models of dissipation, rather on basic ideas for incorporating *controlled* dissipation into our framework without compromising safety, correctness, and progress.

We ask that the limiting behavior as two events are brought to coincide should be *continuous*, *i.e.*, unique and independent of how this limit is approached. Simultaneous events must yield an order-independent outcome. If a perturbation to the problem setup (*e.g.*, time step size, initial vertex position) or a numerical error perturbs the order of nearly contemporaneous events, *reordering continuity* keeps the trajectory predictable.

Sufficient condition. If forces depend on positions, past and present, but *not* on momenta, then the trajectory is independent of the processing order of simultaneous events. *Proof:* an event outcome affects only future positions; a computation based on past and present positions is unaffected by outcomes of simultaneous events.

Conservative forces trivially satisfy this condition. However, since the most straightforward implementation of dissipation computes momenta-dependent quantities, it fails to meet our sufficient conditions, and generally leads to order-dependent outcomes.

Fortunately, a simple solution is at hand. Where a force formulation calls for momentum, we use the Verlet identity $\mathbf{p}^{k-1} = M(\mathbf{q}_k - \mathbf{q}_{k-1})/h$, a finite difference of past and present positions. To ensure a well-behaved, regular temporal discretization, we consider *only the positions associated to the dissipative force’s clock*. This corresponds to the average momentum in the interval between dissipative events, which, because of the interruptions induced by other asynchronous clocks, will generally not correspond to the momentum immediately preceding the dissipative event. All of this is illustrated in the adjacent figure where the red trajectory shows what is being used for the computation of the dissipative force, while the green trajectory includes updates from all events. We invoke this concept in discretizing several dissipative forces.



6.1 Viscous damping

Consider an elastic spring connecting two vertices i and j . Viscous damping acts to slow the *rate* at which the spring changes length. We create a new clock (we could also ride on the elastic clock, stability permitting) and compute a viscous force

$$\mathbf{F}_{k,i} = b(l_k - l_{k-1})\mathbf{e}_k/l_k = -\mathbf{F}_{k,j}, \quad \mathbf{e}_k = \mathbf{q}_{k,j} - \mathbf{q}_{k,i}, \quad l_k = \|\mathbf{e}_k\|,$$
where b is the damping coefficient, $\mathbf{q}_{k,i}$ is the position of the i -th vertex, and the vector \mathbf{e}_k and length l_k are local; a subscript (i, j) is implied. We cache l_{k-1} so that we can use its value at time t_k .

In the case of a single spring, the approach reduces to explicit Verlet integration of the viscous force, with its attendant time step restriction. Bridson et al. [2003] advocate a semi-implicit integration of

the viscous force. Such an approach might also be adopted in place of the one presented above, trading order independent processing of simultaneous events for larger viscous force time steps.

Most of our simulations incorporate some viscous internal (stretching and bending) damping. The curtains (see Fig. 1 and video) illustrate the benefit of starting from a conservative foundation. Using only internal damping, high-frequency vibrations introduced by the prescribed particle are quickly damped out, while the curtains’ graceful swinging continues; using a non-geometric integrator such as backward Euler or BDF2 [Ascher and Petzold 1998], the swinging motion would also be damped.

6.2 Coefficient of restitution

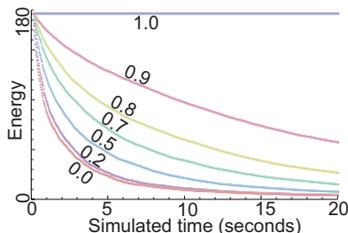
The coefficient of restitution e_{COR} is a “melting pot” approximation, accounting for various unresolved micro-level phenomena [Brilliantov and Pöschel 2004; Schwager and Pöschel 2007] including viscosity and plasticity. To model plastic work, we replace the nested penalty potentials with biphasic potentials [Choi and Ko 2005]

$$V_{\eta}^r(g(\mathbf{q})) = \begin{cases} \frac{1}{2}rcg(\mathbf{q})^2 & g \leq 0 \\ 0 & g \geq 0, \end{cases}$$

where c is e_{COR} if the primitives are separating, 1 otherwise. The penalty layers exert their full force during compression, then weaken according to the coefficient of restitution during decompression. We could (but did not) further extend this model to account for viscous damping during impact, measuring strain rate by (some monotonic function of) the change in the gap function $g(\mathbf{q})$.

While simple, our approach has a drawback in the inelastic limit $e_{\text{COR}} = 0$: the penalty impulses can leave as residue a small separating relative velocity. The magnitude of this velocity is at most $r(l)\eta(l)h$, where h is the layer’s time step, so it can be limited by choosing a small enough $r(l)$ or h .

The long-term good energy behavior accompanying our use of a symplectic-momentum integrator translates into predictable, controllable energy dissipation when a non-unit coefficient of restitution is used in a simulation. To test the energy behavior for a variety of coefficients of restitution,



we simulated a box of 900 particles with random initial velocities. The incident figure shows the energy of the system as a function of time for multiple values of e_{COR} ; in all cases energy decays smoothly and predictably.

Pöschel and Schwager [2005] describe experiments with granular media. They observe that large numbers of particles participating in frequent, dissipative collisions form *clusters*, or groups of proximate particles with very little relative velocity, over time. Fig. 10 illustrates that our method reproduces this clustering when the above experiment is run with $e_{\text{COR}} = 0$.

6.3 Friction

The Coulomb friction model serves as a simple approximation of an extremely complicated physical interaction. Consider the Coulomb friction force $F_f = \mu F_n$, where μ is the coefficient of friction and F_n is the normal force. The force opposes relative tangential motion between points in contact.

We apply friction along with each penalty force, separately for each penalty layer. Just as increasingly stiff penalty forces are applied for contact forces, friction forces are increasingly applied (bounded by each F_n) to correctly halt high-speed tangential motion.

Impulse-based collision response methods cap the magnitude of the Coulomb friction force, so that a large normal impulse does not cause relative tangential motion to reverse direction. Our implementation does not cap, because we have not identified a capping strategy that is compatible with order-independence of simultaneous events. For a pair of primitives in contact, friction is applied piecemeal, at the ticks of the penalty layer clocks, instead of as a single impulse. This serves as a reasonable discretization of kinetic friction, but it is certainly a crude approximation of static friction. In particular, it is possible for a friction update to reverse relative tangential motion; the magnitude of this reverse motion is bounded by $\mu r(l)\eta(l)h$, so it can be limited by choosing sufficiently small stiffness function r or time step h . Structures whose stability depends on static friction, such as the house of cards simulated by Kaufman et al. [2008], would benefit from future work developing a more complete treatment of friction.

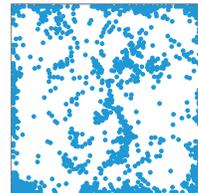


Figure 10. Dissipative collisions form characteristic clusters.

As a test of our friction model, we applied gravity to the box of particles described above, and allowed the particles to come to rest on the floor of the box. We then removed the right side of box and replaced it with a downwards slope. Fig. 11 shows the configuration of the balls 2.5s after removal of the wall; the result varies with the coefficient of friction. When no friction is applied, the particles flow freely down the slope. As friction is increased, the rate of flow decreases. Note that a simulation of granular materials should store as a state variable the angular momentum of each grain [Pöschel and Schwager 2005]; our implementation neglects this, evidence a small vertical stack of grains that slides down the inclined plane without tipping.

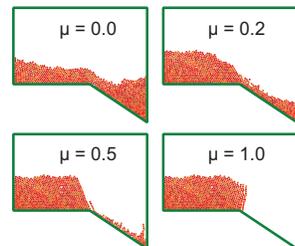


Figure 11. Friction alters the flow of sludge.

7 Generalization: Inextensibility constraint

As foreshadowed in §2, nested penalty layers can enforce a broader class of unilateral constraints. Consider edge inextensibility constraints in 2D. For two vertices i and j delimiting an edge of rest length ℓ , we constrain this edge to not compress or extend by $\pm s\ell$, for some $s \ll 1$. We introduce two nested sequences of penalty layers, their gap functions replaced by constraint functions

$$g_i(\mathbf{q}) = (1+s)\ell - \|\mathbf{x}_j - \mathbf{x}_i\|$$

$$g_j(\mathbf{q}) = \|\mathbf{x}_j - \mathbf{x}_i\| - (1-s)\ell.$$

The penalty layers associated with g_i and g_j prevent excess stretch and compression, respectively.

We simulate a cloth curtain hit by a fast-moving projectile (refer to supplemental video), comparing implementations based on constraints and elastic springs. For any chosen spring stiffness, a sufficiently energetic projectile stretches the curtain by arbitrary amounts, resulting in a “rubbery” curtain. On the other hand, enforcing inextensibility using nested penalty layers avoids stretching

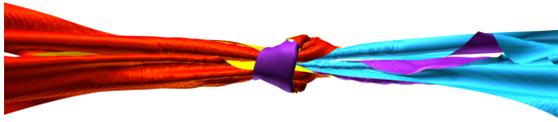


Figure 12. Simulated tying of ribbons into a reef knot.



Figure 13. A closeup of the reef knot.

no matter the projectile’s velocity. Implementation of the inextensibility constraint for 3D triangle meshes would require a constraint formulation that does not lock bending modes, such as that proposed by English and Bridson [2008].

8 Results

In §3 and §6, we described simple experiments and empirical measurements supporting the guaranteed safety and good energy behavior of the proposed contact algorithm, for both conservative and dissipative contact. We turn our attention to challenging problems involving complex contact geometries, sharp features, and sliding during extremely tight contact.

Knots We simulate the tying of ribbons into reef and bowline knots (see Figs. 12 and 14, respectively). The ribbons are modeled as a loose knot, assigned a material with stiff stretching and weak bending, and their ends are pulled by a prescribed force; the bowline knot requires also the prescription of fixed vertices behind the cylinder where a finger normally holds the material in place. The final configuration is faithful to the shape of actual “boyscout manual” knots.

This example demonstrates the strength of asynchrony in allocating resources to loci of tight contact. As the knot tightens, progressively finer time steps are used for the tightest areas of contact. If instead of prescribing reasonable forces we directly prescribe an outward motion of the two ends of the ribbon, the simulations execute to the point where the mesh resolution becomes the limiting reagent, *i.e.*, a tighter knot cannot be tied without splitting

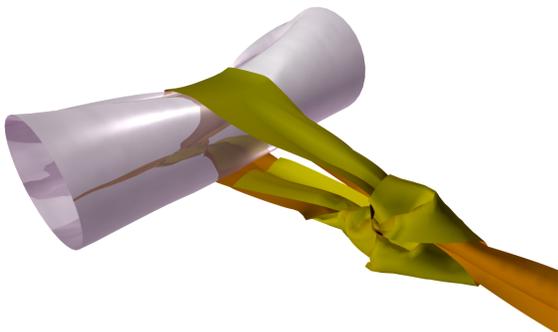


Figure 14. Simulated tying of a ribbon into a bowline knot.

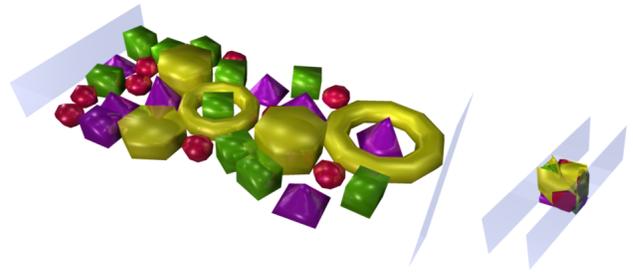


Figure 15. Virtual trash compactor and assorted virtual trash.

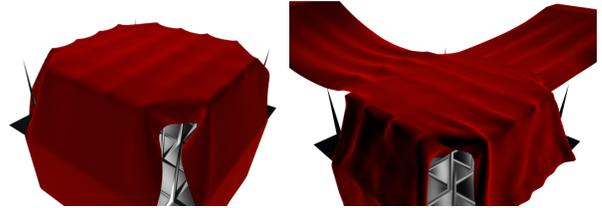
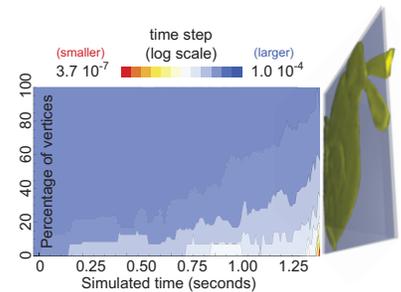


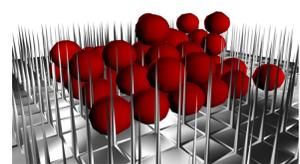
Figure 16. Experiments with a bed of nails highlight the method’s ability to deal with sharp boundaries, isolated points of contact, sliver triangles, and localized points of high pressure between two nearly incident surfaces.

triangles; past this point, the computation slows as penalty interactions burrow to deeper layers and the mean time step decays. This highlights both a feature and a potential artistic objection to the method: when presented with an impossible or nearly-impossible situation (non-stretchy ribbon with prescribed diametrically opposing displacements at its ends) the method’s safety guarantee induces Zeno’s Paradox.

Trash compactor We place triangle meshes of varying complexity into a virtual trash compactor consisting of a floor and four walls, and then prescribe the inward motion of opposing walls (see Fig. 15 and incident image). The method is able to simulate the approach of the walls without ever allowing for seen or unseen penetrations. As with the knots, the overall rate of progress decays as the simulation approaches a limiting configuration.



Bed of nails We crafted a problem to test the handling of isolated point contacts and sharp boundaries. Four sliver triangles are assembled into a nail, and many such nails are placed point-up on a flat bed. We drape two stacked fabrics over the bed of nails (see Fig. 16), and observe that the simulated trajectory is both realistic and free of penetrations, oscillations, or any other artifacts typically associated to contact discontinuities. Next, we prescribe the motion of one end of the fabric, tugging on the draped configuration to demonstrate sliding over sharp features.



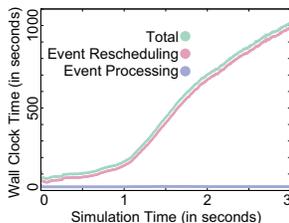
We extend the bed of nails into a landing pad for various coarsely-meshed projectiles. Variably-sized to barely fit or not fit between

the nails, and thrown with different initial velocities and angles, the projectiles exhibit a wide array of behaviors, including bouncing, rolling, simple stacking, ricocheting at high frequencies (this requires resolving each collision when it occurs, as resolving collisions over a fixed collision step size can cause aliasing that prevents the ricochet); sliding and getting stuck between nails (the sliding requires a deformable model and friction, since a perfectly rigid object would be constrained to a sudden stop by the distance

Timing We list computation time for the various examples, as executed on a single thread of a 3.06Ghz Intel Xeon with 4GB RAM. The bulk is allocated to the maintenance of the kinetic data structures used for collision detection. We measured the effect of introducing vague trajectories for the reef knot and bunny compactor, and observed at approximately 30% improvement in overall performance.

Examples	Vertices	Simulation Seconds	Event Processing (hours)	KDS Event Rescheduling (hours)	Total (hours)
Reef Knot	10642	2.00	1.5	16.7	18.5
Bowline Knot	3995	5.00	3.0	141.1	144.5
Trash Compactor	714	3.08	0.5	53.0	53.6
Two Sheets Draped	15982	3.95	4.5	260.8	265.5
Two Sheets Pulled	15982	3.83	13.6	310.5	325.6

As a more detailed study, consider that the reef knot simulation required 4.8% of total simulation time for integration of elastic forces and gravity, 0.09% for integration of penalty forces, 0.9% for processing and 1.0% for rescheduling of separating plane events, respectively, 5.2% and 23.0% for processing and rescheduling of separation list events, respectively. All other time was spent performing vague trajectory checks and queue maintenance. The incident figure demonstrates how per frame runtime increases as the stress on the ribbons elevates.



Parameters We list parameters for the various examples. Bending and stretching stiffness refers to the Discrete Shells [Grinspun et al. 2003] and common edge spring models.

Example	Density	COR	$r(1)$	$r(t)$	Stretching Stiffness	Stretching Damping	Bending Stiffness
Reef Knot	0.1	0.0	1000.0	0.1	750.0	0.1	0.01
Bowline Knot	0.01	0.0	1000.0	0.1	100.0	0.1	0.01
Bunny Compactor	0.01	0.01	10000.0	0.05	1000.0	0.0	1000.0
Trash Compactor	0.001	0.01	1000.0	0.05	1000.0	15.0	10.0
Two Sheets Draped	0.001	0.0	1000.0	0.1	1000.0	1.0	0.1
Reef Knot Untied	0.1	0.0	1000.0	0.1	1000.0	0.1	0.01
Two Sheets Pulled	0.001	0.0	1000.0	0.1	1000.0	1.0	0.1
Balls on Nails	0.016	0.3	10000.0	0.1	50000.0	1.0	100000.0
2D Sludge	-	0.0	1000.0	0.1	-	-	-

9 Discussion

Parameters and the triad of safety, correctness, and progress One of our driving goals is to investigate methods that ensure safety, correctness, and progress regardless of the choice of parameters. The method proposed here does expose some parameters to the user, such as the proximity η and the trajectory vagueness ϵ . These parameters affect performance, not the triad of guarantees. Our experience in running the problem scenarios, therefore, were qualitatively different than when using other methods, in that we did not need to search for parameters to ensure a successful modeling of

contact. On the other hand, our method does not address the spatial discretization of elasticity (stretching and bending models), which can also require user tuning.

Although in theory the nested penalty barrier has infinitely many penalty layers at its disposal, it is impractical to activate penalty layers whose stable time steps are too small, *e.g.*, below the floating point epsilon. Simulations with thicknesses $\eta(1)$ too small, or velocities or masses too high, can thus fail to make progress (but remain safe). This limitation can be worked around by choosing a slow-shrinking layer distribution function, which is why we recommend $\eta(t) = \eta(1)t^{-1/4}$. For more on this we refer to the accompanying technical report.

Multisteping methods such as AVIs are known to have *resonance instabilities* [Hairer et al. 2002; Fong et al. 2008], particularly if the simulation contains adjacent mesh elements of very different size. However, we have not observed any such instabilities or artifacts that we can attribute to such instabilities in our use of the method.

Broader exploration In this paper we were concerned with building the most robust contact implementation we could; therefore, we tied the knots as tight as possible, until each triangle was packed as tightly as possible into its neighbors. In the tightest configurations the spatial discretization becomes evident. It would therefore be interesting to introduce spatial adaptation, refining the mesh where curvature is high. Another alternative would be to improve the smoothness at render time, using for example the collision-aware subdivision of Bridson et al. [2002].

Dissipation and friction are important, complex topics deserving full publications of their own [Kaufman et al. 2005; Kaufman et al. 2008], and certainly more than the space allocated here. Our goal in this area was to provide some initial models that fit the method, and to demonstrate the controllability arising from a conservative foundation. Future work might explore efficient algorithms to handle stacking and static friction while still fitting the multisymplectic treatment.

Immediate and future impact In considering this method for immediate industrial use, we anticipate two important hurdles.

From the standpoint of incorporation into animation systems the first hurdle is the method’s insistence on safety even at the cost of artistic freedom. This effectively disallows all pinching [Baraff et al. 2003; Volino and Magnenat-Thalmann 2006], as well as commencing from invalid configurations. We believe that the method can be extended to permit shallow (“skimming”) pinching, but handling extremely unphysical boundary conditions within this framework seems at least initially at odds with the basic premise, and it will require further research.

Second, the proposed method is not competitive in performance compared to existing methods, which do not attempt to make strong safety and correctness guarantees; if an artist is willing to search for parameters that provide non-penetrating good-looking results, they may become impatient with the method proposed here.

From the standpoint of long-term, curiosity-driven research, however, this method is appealing not just in its formalism but also in terms of performance, since it lays out a formal asynchronous framework from which one can investigate parallelization, optimization, and even approximation techniques that preserve guarantees of safety, correctness, and progress. To aid such future investigation, source code for our initial C++ implementation, along with data files needed to generate the examples shown in this paper, are available online.

Acknowledgements We thank David Mooy for modeling the knots, and Igor Boshoyer, Matt Kushner, Kori Valz for lighting and rendering. We are grateful for the valuable feedback provided by Miklós Bergou, Rony Goldenthal, Bernhard Thomaszewski, Max Wardetzky, and the anonymous reviewers. This work was supported in part by the NSF (MSPA Award No. IIS-05-28402, CSR Award No. CNS-06-14770, CAREER Award No. CCF-06-43268) and the Amazon Elastic Compute Cloud. The Columbia authors are supported in part by generous gifts from Adobe, ATI, Autodesk, mental images, NVIDIA, the Walt Disney Company, and Weta Digital.

References

- AGARWAL, P., BASCH, J., GUIBAS, L. J., HERSHBERGER, J., AND ZHANG, L. 2002. Deformable free space tilings for kinetic collision detection. *Intl. J. Robotics Research* 21, 179–197.
- AGARWAL, P., GUIBAS, L., NGUYEN, A., RUSSEL, D., AND ZHANG, L. 2004. Collision detection for deforming necklaces. *Computational Geometry: Theory and Applications* 28, 137–163.
- AGARWAL, P. K., HAR-PELED, S., AND VARADARAJAN, K. R. 2005. Geometric approximation via coresets. In *Combinatorial and Computational Geometry*, J. E. Goodman, J. Pach, and E. Welzl, Eds. Cambridge University Press, New York, 1–30.
- ASCHER, U. M., AND PETZOLD, L. R. 1998. *Computer methods for ordinary differential equations and differential-algebraic equations*. Society for Industrial and Applied Mathematics, Philadelphia.
- BARAFF, D., WITKIN, A., AND KASS, M. 2003. Untangling cloth. *ACM Trans. Graph.* 22, 3, 862–870.
- BARAFF, D. 1989. Analytical methods for dynamic simulation of non-penetrating rigid bodies. In *SIGGRAPH '89: Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 223–232.
- BARAFF, D. 1994. Fast contact force computation for nonpenetrating rigid bodies. In *SIGGRAPH '94*, 23–34.
- BASCH, J., GUIBAS, L. J., AND ZHANG, L. 1997. Proximity problems on moving points. In *Proc. 13th Annu. ACM Sympos. Comput. Geom.*, 344–351.
- BASCH, J., GUIBAS, L. J., AND HERSHBERGER, J. 1999. Data structures for mobile data. *Journal of Algorithms* 31, 1–28.
- BRIDSON, R., FEDKIW, R., AND ANDERSON, J. 2002. Robust treatment of collisions, contact and friction for cloth animation. In *SIGGRAPH '02*, 594–603.
- BRIDSON, R., MARINO, S., AND FEDKIW, R. 2003. Simulation of clothing with folds and wrinkles. In *SCA '03*, 28–36.
- BRILLIANTOV, N. V., AND PÖSCHEL, T. 2004. *Kinetic Theory of Granular Gases*. Oxford University Press, USA.
- CELES, W. 1998. Efficient asynchronous evolution of physical simulations. In *SIBGRAPI '98: Proceedings of the International Symposium on Computer Graphics, Image Processing, and Vision*, IEEE Computer Society, Washington, DC, USA, 224.
- CHOI, K.-J., AND KO, H.-S. 2005. Stable but responsive cloth. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Courses*, ACM, New York, NY, USA, 1.
- CIRAK, F., AND WEST, M. 2005. Decomposition-based contact response (DCR) for explicit finite element dynamics. *Int'l Journal for Numerical Methods in Engineering* 64, 8, 1078–1110.
- DEBUNNE, G., DESBRUN, M., CANI, M.-P., AND BARR, A. H. 2001. Dynamic real-time deformations using space & time adaptive sampling. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 31–36.
- DEQUIDT, J., GRISONI, L., AND CHAILLOU, C. 2004. Asynchronous interactive physical simulation. Tech. Rep. RR-5338, INRIA.
- ECK, C., JANUŠEK, J., AND KRBEK, M. 2005. *Unilateral contact problems: variational methods and existence theorems*. Chapman and Hall/CRC Press, Boca Raton.
- ENGLISH, E., AND BRIDSON, R. 2008. Animating developable surfaces using nonconforming elements. In *SIGGRAPH '08: ACM SIGGRAPH 2008 papers*, ACM, New York, NY, USA, 1–5.
- ERICKSON, J., GUIBAS, L. J., STOLFI, J., AND ZHANG, L. 1999. Separation-sensitive collision detection for convex objects. In *Proc. 10th ACM-SIAM Symp. Discrete Algorithms*, 102–111.
- ERICSON, C. 2004. *Real-Time Collision Detection (The Morgan Kaufmann Series in Interactive 3D Technology)*. Morgan Kaufmann, December.
- FONG, W., DARVE, E., AND LEW, A. 2008. Stability of asynchronous variational integrators. *J. Comput. Phys.* 227, 18, 8367–8394.
- GAO, J., GUIBAS, L., HERSHBERGER, J., ZHANG, L., AND ZHU, A. 2003. Discrete mobile centers. *Discrete and Computational Geometry* 30, 1, 45–65.
- GAO, J., GUIBAS, L. J., AND NGUYEN, A. 2005. Distributed proximity maintenance in ad hoc mobile network. In *IEEE International Conference on Distributed Computing in Sensor System (DCOSS'05)*, 4–19.
- GRINSPUN, E., HIRANI, A., DESBRUN, M., AND SCHRÖDER, P. 2003. Discrete Shells. In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, 62–67.
- GUENDELMAN, E., BRIDSON, R., AND FEDKIW, R. 2003. Non-convex rigid bodies with stacking. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, ACM, New York, NY, USA, 871–878.
- GUIBAS, L., XIE, F., AND ZHANG, L. 2001. Kinetic collision detection: Algorithms and experiments. In *Proceedings of the International Conference on Robotics and Automation*, 2903–2910.
- GUIBAS, L. J., XIE, F., AND ZHANG, L. 2001. Kinetic collision detection: Algorithms and experiments. In *ICRA*, 2903–2910.
- GUIBAS, L., KARAVELES, M., AND RUSSEL, D. 2004. A computational framework for handling motion. In *Proceedings of the Sixth Workshop on Algorithm Engineering and Experiments*, 129–141.
- GUIBAS, L. J. 1998. Kinetic data structures — a state of the art report. In *Proc. 3rd Workshop on Algorithmic Foundations of Robotics (WAFR)*, 191–209.
- HAHN, J. K. 1988. Realistic animation of rigid bodies. In *SIGGRAPH '88: Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 299–308.
- HAIRER, E., LUBICH, C., AND WANNER, G. 2002. *Geometric Numerical Integration: Structure-preserving Algorithms for Ordinary Differential Equations*. Springer.

- HARMON, D., VOUGA, E., TAMSTORF, R., AND GRINSPUN, E. 2008. Robust Treatment of Simultaneous Collisions. *SIGGRAPH (ACM Transactions on Graphics)* 27, 3, 1–4.
- JOHNSON, K. L. 2008. *Contact mechanics*. Cambridge University Press.
- KAUFMAN, D. M., EDMUNDS, T., AND PAI, D. K. 2005. Fast frictional dynamics for rigid bodies. In *SIGGRAPH '05*, 946–956.
- KAUFMAN, D. M., SUEDA, S., JAMES, D. L., AND PAI, D. K. 2008. Staggered projections for frictional contact in multibody systems. In *SIGGRAPH Asia '08: ACM SIGGRAPH Asia 2008 papers*, ACM, New York, NY, USA, 1–11.
- KHAREVYCH, L., YANG, W., TONG, Y., KANSO, E., MARSDEN, J. E., SCHRÖDER, P., AND DESBRUN, M. 2006. Geometric, variational integrators for computer animation. In *SCA '06: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 43–51.
- KLOSOWSKI, J. T., HELD, M., MITCHELL, J. S. B., SOWIZRAL, H., AND ZIKAN, K. 1998. Efficient collision detection using bounding volume hierarchies of k-dops. *IEEE Transactions on Visualization and Computer Graphics* 4, 1, 21–36.
- KONEČNÝ, P., AND ZIKAN, K. 1997. Lower Bound of Distance in 3D. In *Proceedings of WSCG 1997*, vol. 3, 640–649.
- KORNEEV, V., AND KISELEV, A. 2004. *Modern Microprocessors*. Charles River Media.
- LEW, A., MARSDEN, J. E., ORTIZ, M., AND WEST, M. 2003. Asynchronous variational integrators. *Archive for Rational Mechanics And Analysis* 167, 85–146.
- LUBACHEVSKY, B. 1991. How to simulate billiards and similar systems. *Journal of Computational Physics* 94, 2 (June), 255–283.
- MARSDEN, J., PATRICK, G., AND SHKOLLER, S. 1998. Multisymplectic Geometry, Variational Integrators, and Nonlinear PDEs. *Communications in Mathematical Physics* 199, 2, 351–395.
- MARSDEN, J., PEKARSKY, S., SHKOLLER, S., AND WEST, M. 2001. Variational methods, multisymplectic geometry and continuum mechanics. *Journal of Geometry and Physics* 38, 3–4 (June), 253–284.
- MILENKOVIC, V. J., AND SCHMIDL, H. 2001. Optimization-based animation. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 37–46.
- MIRTICH, B., AND CANNY, J. 1995. Impulse-based dynamic simulation. In *WAFR: Proceedings of the workshop on Algorithmic foundations of robotics*, A. K. Peters, Ltd., Natick, MA, USA, 407–418.
- MIRTICH, B. 2000. Timewarp rigid body simulation. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 193–200.
- PFEIFFER, F., AND GLOCKER, C., Eds. 2000. *Multibody Dynamics With Unilateral Contacts*. SpringerWienNewYork, ch. 2, 69–146.
- PÖSCHEL, T., AND SCHWAGER, T. 2005. *Computational Granular Dynamics: Models and Algorithms*. Springer.
- PROVOT, X. 1997. Collision and self-collision handling in cloth model dedicated to design garments. In *Computer Animation and Simulation '97*, Springer Verlag, Wien, 177–189.
- SCHWAGER, T., AND PÖSCHEL, T. 2007. Coefficient of restitution and linear dashpot model revisited. *Granular Matter* 9, 6 (November), 465–469.
- SIFAKIS, E., MARINO, S., AND TERAN, J. 2008. Globally coupled collision handling using volume preserving impulses. In *2008 ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, 147–154.
- TERZOPOULOS, D., PLATT, J., BARR, A., AND FLEISCHER, K. 1987. Elastically deformable models. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 205–214.
- THOMASZEWSKI, B., PABST, S., AND STRASSER, W. 2008. Asynchronous cloth simulation. In *Computer Graphics International*.
- TRINKLE, D. S. J. 1996. An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction. *Intl. Journal for Numerical Methods in Engineering* 39, 2673–2691.
- VOLINO, P., AND MAGNENAT-THALMANN, N. 2006. Resolving surface collisions through intersection contour minimization. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, ACM, New York, NY, USA, 1154–1159.
- VOUGA, E., HARMON, D., TAMSTORF, R., AND GRINSPUN, E. 2009. Discrete penalty layers admit multisymplectic integration. Tech. rep., Columbia University.
- WELLER, R., AND ZACHMANN, G. 2006. Kinetic separation lists for continuous collision detection of deformable objects. In *Third Workshop in Virtual Reality Interactions and Physical Simulation (Vriphys)*.
- WRIGGERS, P., AND LAURSEN, T. A. 2007. *Computational contact mechanics*, vol. no. 498 of *CISM courses and lectures*. Springer, Wien.
- WRIGGERS, P., AND PANAGIOTOPOULOS, P., Eds. 1999. *New Developments in Contact Problems*. SpringerWienNewYork, ch. 1, 1–54.
- ZHONG, G., AND MARSDEN, J. E. 1988. Lie-Poisson Hamilton-Jacobi theory and Lie-Poisson integrators. *Physics Letters A* 133 (Nov.), 134–139.

Asynchronous Variational Contact Mechanics

E. Vouga^{a,*}, D. Harmon^a, R. Tamstorf^b, E. Grinspun^a

^aComputer Science Department, Columbia University, New York, New York 10027

^bWalt Disney Animation Studios, Burbank, California 91521

Abstract

An asynchronous, variational method for simulating elastica in complex contact and impact scenarios is developed. Asynchronous Variational Integrators [1] (AVIs) are extended to handle contact forces by associating different time steps to forces instead of to spatial elements. By discretizing a barrier potential by an infinite sum of nested quadratic potentials, these extended AVIs are used to resolve contact while obeying momentum- and energy-conservation laws. A series of two- and three-dimensional examples illustrate the robustness and good energy behavior of the method.

Keywords: contact, impact, variational integrators

1. Introduction

Variational integrators (VIs) [2, 3, 4] are a general class of time integration methods for Hamiltonian systems whose construction guarantees certain properties highly desirable of numerical simulations. Instead of directly discretizing the smooth equations of motion of a system, the variational approach instead discretizes the system's action integral. By analogy to Hamilton's least action principle, a discrete action can be formed, and *discrete* Euler-Lagrange equations derived by examining paths which extremize it. From the Euler-Lagrange equations, discrete equations of motion are readily recovered. As a consequence of this special construction, VIs are guaranteed to satisfy a discrete formulation of Noether's Theorem [5], and as a special case conserve linear and angular momentum. VIs are automatically *symplectic* [6]; while they do not necessarily conserve energy, conservation of the symplectic form assures no-drift conservation of energy over exponentially many time steps [6].

Given the many advantages of VIs, it is natural to apply them to the handling of contact and impact, a long-studied and challenging problem in physical simulation. Unfortunately, a naïve application of a contact algorithm to a variational integrator is not guaranteed to preserve the variational structure of the time integration method, and in practice one observes that the good energy behavior is lost. For this reason, a few recent works have explored structure-preserving approaches for contact mechanics [7, 8, 9, 10]. Common to all these approaches is a *synchronous* treatment of global time, in which the entire configuration is advanced from one instant in time to the next. While synchronous integration is attractive for its simplicity, it has the drawback that a spatially-localized stiff mode—such as that associated to a localized contact—can force the global configuration to advance at fine time steps.

Indeed, mechanical systems are almost never uniformly stiff. Different potentials have different stable time step requirements, and even for identical potentials this requirement depends on element

*Corresponding author

size, since finer elements can support higher-energy modes than coarser elements. Any global time integration scheme cannot take advantage of this variability, and instead must integrate the entire system at the globally stiffest time step. Suppose the system can be partitioned into elements such that each force acts entirely within one element. Then *asynchronous variational integrators* (AVIs) [1] generalize VIs by allowing each element to have its own, independent time step. Coarser elements can then be assigned a slower “clock,” and finer elements a faster one. Asynchrony avoids the undesirable situation in which a small number of very fine elements degrade overall performance. AVIs retain all of the properties of variational integrators mentioned above, except for discrete symplecticity. However, AVIs instead preserve an analogous *discrete multisymplectic* form, and it has been shown experimentally that preservation of this form likely induces the same long-time good energy behavior that characterize symplectic integrators [1].

To our knowledge, this work is the first to consider an asynchronous, variational treatment of contact shown to retain multisymplecticity. Rangarajan et al. [11] suggest AVIs for simulating penetration of a soft hyperelastic material by rigid bodies, and propose handling contact by reflecting momentum at the end of any elemental time step during which contact occurred. This method was observed to dissipate energy during contact events; the amount of drift can be controlled by appropriately decreasing the time steps of elements involved in contact. We are also aware that Ryckman and Lew [12] are concurrently investigating extending the AVI framework to incorporate contact response.

The starting point for this approach is the selection of the penalty method as a model for contact [13, 14]. For each pair of elements in the system, a potential is added that is (piecewise) quadratic in the *gap function* measuring the separation distance between the two elements. This potential vanishes when elements are sufficiently far apart, and increases with increasing interpenetration, so that approaching elements feel a force that resists impact. This approach suffers two limitations, however. Firstly, these contact potentials are fundamentally nonlocal phenomena: for every pair of elements that might come into contact during the course of the simulation, a potential coupling the two must be added. As will be shown, the fact that contact potentials cannot be expressed as the integration over the material domain of an energy density depending only on a neighborhood of the domain will present a technical obstruction to the original formulation of AVIs, but fortunately one that can be overcome by a natural generalization.

Secondly, penalty forces have a well-studied performance-robustness tradeoff [15]: adding a half-quadratic potential requires choosing an arbitrary stiffness parameter, and for any stiffness chosen for the penalty potential, two approaching elements will interpenetrate some distance, and in the worst case *tunnel* completely through each other. Moreover, the stable time step of the penalty force decreases as stiffness increases, so choosing a very stiff penalty potential is untenable as a solution to excessive penetration or tunneling. In practice, users of the method must determine an adequate penalty stiffness by iterated tweaking of parameters, until the simulation completes without collision artifacts. An appealing modification of the penalty approach replaces the quadratic potential with a nonlinear *barrier potential* [16] that diverges as the configuration approaches contact. Because the barrier diverges, its stiffness is unbounded, necessitating a time-adaptive time stepping method. This work presents a discrete analogue of the barrier potential—an infinite sequence of *discrete penalty layers*—that in effect enables AVIs to serve as adaptive integrators.

This paper

- extends the construction of AVIs so that a discretization into disjoint elements is no longer necessary, by associating a clock to each force instead of to each element;
- demonstrates that this generalization does not destroy the desirable integration properties guar-

anted by the variational paradigm, most importantly the conservation of the discrete multisymplectic form;

- leverages this extension to equip the AVI framework with a contact model. The proposed barrier method uses a divergent sequence of quadratic potentials that guarantees non-penetration and retains the asynchrony or conservation properties of AVIs;
- presents numerical evidence to support the claim that by retaining the symplectic structure of the smooth system, simulations of thin shells undergoing complex (self-)interactions have demonstrably good long-time energy and momentum behavior;
- describes simple extensions to the contact model to allow for controlled, dissipative phenomena, such as a coefficient of restitution and kinetic friction. Although there is not yet theory explaining the energy behavior of dissipative simulations run under a variational integrator, empirical evidence is presented to show that the proposed method produces smooth, controlled, and qualitatively correct energy decay.

This paper complements the publication [17], which provides a detailed description of the software implementation using *kinetic data structures* [18]. For completeness, Section 6 briefly introduces these concepts.

2. Related Work

The simplest contact models for finite element simulation follow the early analytical work of Hertz [19] in assuming frictionless contact of planar (or nearly planar) surfaces with small strain. In this regime, several approaches have been explored to arrive at a weak formulation of contact; for a high-level survey of these approaches, see for example the overview by Belytschko et al. [20] or Wriggers [21]. The first of these are the use of penalty forces, described for instance by Oden [22] and Kikuchi and Oden [23]. The penalty approach results in a contact force proportional to an arbitrary *penalty stiffness* parameter and to the rate of interpenetration, or in more general formulations to an arbitrary function of rate of interpenetration and interpenetration depth; Belytschko and Neal [15] discuss the choosing of this parameter in Section 8. Recent work by Belytschko et al. [24] uses moving least squares to construct an implicit smooth contact surface, from which the interpenetration distance is evaluated. Peric and Owen [25] describe how to equip penalty forces with a Coulomb friction model.

Seeking to exactly enforce non-penetration along the contact surface leads to generalizations of the method of Lagrange multipliers. Hughes et al. [26] and Nour-Omid and Wriggers [27] provide an overview of this approach in the context of contact response. Such constraint enforcement can be viewed as a penalty force in the limit of infinite stiffness, impossible to attain in practice since the system becomes ill-conditioned. Taylor and Papadopoulos [28] considers persistent contact by extending Newmark to treat jump conditions in kinematic fields, thus reducing undesirable oscillatory modes. However, the effects of these modifications on numerical dissipation and long-time energy behavior is not considered.

The Augmented Lagrangian method blends the penalty and Lagrange multiplier approaches, and combines the advantages of both: unlike for pure penalty forces, convergence to the exact interpenetration constraint does not require taking the penalty stiffness to infinity, and the Lagrange multiplier solve tends to be well-conditioned. Bertsekas [29] gives a mathematical overview of the augmented Lagrangian method, and Wriggers et al. [30] and Simo and Laursen [31] expand on its application to contact problems in finite elements.

Non-smooth contact requires special consideration, since in the non-smooth regime there is no straightforward way of defining a contact normal or penetration distance. Simo et al. [32] discretize the contact surface into segments over which they assume constant contact pressure; this formulation allows them to handle non-node-to-node contact using a perturbed Lagrangian. Kane et al. [33] apply non-smooth analysis to resolve contact constraints between sharp objects. Pandolfi et al. [10] extend the work of Kane et al. by describing a variational model for non-smooth contact with friction. Cirak and West [34] decompose contact resolution into an impenetrability-enforcement and momentum-transfer step, thereby exactly enforcing non-interpenetration while nearly conserving momentum and energy.

Several authors have explored a structure-preserving approach to solving the contact problem. Barth et al. [7] consider an adaptive-step-size algorithm that preserves the time-reversible symmetry of the RATTLE algorithm, and demonstrate an application to an elastic rod interacting with a Lennard-Jones potential. Kane et al. [8] show that the Newmark method, for all parameters, is variational, and construct two two-step dissipative integrators that yield good energy decay. Laursen and Love [9], by taking into account velocity discontinuities that occur at contact interfaces, develop a momentum- and energy-preserving method for simulating frictionless contact. This paper shares with these last approaches the viewpoint that structured integration, with its associated conservation guarantees, is an invaluable tool for accurately simulating dynamic systems with contact.

Although several previous approaches are also adaptive, the algorithm described in this paper is the first structured integrator for contact mechanics that achieves time adaptivity using asynchrony. This novel approach guarantees the robustness of the proposed integrator, without compromising the good properties of structured integration.

3. Variational Integrators

This section presents a background on variational integration and symplectic structure [6, 4, 5].

Let $\gamma(t)$ be a piecewise-regular trajectory through configuration space \mathbf{Q} , and $\dot{\gamma}(t) = \frac{d}{dt}\gamma(t)$ be the configurational velocity at time t . For simplicity, assume that the kinetic energy of the system T depends only on configurational velocity, and that the potential energy V depends only on configurational position, so that the Lagrangian L at time t may be written as

$$L(q, \dot{q}) = T(\dot{q}) - V(q). \tag{1}$$

Then given the configuration of the system q_0 at time t_0 and q_f at t_f , Hamilton's principle [35] states that the trajectory of the system $\gamma(t)$ joining $\gamma(t_0) = q_0$ and $\gamma(t_f) = q_f$ is a stationary point of the action functional

$$S(\gamma) = \int_{t_0}^{t_f} L[\gamma(t), \dot{\gamma}(t)] dt$$

with respect to taking variations $\delta\gamma$ of γ which leave γ fixed at the endpoints t_0, t_f . In other words, γ satisfies

$$dS(\gamma) \cdot \delta\gamma = 0. \tag{2}$$

Integrating by parts, and using that $\delta\gamma$ vanishes at t_0 and t_1 ,

$$dS(\gamma) \cdot \delta\gamma = \int_{t_0}^{t_f} \left(\frac{\partial L}{\partial q}(\gamma, \dot{\gamma}) \cdot \delta\gamma + \frac{\partial L}{\partial \dot{q}}(\gamma, \dot{\gamma}) \cdot \delta\dot{\gamma} \right) dt = \int_{t_0}^{t_f} \left(-\frac{\partial V}{\partial q}(\gamma) - \frac{\partial^2 T}{\partial \dot{q}^2}(\dot{\gamma})\ddot{\gamma} \right) \cdot \delta\gamma dt = 0.$$

Since this equality must hold for all variations $\delta\gamma$ that fix γ 's endpoints,

$$\frac{\partial V}{\partial q}(\gamma) + \frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}}(\dot{\gamma}) \right) = 0, \quad (3)$$

the *Euler-Lagrange equation* of the system. This equation is a second-order ordinary differential equation, and so has a unique solution γ given two initial values $\gamma(t_0)$ and $\dot{\gamma}(t_0)$.

3.1. Symplecticity

The flow $\Theta_s : [\gamma(t), \dot{\gamma}(t)] \mapsto [\gamma(t+s), \dot{\gamma}(t+s)]$ induced by (3) has many structure-preserving properties; in particular it is momentum-preserving, energy-preserving, and symplectic [36]. To derive this last property, for the remainder of this section the space of trajectories is restricted to those that satisfy the Euler-Lagrange equations. For such trajectories, if the requirement that $\delta\gamma$ fix the endpoints of γ is relaxed, then the boundary terms of the integration by parts are no longer 0 and

$$dS(\gamma) \cdot \delta\gamma = \frac{\partial T}{\partial \dot{q}} [\pi_{\dot{q}}(q, \dot{q})] \cdot \delta\gamma \Big|_{t_0}^{t_f}, \quad (4)$$

where $\pi_{\dot{q}}$ is projection onto the second factor.

Since initial conditions (q, \dot{q}) are in bijection with trajectories satisfying the Euler-Lagrange equation, such trajectories γ can be uniquely parametrized by initial conditions $[\gamma(t_0), \dot{\gamma}(t_0)]$. For the remainder of this section variations $\delta\gamma$ are also restricted to *first variations*: those variations in whose direction γ continues to satisfy the Euler-Lagrange equations. These are also parametrized by variations of the initial conditions, $(\delta q, \delta \dot{q})$. For conciseness of notation, the change of variables $\nu(t) = (\gamma(t), \dot{\gamma}(t))$ and $\delta\nu(t) = [\delta\gamma(t), \delta\dot{\gamma}(t)]$ can be used; using this notation the above two facts can be rewritten as $\nu(t) = \Theta_{t-t_0}\nu(t_0)$ and $\delta\nu(t) = \Theta_{t-t_0,*}\delta\nu(t_0)$. The action (1), a functional on trajectories γ , can also be rewritten as a function S_i of the initial conditions,

$$S_i(q, \dot{q}) = \int_0^{t_f-t_0} L[\Theta_t(q, \dot{q})] dt,$$

so that

$$dS(\gamma) \cdot \delta\gamma = dS_i[\nu(t_0)] \cdot \delta\nu(t_0).$$

Substituting all of these expressions into (4),

$$\begin{aligned}
dS_i[\nu(t_0)] \cdot \delta\nu(t_0) &= \left(\frac{\partial T}{\partial \dot{q}} \circ \pi_{\dot{q}} \right) [\Theta_{t-t_0} \nu(t_0)] \cdot \delta\gamma(t) \Big|_{t_0}^{t_f} \\
&= \left(\frac{\partial T}{\partial \dot{q}} \circ \pi_{\dot{q}} \right) [\Theta_{t-t_0} \nu(t_0)] dq \cdot \delta\nu(t) \Big|_{t_0}^{t_f} \\
&= \left(\frac{\partial T}{\partial \dot{q}} \circ \pi_{\dot{q}} \right) [\Theta_{t-t_0} \nu(t_0)] dq \cdot \Theta_{t-t_0}^* \delta\nu(t_0) \Big|_{t_0}^{t_f} \\
&= (\Theta_{t_f-t_0}^* \theta_L - \theta_L)_{\nu(t_0)} \cdot \delta\nu(t_0),
\end{aligned}$$

where θ_L is the one-form $\left(\frac{\partial T}{\partial \dot{q}} \circ \pi_{\dot{q}} \right) dq$. Since dS_i is exact,

$$d^2 S_i = 0 = \Theta_{t_f-t_0}^* d\theta_L - d\theta_L,$$

so since t_0 and t_f are arbitrary, $\Theta_s^* d\theta_L = d\theta_L$ for arbitrary times s , and Θ preserves the so-called *symplectic form* $d\theta_L$.

3.2. Discretization

Discrete mechanics [37, 2, 38, 39, 4, 6] describes a discretization of Hamilton's principle, yielding a numerical integrator that shares many of the structure-preserving properties of the continuous flow Θ_s . Consider a discretization of the trajectory $\gamma : [t_0, t_f] \rightarrow \mathbf{Q}$ by a piecewise linear trajectory interpolating n points $\mathbf{q} = \{q_0, q_1, \dots, q_{n-1}\}$, with $q_0 = \gamma(t_0)$ and $q_{n-1} = \gamma(t_f)$, where the discrete velocity $\dot{q}_{i+1/2}$ on the segment between q_i and q_{i+1} is

$$\dot{q}_{i+1/2} = \frac{q_{i+1} - q_i}{h}, \quad h = \frac{t_f - t_0}{n}.$$

An analogue of (3) in this discrete setting is needed. To that end, a discrete Lagrangian

$$L_d(q_a, q_b) = T \left(\frac{q_b - q_a}{h} \right) - V(q_b) \quad (5)$$

can be formulated, as well as a discrete action

$$S_d(\mathbf{q}) = \sum_{i=0}^{n-2} h L_d(q_i, q_{i+1}). \quad (6)$$

Motivated by (2), a discrete Hamilton's principle can be imposed:

$$dS_d(\mathbf{q}) \cdot \delta\mathbf{q} = 0$$

for all variations $\delta\mathbf{q} = \{\delta q_0, \delta q_1, \dots, \delta q_{n-1}\}$ that fix \mathbf{q} at its endpoints, i.e., with $\delta q_0 = \delta q_{n-1} = 0$. For ease of notation, the kinetic and potential energy terms in (5) can be written to depend on (q_a, q_b) ,

two points of phase space consecutive in time, instead of (q, \dot{q}) :

$$\begin{aligned} T_d(q_a, q_b) &= T\left(\frac{q_b - q_a}{h}\right) & T'_d(q_a, q_b) &= \frac{\partial T}{\partial \dot{q}}\left(\frac{q_b - q_a}{h}\right) \\ V_d(q_a, q_b) &= V(q_b) & V'_d(q_a, q_b) &= \frac{\partial V}{\partial q}(q_b). \end{aligned}$$

Then

$$\begin{aligned} dS_d(\mathbf{q}) \cdot \delta \mathbf{q} &= \sum_{i=0}^{n-2} h (D_1 L_d(q_i, q_{i+1}) \cdot \delta q_i + D_2 L_d(q_i, q_{i+1}) \cdot \delta q_{i+1}) \\ &= \sum_{i=0}^{n-2} h \left(-\frac{1}{h} T'_d(q_i, q_{i+1}) \cdot \delta q_i + \frac{1}{h} T'_d(q_i, q_{i+1}) \cdot \delta q_{i+1} - \frac{\partial V}{\partial q}(q_{i+1}) \cdot \delta q_{i+1} \right) \\ &= T'_d(q_{n-2}, q_{n-1}) \cdot \delta q_{n-1} - T'_d(q_0, q_1) \cdot \delta q_0 - h \frac{\partial V}{\partial q}(q_{n-1}) \cdot \delta q_{n-1} \\ &\quad + \sum_{i=1}^{n-2} \left(T'_d(q_{i-1}, q_i) - T'_d(q_i, q_{i+1}) - h \frac{\partial V}{\partial q}(q_i) \right) \cdot \delta q_i \\ &= \sum_{i=1}^{n-2} \left(T'_d(q_{i-1}, q_i) - T'_d(q_i, q_{i+1}) - h \frac{\partial V}{\partial q}(q_i) \right) \cdot \delta q_i = 0. \end{aligned}$$

Since δq_i is unconstrained for $1 \leq i \leq n-2$,

$$\frac{\partial T}{\partial \dot{q}}(\dot{q}_{i+1/2}) - \frac{\partial T}{\partial \dot{q}}(\dot{q}_{i-1/2}) = -h \frac{\partial V}{\partial q}(q_i), \quad i = 1, \dots, n-2, \quad (7)$$

the *discrete Euler-Lagrange equations* of the system.

Unlike in the continuous settings, the discrete Euler-Lagrange equations do not always have a unique solution given initial values q_0 and q_1 . Therefore in all that follows it is assumed that T_d and V_d are of a form so that (7) gives a unique q_{i+1} given q_i and q_{i-1} —this assumption always holds, for instance, in the typical case where T_d is quadratic in \dot{q} . Then the discrete Euler-Lagrange equations give a well-defined discrete flow

$$F : (q_{i-1}, q_i) \mapsto (q_i, q_{i+1}),$$

which recovers the entire trajectory from initial conditions, in perfect analogy to the continuous setting.

3.3. Symplecticity of the Discrete Flow

By analogy to the continuous setting, it is desired that F preserve a symplectic form, just as $d\theta_L$ is preserved by Θ . As in the continuous setting, trajectories \mathbf{q} are restricted to those that satisfy the discrete Euler-Lagrange equations, and variations to first variations (and the condition that these variations vanish at the endpoints is lifted), yielding

$$dS_d(\mathbf{q}) \cdot \delta \mathbf{q} = T'_d(q_{n-2}, q_{n-1}) \cdot \delta q_{n-1} - T'_d(q_0, q_1) \cdot \delta q_0 - h \frac{\partial V}{\partial q}(q_{n-1}) \cdot \delta q_{n-1}.$$

F^k denotes the discrete flow F composed with itself k times, or k “steps” of F . Again, all \mathbf{q} satisfying (7) can be parametrized by initial conditions $\nu_0 = (q_0, q_1)$, and first variations by $\delta\nu_0 = (\delta q_0, \delta q_1)$, so that the discrete action can be rewritten as

$$S_{id}(\nu_0) = \sum_{i=0}^{n-2} hL_d(F^i\nu_0).$$

Putting together all of the pieces,

$$\begin{aligned} dS_{id}(\nu_0) \cdot \delta\nu_0 &= dS_d(\mathbf{q}) \cdot \delta\mathbf{q} \\ &= T'_d(q_{n-2}, q_{n-1}) \cdot \delta q_{n-1} - T'_d(q_0, q_1) \cdot \delta q_0 - h \frac{\partial V}{\partial q}(q_{n-1}) \cdot \delta q_{n-1} \\ &= \left(T'_d(q_a, q_b) - h \frac{\partial V}{\partial q}(q_b) \right) dq_b \cdot (\delta q_{n-2}, \delta q_{n-1}) \Big|_{q_a=q_{n-2}, q_b=q_{n-1}} \\ &= [T'_d(F^{n-2}\nu_0) - hV'(F^{n-2}\nu_0)] dq_b \cdot F^{n-2} \delta\nu_0 - T'_d(\nu_0) dq_a \cdot \delta\nu_0 \\ &= \theta_{F^{n-2}\nu_0}^+ \cdot F^{n-2} \delta\nu_0 + \theta_{\nu_0}^- \cdot \delta\nu_0 \\ &= \left(F^{n-2} \theta^+ \right)_{\nu_0} \cdot \delta\nu_0 + \theta_{\nu_0}^- \cdot \delta\nu_0. \end{aligned}$$

for the indicated two-forms θ^+ and θ^- . Since $d(hL_d) = \theta^+ + \theta^-$, $d^2(hL_d) = 0 = d\theta^+ + d\theta^-$. Moreover the initial conditions ν_0 are arbitrary, hence

$$d^2 S_{id} = 0 = (F^{n-2})^* d\theta^+ + d\theta^- = -(F^{n-2})^* d\theta^- + d\theta^-,$$

so

$$d\theta^- = (F^{n-2})^* d\theta^-.$$

Since n is arbitrary, the discrete flow F preserves the symplectic form $d\theta^-$. Using backwards error analysis, it can be shown that this geometric property guarantees that integrating with F introduces no energy drift for a number of steps exponential in h [6], a highly desirable property when simulating molecular dynamic or other Hamiltonian systems whose qualitative behavior is substantially affected by errors in energy.

4. Asynchronous Variational Integrators

In Section 3.2 an action functional (6) was formulated as the integration of a single discrete Lagrangian over a single time step size h . Such a construction is cumbersome when modeling multiple potentials of varying stiffnesses acting on different parts of the system: to prevent instability one must integrate the entire system at the resolution of the stiffest force. Asynchronous variational integrators (AVIs), introduced by Lew et al. [1], are a family of numerical integrators, derived from a discrete Hamilton’s principle, that support integrating potentials at different time steps. Their formulation assumes a spatial partition, with each potential depending only on the configuration of a single element; in this exposition, the general arguments set forth by Lew et al. are followed, but the notation and derivation departs from their work as necessary to support potentials with arbitrary, possibly non-disjoint spatial stencil.

Let $\{V^i\}$ be potentials with time steps h^i . Each potential V^i is concerned with certain moments in time—namely, integer multiples of h^i —and these moments are inconsistent across triangles. Time is therefore subdivided in a way compatible with all triangles: for a τ -length interval of time, the set $\Xi(\tau)$ is defined by

$$\Xi(\tau) = \bigcup_{V^i} \bigcup_{j=0}^{\lfloor \tau/h^i \rfloor} jh^i.$$

That is, $\Xi(\tau)$ is the set of all integer multiples less than τ of all time steps. Ξ can be ordered, and in particular let $\xi(i)$ be the $(i+1)$ -st least element of Ξ . For ease of notation, also let $\omega^i(j) = \xi^{-1}(jh^i)$; that is, ω converts the j th timestep of potential i into a global time.

If n is the cardinality of Ξ , a trajectory of duration τ is then discretized by linearly interpolating intermediate configurations q_0, q_1, \dots, q_{n-1} , where q_i is the configuration of the system at time $\xi(i)$. Velocity is discretized as $\dot{q}_{k+1/2} = \frac{q_{k+1} - q_k}{\xi(k+1) - \xi(k)}$ on the segment of the trajectory between q_k and q_{k+1} . A global action functional of these trajectories is needed, and can be constructed in the natural way:

$$S_g(\mathbf{q}) = \sum_{j=0}^{n-2} [\xi(j+1) - \xi(j)] T_d[q_j, q_{j+1}, \xi(j), \xi(j+1)] - \sum_{V^i} \sum_{j=1}^{\lfloor \tau/h^i \rfloor} h^i V^i(q_{\omega^i(j)}),$$

where, for $T(\dot{q})$ the kinetic energy of the entire configuration, $T_d(q_a, q_b, t_a, t_b) = T\left(\frac{q_b - q_a}{t_b - t_a}\right)$. For use in the following, also let $T'_d(q_a, q_b, t_a, t_b) = \frac{\partial T}{\partial \dot{q}}\left(\frac{q_b - q_a}{t_b - t_a}\right)$.

No attempt has been made to define a Lagrangian pairing the kinetic and potential energy terms; it will be seen that an action defined in this way still leads to a multisymplectic numeric integrator. To this end, Hamilton's principle $dS_g(\mathbf{q}) \cdot \delta\mathbf{q} = 0$ is imposed for variations $\delta\mathbf{q} = \{\delta q_0, \dots, \delta q_{n-1}\}$ with $\delta q_0 = \delta q_{n-1} = 0$. Then S_g can be rewritten as

$$S_g(\mathbf{q}) = \sum_{j=0}^{n-2} [\xi(j+1) - \xi(j)] T_d[q_j, q_{j+1}, \xi(j), \xi(j+1)] - \sum_{j=1}^{n-1} \sum_{h^i | \xi(j)} h^i V^i(q_j), \quad (8)$$

where the notation $h^i|\xi(j)$ is abused to mean “all indices i for which h^i evenly divides $\xi(j)$,” so that

$$\begin{aligned}
dS_g(\mathbf{q}) \cdot \delta\mathbf{q} &= \sum_{j=0}^{n-2} T'_d [q_j, q_{j+1}, \xi(j), \xi(j+1)] \cdot (\delta q_{j+1} - \delta q_j) - \sum_{j=1}^{n-1} \sum_{h^i|\xi(j)} h^i \frac{\partial V_i}{\partial q}(q_j) \cdot \delta q_j \\
&= T'_d [q_{n-2}, q_{n-1}, \xi(n-2), \xi(n-1)] \cdot \delta q_{n-1} - T'_d [q_0, q_1, \xi(0), \xi(1)] \cdot \delta q_0 \\
&\quad - \sum_{h^i|\xi(n-1)} h^i \frac{\partial V_i}{\partial q}(q_{n-1}) \cdot \delta q_{n-1} \\
&\quad + \sum_{j=1}^{n-2} \left(T'_d [q_{j-1}, q_j, \xi(j-1), \xi(j)] - T'_d [q_j, q_{j+1}, \xi(j), \xi(j+1)] - \sum_{h^i|\xi(j)} h^i \frac{\partial V_i}{\partial q}(q_j) \right) \cdot \delta q_j \\
&= \sum_{j=1}^{n-2} \left(T'_d [q_{j-1}, q_j, \xi(j-1), \xi(j)] - T'_d [q_j, q_{j+1}, \xi(j), \xi(j+1)] - \sum_{h^i|\xi(j)} h^i \frac{\partial V_i}{\partial q}(q_j) \right) \cdot \delta q_j.
\end{aligned}$$

The Euler-Lagrange equations are then

$$\frac{\partial T}{\partial \dot{q}}(\dot{q}_{k+1/2}) - \frac{\partial T}{\partial \dot{q}}(\dot{q}_{k-1/2}) = - \sum_{h^i|\xi(k)} h^i \frac{\partial V_i}{\partial q^i}(q_k), \quad (9)$$

These equations are similar to those derived for synchronous variational integrators (7), except that only a subset of potentials V_d^i contribute during each time step. As in the synchronous case, if, as is typical, $T_d(\dot{q})$ is quadratic in \dot{q} , the system (9) gives rise to an explicit numerical integrator that is particularly easy to implement in practice. Algorithm 1 gives pseudocode for such integration when $T_d = \dot{q}^T \mathbf{M} \dot{q}$ for a mass matrix \mathbf{M} ; Lew et al. [36] discuss the algorithm in greater detail.

4.1. Multisymplecticity

The right hand side of (9) depends on $\xi(k)$, and so unlike (7), the Euler-Lagrange equations for AVIs are time dependent, and do not give rise to a uniform update rule $F(q_{i-1}, q_i) \mapsto (q_i, q_{i+1})$. Instead, consider the total, time-dependent flow $\hat{F}^k(q_0, q_i) \mapsto (q_k, q_{k+1})$. Once again, trajectories satisfying (9) are parametrized by $\nu_0 = (q_0, q_1)$, and first variations by $\delta\nu_0 = (\delta q_0, \delta q_1)$. By restricting to such trajectories and variations, the action (8) can be rewritten as

$$S_{\text{iAVI}} = \sum_{j=0}^{n-2} [\xi(j+1) - \xi(j)] T_d \left(\hat{F}^j(\nu_0), \xi(j), \xi(j+1) \right) - \sum_{V^i} \sum_{j=0}^{\lfloor \tau/h^i \rfloor - 1} h^i V_d^i(\hat{F}^{\omega^i(j+1)}(\nu_0)).$$

Algorithm 1 An algorithm for integrating the trajectory given by the AVI Euler-Lagrange equations (9) adapted from Lew et al. [36]

Let events be (potential, time step, time) triplets $E = (V, h, t)$.
Denote by q_V the configuration subspace on which V depends.
Let PQ be a priority queue of events, sorted by event times $E.t$.
 $T_g \leftarrow 0$ $\{T_g$ maintains the value of the simulation clock $\}$
 $q \leftarrow q_0$ $\{\text{Set up initial conditions}\}$
 $\dot{q} \leftarrow \dot{q}_0$
for all V_i **do**
 $E_i \leftarrow (V_i, h^i, h^i)$ $\{\text{Add all potentials to the queue as events}\}$
 $PQ.\text{push}(E_i)$
end for
loop
 $(V, h, t) \leftarrow PQ.\text{pop}$
 $q \leftarrow q + (t - T_g)\dot{q}$
 $\dot{q}_V \leftarrow \dot{q}_V - hM_V^{-1} \frac{\partial V}{\partial q_V}$ $\{\text{Update only those elements affected by this event.}\}$
 $PQ.\text{push}(V, h, t + h)$ $\{\text{Return the event to the queue, with a new, later time}\}$
 $T_g \leftarrow t$ $\{\text{Update the simulation clock}\}$
end loop

Then, for $V_d^{i'}(q_a, q_b) = \frac{\partial V^i}{\partial q}(q)$,

$$\begin{aligned}
dS_{\text{iAVI}}(\nu) \cdot \delta\nu &= dS_g(\mathbf{q}) \cdot \delta\mathbf{q} \\
&= T'_d[q_{n-2}, q_{n-1}, \xi(n-2), \xi(n-1)] \cdot \delta q_{n-1} - T'_d[q_0, q_1, \xi(0), \xi(1)] \cdot \delta q_0 \\
&\quad - \sum_{V^i} \sum_{h^i|\xi(n-1)} h^i \frac{\partial V^i}{\partial q^i}(q_{n-1}) \cdot \delta q_{n-1} \\
&= T'_d[\hat{F}^{n-2}(\nu_0), \xi(n-2), \xi(n-1)] \cdot \delta q_{n-1} - T'_d[\nu_0, \xi(0), \xi(1)] \cdot \delta q_0 \\
&\quad - \sum_{V^i} \sum_{h^i|\xi(n-1)} h^i V_d^{i'}[\hat{F}^{n-2}(\nu_0)] \cdot \delta q_{n-1} \\
&= \theta_{\nu_0}^- \cdot \delta\nu_0 + \theta_{\hat{F}^{n-2}\nu_0}^+ \cdot \hat{F}^{n-2*} \delta\nu_0 \\
&= (\theta^- + \hat{F}^{n-2*}\theta^+)_{\nu_0} \cdot \delta\nu_0
\end{aligned}$$

for one-forms θ^- and θ^+ . Once again

$$0 = d^2 S_{\text{iAVI}} = d\theta^- + \hat{F}^{n-2*} d\theta^+, \quad (10)$$

but unlike when the action was a sum of Lagrangians, from the *multisymplectic form formula* (10) there is no way of relating $d\theta^-$ to $d\theta^+$, and thus discrete symplectic structure preservation is not recovered. Nevertheless, Lew et al. [1] conjecture that this multisymplectic structure leads to the good energy behavior observed for AVIs.

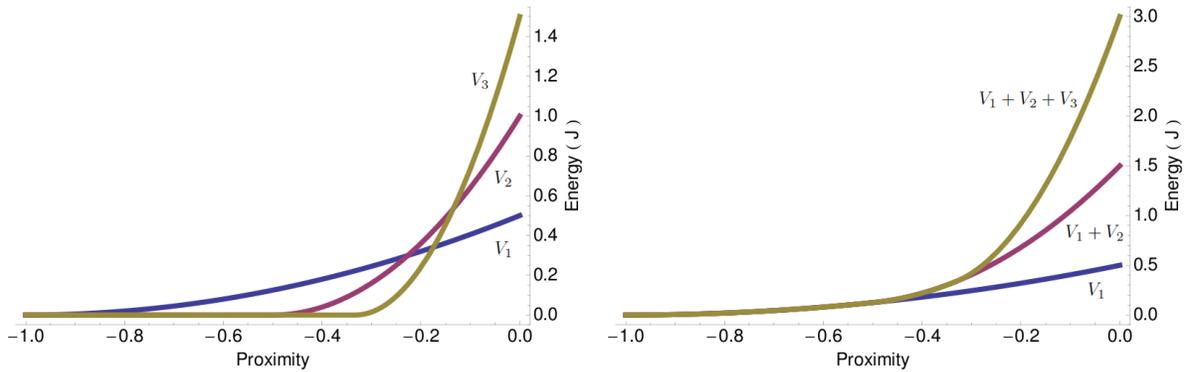


Figure 1: Plots of the potential energy of the first three layers as a function of gap function g (left), and a plot of the total potential energy contributed by all layers $\leq n$ for $n = 1, 2, 3$ (right). Notice the potential energy diverges as separation distance approaches 0, guaranteeing that collision response is robust.

5. Discrete Penalty Layers

The above reformulation of AVIs can be leveraged to resolve collisions with guaranteed perfect robustness, and via momentum-symplectic integration, so that the energy behavior of the system as a whole remains good. Consider a standard penalty force approach, which for every two elements A, B and surface thickness η defines the gap function

$$g_\eta(q) = \inf_{a \in A, b \in B} \|a - b\| - 2\eta$$

measuring the proximity of A to B .

The penalty potential is then defined as

$$V(q) = \begin{cases} 0 & g_\eta(q) > 0 \\ k g_\eta(q)^2 & g_\eta(q) \leq 0, \end{cases}$$

where k is a user-specified stiffness. As previously discussed, V alone does not robustly prevent interpenetrations: the potential can be viewed as placing a spring between the approaching elements, and for sufficiently large relative momentum in the normal direction, the spring will fully compress, then fail. However, consider placing an infinite family of potentials V_l , $l = 1, 2, \dots$, between the primitives, where

$$V_l(q) = \begin{cases} 0 & g_{\eta/l}(q) > 0 \\ l^3 k g_{\eta/l}^2 & g_{\eta/l}(q) \leq 0. \end{cases}$$

The region $\frac{\eta}{n+1} \leq d(q) \leq \frac{\eta}{n}$, where exactly n of the potentials are active, is called the n -th *discrete penalty layer*. Figure 1 shows a plot of the potential energy of the first few potentials for the case $\eta = k = 1$, as well as the cumulative potential energy of all of the potentials.

The total potential energy of the springs when fully compressed is

$$\sum_{l=1}^{\infty} l^3 k_4 \left(\frac{\eta}{l}\right)^2 = 4k\eta^2 \sum_{l=1}^{\infty} l,$$

which diverges. The infinite array of potentials is guaranteed to stop all collisions. This guarantee in no way depends on the chosen stiffness k : although performance and error will vary with the choice of stiffness, unlike for penalty forces the stiffness does not affect the guarantee. The method is always guaranteed to be robust.

There is one obstruction to implementing this scheme in practice: integrating the l -th spring stably and with good energy behavior requires a time step proportional to $\frac{1}{l^{3/2}}$, which vanishes as $l \rightarrow \infty$. Using a traditional integrator, one could decide ahead of time to only simulate the first few springs—but then the guarantee that no penetrations will occur is lost, and the simulation must be run at a prohibitively small time step. AVIs, with the above modifications, and a bit of extra bookkeeping, are a first step towards alleviating the problem, by allowing the user to assign each spring its own time step. This bookkeeping is now described, in terms of modifications to the basic Algorithm 1.

6. The Asynchronous Algorithm

AVIs allow each penalty layer to be assigned a different time step, so that less stiff (l small) layers can take large time steps regardless of the presence of the stiffer layers. However, it is still not possible as a practical matter to integrate the system, since arbitrarily large l would need arbitrarily small time steps, and the global time in Algorithm 1 would never advance. The following observation surmounts this obstacle: at any time during a well-posed simulation, the number of layers that are exerting a non-zero force, or that are *active*, is finite. More precisely, a simulation is well-posed if its total energy over time is bounded—that is, if the simulation begins in a non-penetrating state; all prescribed, infinite-mass bodies are stationary; and only a finite amount of energy is added over time in the form of external forcing. Inactive penalty potentials can be ignored by Algorithm 1 entirely, since they do not change configurational velocity, and the position integration that would take place during the handling of an inactive potential can just as well be done by the following event. Therefore the simulation would be guaranteed to never stop making progress if there is a lower bound for the amount of global time T_g that elapses with the processing of any event. Such a lower bound exists if there is a way to detect which penalty potentials are active or inactive at all times and remove all inactive events from the priority queue PQ .

Suppose that at the start of the simulation, all penalty layers are inactive. Thus no penalty layer events are needed on the queue. For each pair of simulation elements, the time t_a that the first penalty layer would become active (assuming all elements continue along the trajectory described by their initial velocities) can be calculated, and the corresponding event added to the queue at that time. Such an approach suffers from two problems, however. Firstly, solving for the time when the gap function will be zero is easy in some cases, such as if the elements are two spheres or two planes, but can involve expensive root solves in others, such as if the elements are two non-rigid triangular elements of a thin shell simulation. Secondly, the times computed are fragile: should any event alter the velocity of one of the elements (such as a material force, or gravity, or another penalty force if one of the elements collides with a third party) the activation time is no longer valid and must be recomputed.

Instead of an exact time, only a conservative guarantee, or *certificate* [18], that the first penalty layer will not be active before some time t_c (where necessarily $t_c \leq t_a$) is truly needed. For example, one certificate is the existence of an 2η -thick *planar slab* S that separates the two elements up until time t_c , where η is the thickness of the first penalty layer. For an m -dimensional configuration space, such a planar slab is understood to be an extrusion of an $(m-1)$ -dimensional affine subspace. Concretely, let w be a unit vector in \mathbb{R}^m , w_i be $m-1$ linearly independent vectors in \mathbb{R}^m orthogonal to w , and p a point in \mathbb{R}^m . Then the slab $S_{w,p}$ is the set

$$S_{w,p} = \left\{ p + \alpha w + \sum_i \beta_i w_i \mid -\eta \leq \alpha \leq \eta, \beta_i \in \mathbb{R} \right\}.$$

If such a slab separates the two elements, the first penalty layer cannot become active before t_c . This certificate can be placed as an event on the queue, with time t_c . The certificate might then suffer several fates: [17]

- An event modifies the velocity of one of the elements before time t_c . The certificate placed on the queue is then no longer valid until time t_c , but instead until a new time t'_c which may be sooner or later than t_c . The algorithm must thus *reschedule* the certificate, by removing its event from the queue, and reinserting it at the appropriate new time.
- The certificate event is popped from the queue without incident, but it is possible and convenient to find a new separating slab that guarantees the penalty layer does not activate before time $t'_c > t_c$. This new certificate can then be pushed on the queue for time t'_c .
- The certificate event is popped from the queue without incident, but finding a new slab is impossible, costly, or a slab can be found, but the new time t'_c is judged heuristically to be too near t_c . The first penalty layer may then be activated early: doing so affects the efficiency, but not the correctness, of the simulation. Simultaneously, the algorithm searches for an η -thick separating slab to serve as a certificate that layer two is not yet active, and the whole process described above is repeated.

Detecting when a penalty layer event becomes inactive, and should be removed from the queue, is much simpler than detecting layer activation: whenever a penalty force for layer n is integrated, the algorithm simply checks if the force applied was 0. If so, and if the two elements in question are separating, layer n is now inactive: it is not pushed back onto the queue (and instead a separating slab of thickness η/n is sought.)

It is very important to note that when an event becomes active and is added back into the event priority queue, it is done so at a time that is *an integer multiple of its timestep from its last time of integration*. That is, those times when integration would do nothing have been optimized away, but the potential’s “integration clock” has not been tampered with or realigned, since every potential having a fixed-size time step was fundamental to the proof that asynchronous variational integration is multisymplectic. The spring-on-a-plane example described below underlines the danger of failing to maintain such a fixed time step.

For an event E , denote all simulation elements on which E depends the *support* of V . Denote all simulation elements whose velocities are modified by E the *stencil* of E . For force integration events, there is no distinction between stencil and support. Certificates have a support, but no stencil. Algorithm 2 uses this terminology to incorporate the above into the AVI algorithm.

In Algorithm 2 and its accompanying subalgorithms, the behavior of the functions FindCertificate and Schedule will depend on the type of certificate chosen. FindCertificate returns a new certificate

Algorithm 2 Proposed algorithm for asynchronous contact resolution.

Let force events be (potential, time step, time) triplets $E = (V, h, t)$.
Let PQ be a priority queue of events, sorted by event times $E.t$.
 $T_g \leftarrow 0$ $\{T_g$ maintains the value of the simulation clock $\}$
 $q \leftarrow q_0$ $\{\text{Set up initial conditions}\}$
 $\dot{q} \leftarrow \dot{q}_0$
Push non-penalty (e.g. material) events on the queue
for all pairs of elements K_1, K_2 **do**
 $E \leftarrow \text{FindCertificate}(K_1, K_2)$
 $PQ.\text{push}(E)$
end for
loop
 $E \leftarrow PQ.\text{pop}$
 $q \leftarrow q + (E.t - T_g)\dot{q}$
 if E is a force event **then**
 $\text{handleForceEvent}(PQ, E)$
 else
 $\text{handleCertificateEvent}(PQ, E)$
 end if
 $T_g \leftarrow E.t$ $\{\text{Update the simulation clock}\}$
end loop

for a given pair of elements, if possible and practical, and `Schedule` computes the time a certificate becomes invalid, as described in the paragraphs above. For thin shell simulation, where all simulation elements are convex triangles, edges, and vertices, separating slabs serve as ideal certificates, since it is cheap to compute `Schedule`, in this case by calculating element-plane intersection times. Although any choice of certificate, and heuristic for when to abort searching for a new certificate, preserves the correctness of the algorithm, the progress property described in the first paragraph of this section relies on the certificates efficiently weeding out inactive events so that some certificate is found before all (infinitely many) layers for a pair of elements are activated. No problems have been observed using separating slabs for thin-shell simulations, but different certificates may be needed, e.g., for concave rigid bodies.

6.1. Further Optimizations

The technique explored in the previous section, of finding a sequence of conservative certificates guaranteeing that some property holds, instead of calculating an exact time when that property stops holding, is the central idea behind a wide class of algorithms known as *Kinetic Data Structures* (KDSs) [18]. In the case described above, the property was inactivity of a given penalty layer. KDSs are particularly well-suited for an asynchronous approach, since certificate expiration times may not all align to some convenient simulation clock, and the required rescheduling of certificates/searching for new certificates can reuse the priority queue data structure already needed for force integration events. To improve the efficiency of the implementation used to create the examples below, several more KDSs in addition to the separating slabs discussed above were implemented: a bounding volume hierarchy [40] was used to take advantage of the fact that spatially distant elements are unlikely to collide, separation lists [41] to optimize the bookkeeping of this hierarchy, and a novel KDS was devised to

Algorithm 3 handleForceEvent

Require: Priority queue of events PQ and force event E that needs processing
{Processing a force event E is a three-step process: integrating the force, rescheduling all events whose support depends on E 's stencil, and lastly, resceduling E itself.}
for all i in $\text{Stencil}(E)$ **do**
 $\dot{q}_i \leftarrow \dot{q}_i - (E.h)M_i^{-1} \frac{\partial E.V}{\partial q_i}$ {Update only those elements affected by this event.}
end for
{Reschedule all events whose support depends on E 's stencil}
for all Certificate events E' with $\text{Stencil}(E) \cap \text{Support}(E') \neq \emptyset$ **do**
 $PQ.\text{remove}(E')$
 $\text{Schedule}(E')$
 $PQ.\text{push}(E')$
end for
{If E was a penalty force event, it exerted 0 force, and the two primitives in question are separating, then we no longer need it}
if E is a penalty force event and $\frac{\partial E.V}{\partial q_i} = 0$ **then**
 if $E.V.K1$ and $E.V.K2$ have positive relative velocity (are separating) **then**
 return
 end if
 if $\text{addCertificate}(E.V.K1, E.V.K2)$ **then**
 return
 end if
end if
{Otherwise, reschedule E itself}
 $PQ.\text{push}(V, h, t + h)$

Algorithm 4 handleCertificateEvent

Require: Priority queue of events PQ and certificate event E that needs rescheduling
if not $\text{addCertificate}(E.K1, E.K2)$ **then**
 {Finding a new certificate failed. We must thus activate a penalty force, one layer deeper than the deepest currently active penalty force event.}
 $CurLayer \leftarrow \max_{\{\text{penalty events } E' \text{ on queue for } E.K1 \text{ and } E.K2\}} E'.layer$
 $E' \leftarrow \text{new PenaltyForceEvent}(E.K1, E.K2, CurLayer + 1)$
 $PQ.\text{push}(E')$ {Push the appropriate penalty force event on the queue}
end if

Algorithm 5 addCertificate

Require: Priority queue of events PQ , and two elements $K1$ and $K2$

{Attempts to find a certificate for the collision of $K1$ against $K2$ and add it to the queue. Returns true if one was found.}

$E' \leftarrow \text{FindCertificate}(K1, K2)$

if E' was successfully found **then**

$PQ.\text{push}(\text{FindCertificate}(K1, K2))$

return true

end if

return false

leverage the observation that high-frequency, low amplitude oscillations in velocity do not significantly change a separation slab's expiration time, so that rescheduling is in many cases unnecessary. All of the improvements are described in greater detail in [17].

7. Dissipation

The framework, as described so far, gives near-perfect long-time energy conservation. In the real world, however, many dissipative phenomena are observed — for instance friction, spring damping, and non-unit coefficients of restitution during collisions. Several simple modifications can be made to the proposed method to take such dissipation into account. Qualitatively, these have performed well in practice: energy seems to behave well over long times for dissipative systems analogously to the near-conservation observed for Hamiltonian systems, but a theoretical understanding of this good behavior remains future work.

7.1. Coefficient of Restitution

It is often desirable to simulate semi-elastic or inelastic collisions. A simple modification to the potential V_l allows the use of arbitrary coefficients of restitution e :

$$V_l(q) = \begin{cases} 0 & g_{\eta/l}(q) > 0 \\ l^3 k s g_{\eta/l}(q)^2 & g_{\eta/l}(q) \leq 0, \end{cases}$$

where s is e if the primitives are separating, 1 otherwise. The penalty layers exert their full force during compression, then weaken according to the coefficient of restitution during decompression.

This approach, while simple, does have a limitation in the inelastic limit $e = 0$: due to error introduced by numerical integration, two colliding primitives may have non-zero, though small, post-response normal relative velocity. The magnitude of this velocity is at most $k\eta/l$, so it can be limited by choosing a small enough base stiffness k .

7.2. Friction

The Coulomb friction model is a simple approximation to kinetic friction: at a point of contact between two bodies, the Coulomb force has magnitude $\mu|F_n|$, where μ is a coefficient of friction and F_n is the normal force at the contact points, and has direction opposite the relative tangential motion of the contact points.

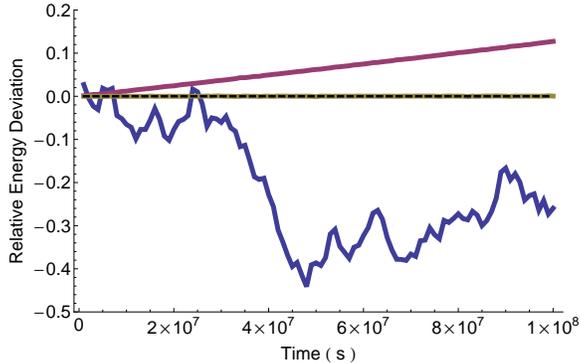


Figure 2: The relative error in energy of a spring bouncing on a plane when *a*) the system is integrated by mixing variational material force integration with impulse-based collision response (blue), *b*) the alignment of the integration clocks is not respected (maroon), *c*) using the proposed method (brown).

Whenever an impulse is applied during integration of a penalty layer, a corresponding frictional impulse can also be applied. Just as increasingly stiff penalty forces are applied for contact forces, friction forces are increasingly applied (equal to $\mu|F_n|$) to correctly halt high-speed tangential motion. Notice that these friction forces, like the material and contact penalty forces, are applied asynchronously: every layer applies friction independently at its own time step.

This simple, asynchronous formulation of friction fits very naturally into the framework of AVIs. Unfortunately, it is unsuitable for simulations featuring static friction, such as a block of wood resting on an inclined plane. The above formulation, with friction applied piecemeal during penalty integration, is reactive instead of proactive, and in simulations of this type the block of wood has been observed “creeping” down the incline no matter how high a coefficient of restitution is chosen. A more comprehensive model of friction compatible with the AVI framework, which correctly handles static friction, remains future work.

8. Results

8.1. Spring on a plane

As a simplest didactic demonstration of the proposed method, three experiments were conducted. A vertical spring of unit rest length, stiffness, and endpoint masses began each of the three simulations stationary a unit height above a fixed horizontal plane. The springs fell under a gravitational force of strength $1m/s^2$, with impact handled in one of three different ways:

In the first experiment, gravity and the stretching force were integrated synchronously, and an instantaneous impulse was applied whenever the bottom of the spring touched the plane. Figure 2 shows error in energy over time when using this method. Energy in this experiment, far from being conserved, took a random walk.

In the second experiment, all forces were integrated asynchronously using the proposed method. The thickness η was chosen to be 0.1, and the penalty base stiffness k , 1000. Energy in this case was well-conserved over long time: although energy experiences high-frequency, low-amplitude oscillations, there was no drift.

The importance of respecting the integrity of each potential’s integration clock is highlighted in the third experiment. Instead of adding a force event onto the priority queue at an integer multiple of

its time step, the events are added at the precise moment when each layer becomes active. As can be seen from the resulting plot of energy error (Figure 2), energy is no longer well-conserved, but instead seems to increase monotonically over time.

8.2. Balls of Particles

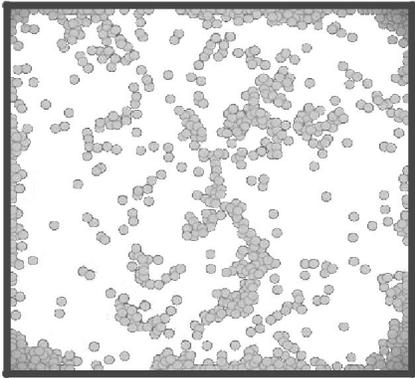


Figure 3: A rigid box containing 900 spheres with random initial velocity, several minutes after the start of the simulation.

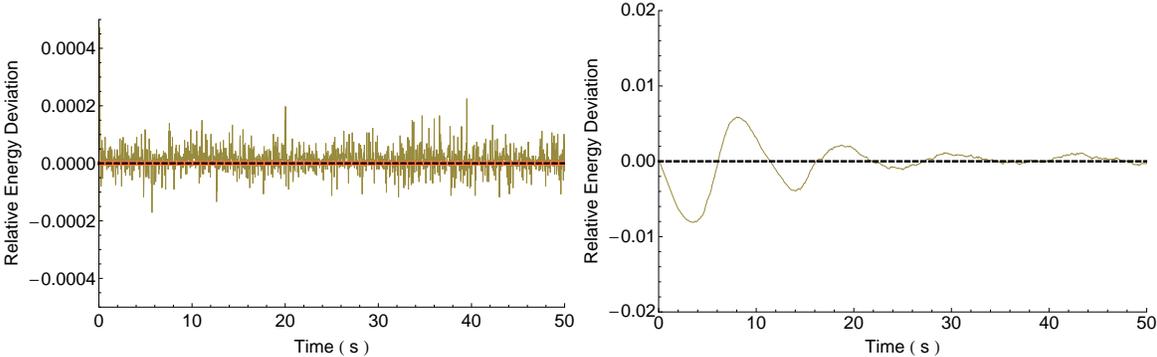


Figure 4: Left: The relative error in measured energy (brown) and momentum (orange), as compared to the same quantities at the start of the simulation, for the box of spheres. Right: The relative error of energy of the box, with gravity added, over time.

As an example that involves more collisions, consider a fixed $3\text{ m} \times 3\text{ m}$ square box. Inside this box 900 spheres of radius 10 cm were uniformly distributed, each of which was given a random velocity of magnitude between 0 and 10 m/s. Figure 3 depicts this box after several minutes have elapsed. Energy error over time is plotted in Figure 4 (left), and it is again almost perfectly conserved. The same plot also shows the error in total momentum of the box over time, and it is exactly zero, as expected since a multisymplectic integrator is used. Gravity (9.8 m/s^2) was added to the box and again the relative error of energy was plotted over time (Figure 4, right), Good behavior of the energy error was still observed.

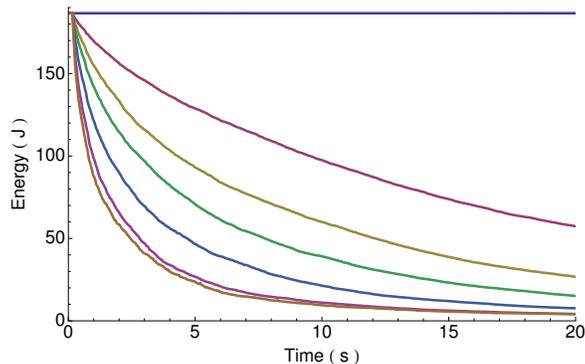


Figure 5: The energy of the box of 900 spheres under different coefficients of restitution: from top to bottom, 1.0, 0.9, 0.8, 0.7, 0.5, 0.2, 0.0.

As a test of controllable dissipation by using a coefficient of restitution, the box with gravity was resimulated several times using different coefficients of restitution. Figure 5 shows the resulting energy plots. For any chosen coefficient of restitution, the non-conservative energy behavior is qualitatively as one would expect.

8.3. Sphere-Plate Impact

The impact experiment of a spherical shell against a thin plate, as described in Cirak and West’s article on Decomposition Contact Response (DCR) [34], was reproduced using the proposed framework. A sphere of radius 12.5 cm approaches a plate of radius 35 cm with relative velocity 100 m/s. Both the sphere and the plate have thickness 0.35 cm. The time steps of the material forces (stretching and bending) are 10^{-7} s (the same as those chosen by Cirak and West.)

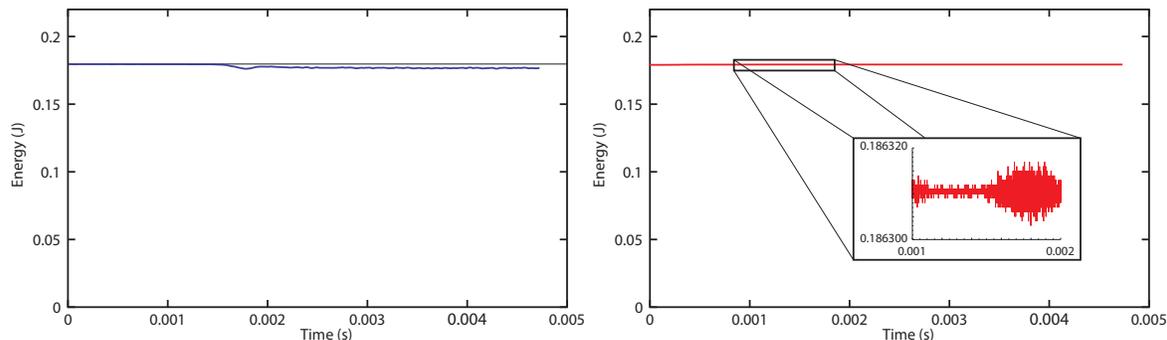


Figure 6: Total energy over time of a thin sphere colliding against a thin plate, simulated using the proposed contact response method (right) compared to data provided for decomposition contact response [34] (left).

Figure 6 compares energy over time when this simulation is run using both the proposed method and DCR. Using the former there is no noticeable long-term drift; closely examining the energy data reveals the high-frequency, low-amplitude, qualitatively-negligible oscillations characteristic of symplectic integration. The latter introduces noticeable artificial energy decay.

8.4. Large-scale Three-dimensional Examples

Harmon et al. [17] describe a series of optimizations that improve the efficiency of Algorithm 2. These optimizations were incorporated to form our Asynchronous Contact Mechanics (ACM) code. This code continues to yield qualitatively good results when scaled to additional large-scale problems.

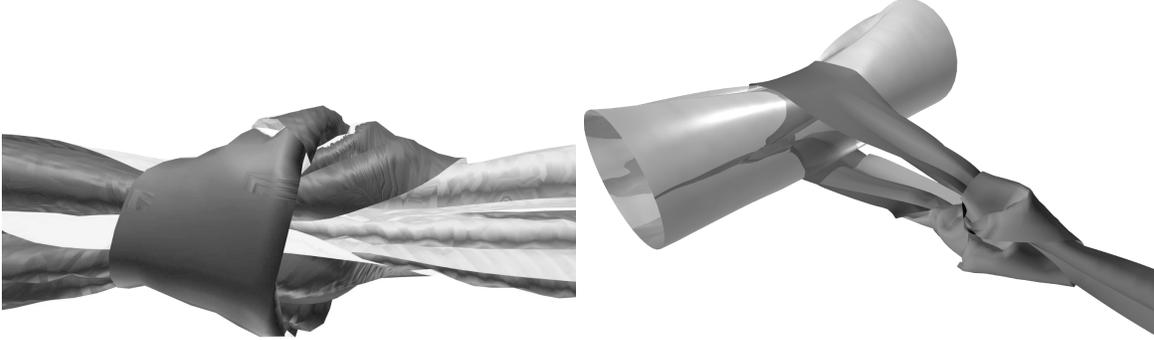


Figure 7: Simulated tying of a cloth reef knot (left) and bowline knot (right).

Two thin rectangular $27\text{ cm} \times 2\text{ cm}$ ribbons were modeled as thin shells of 5321 vertices, subject to constant-strain triangle stretching forces [42] (stiffness 750) and discrete shell bending forces formulated by Grinspun et al. [43] (stiffness 0.05). These ribbons were positioned into a loose reef knot by an artist. The knot was then tightened by constraining the end of the ribbon to move apart at 10 cm/s , and running the simulation.

Figure 7, left, shows the ribbon after 2 seconds. Since the velocities of the ends of the ribbons were constrained, the knot material became arbitrarily stretched once the knot was tight. The forces pressing the two ribbons into each other thus grew unbounded, but the two ribbons never interpenetrated, nor were other collision-related artifacts observed. It should be stressed that this good behavior did not require the tweaking of the penalty stiffnesses nor any other artificial parameters.

As a second large-scale example, a ribbon similar to the ones in the reef knot simulation was positioned by an artist into a loose bowline knot tied around a cylindrical thin shell of 1334 vertices. The bowline was then tightened by fixing one end of the ribbon and constraining the other to move away from the cylinder at 10 cm/s . Again, the knot successfully tightened with no penetrations or other artifacts (figure 7, right).

8.5. Sphere and Wedge

Inspired by Pandolfi et al. [10], a rigid thin-shell sphere was dropped into a wedged formed by two thin shell triangular prism, shown in Figure 8. Each prism has an isosceles base with width 12.92 cm and height 20.05 cm , and length 38.41 cm . The prisms contain 71 vertices each. The sphere contains 92 vertices, has radius 4.97 cm and begins the simulation 20.84 cm above the ground plane on which the prisms rest. The sphere has initial downwards velocity of -100 cm/s (no gravity). The sphere and shells use the same thin shell model as the debris in the above trash compactor example, with bending and stretching stiffness parameters 100000 and 50000 respectively.

As the sphere descends, it enters into multiple contact with the faces of the wedge, which undergo elastic deformation and high-frequency vibration. Despite the large areas of simultaneous contact and

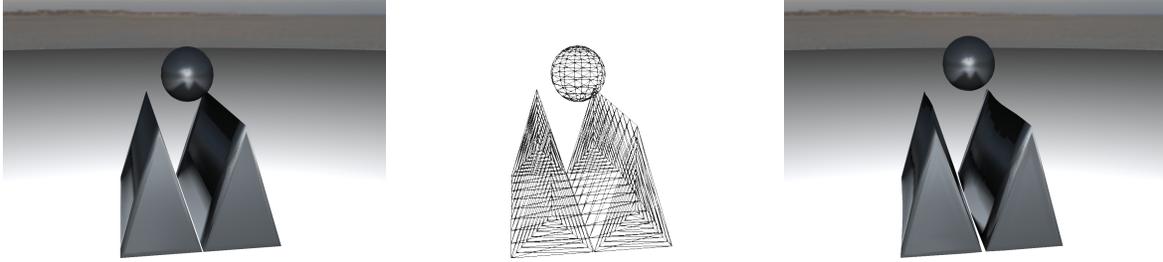


Figure 8: A sphere falling into a wedge, at the beginning of the simulation (left and center) and 0.42 seconds later, after the sphere has reflected off of the wedge (right). The center figure shows the mesh elements of the bodies.

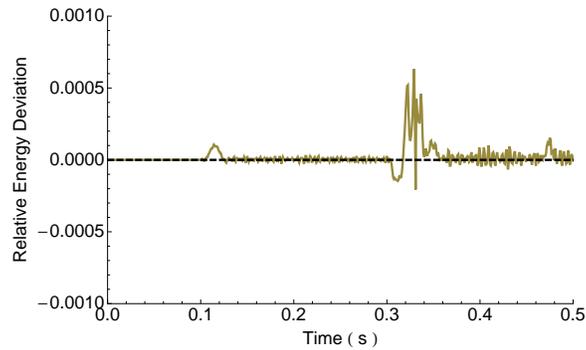


Figure 9: The relative error in energy of the wedge-sphere system as a function of time. The energy oscillates about its initial value without drift.

high velocity at the time of impact, the energy of this system, plotted in Figure 9, exhibits good behavior and does not drift.

8.6. Draping on Spikes

ACM’s ability to robustly handle degenerate geometry was tested by dropping two 1994-vertex, 15 cm \times 50 cm cloth meshes (stretching stiffness 500, bending stiffness 0.1, stretching damping 1.0, bending damping 0.1) on top of a rigid 20 cm \times 20 cm plate from which protrude 36 7.8 cm spikes (see figure 10). Each spike was modeled using six highly-degenerate, sliver triangles: each triangle’s most acute angle measures 3.47 degrees (see figure 11). The cloth was allowed to fall under gravity (9.8 m/s^2) and drape on top of the spikes until it had come to rest. No penetrations, oscillations, or other artifacts were observed.

After the cloth came to rest, the bottom cloth was pulled out from under the top one by constraining one side of the cloth to move at 10 cm/s parallel to and away from spiked plate; see figure 12. The bottom cloth scraped against the spikes and slid, with no dissipation, against the top cloth. No interpenetrations occurred.

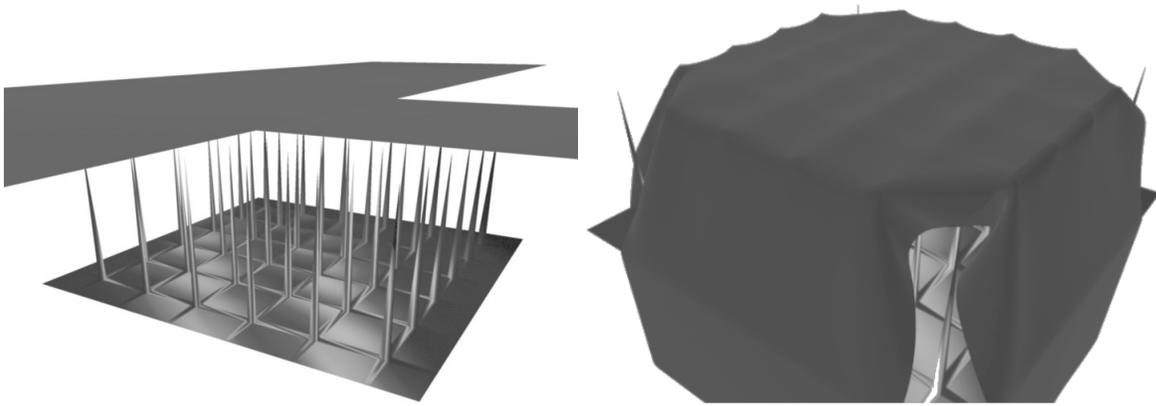


Figure 10: Two cloth rectangles were draped on a bed of spikes. The system at the start of the simulation (left), and after the cloth has come to rest (right).

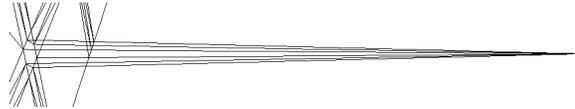


Figure 11: A close-up of one of the spikes; the spike has been rotated clockwise 90 degrees to conserve space. Each spike is composed of six triangles with apex angle 3.47 degrees.

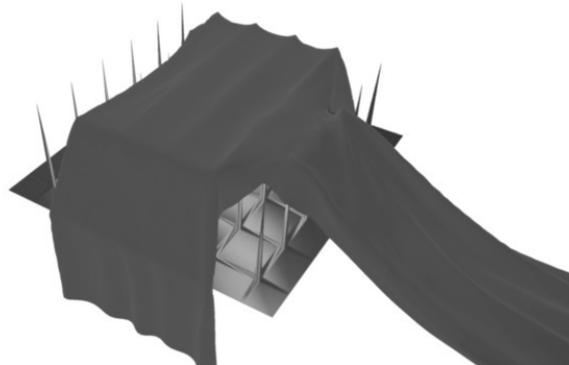


Figure 12: After the cloth came to rest, the bottom cloth was pulled out from under the top one. The simulation 3 seconds after pulling began.

8.7. *Trash Compactor*

Various coarse thin-shell solid objects (platonic solids, tori, etc.) modeled as triangle meshes were placed in a rectangular box measuring 71.5 cm \times 36.7 cm \times 9.3 cm. The four sides were scripted to close in and compress the objects within: the length at 20 cm/s, and the width at 10 cm/s. All objects were given the same material parameters (stretching stiffness 1000, bending stiffness 10, stretching damping

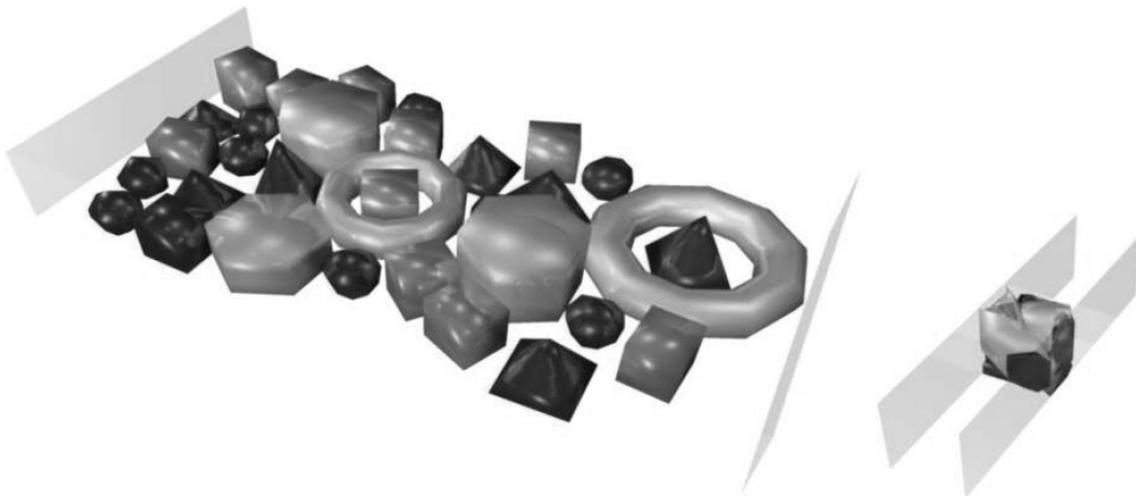


Figure 13: Walls close in and compress various thin-shell objects. The beginning (left) and end (right) of the simulation.

15, bending damping 0.5) and held to the floor of the box by gravity (9.8 m/s^2). Figure 13 shows the box at the beginning of the simulation, and after the simulation had run for 3.4 seconds. A simple plastic deformation model, described by Bergou et al. [44], allowed the objects to crush plastically when stressed by the encroaching walls. Nevertheless, the material forces acting on the objects grew larger as the box decreases to a small fraction of its original volume, yet no object penetrated any other object or wall, as guaranteed by the method.

9. Effects of Stiffness and Thickness Parameters

The proposed algorithm requires choosing values for two parameters: k , the stiffness of the outermost layer, and η , the outermost layer’s thickness. In penalty methods the choice of stiffness is often critical – there is no guaranteed maximum degree of constraint violation, so failure to judiciously set the stiffness to a problem-dependent optimal value can result in arbitrary large penetrations and errors in trajectories and, in the worst case, the *tunnelling* of objects through each other.

The proposed method using discrete penalty layers, by contrast, is guaranteed by construction to prevent interpenetrations for *any* choice of stiffness parameter. Different choices of parameter value do, however, affect the trajectory of the simulation – increasing the stiffness decreases the time objects are in contact during impact events, and more closely approximates exact enforcement of the constraint $g_\eta > 0$. Changing the stiffness also requires changing the time step of penalty force events to retain stability and good energy behavior. A full theoretical understanding of the relationship between stiffnesses and stable time steps for AVIs remains future work; for instance recent research [45] suggests that poorly chosen time step ratios can lead to resonance instabilities. Nevertheless, in practice, a penalty time step proportional to $\frac{1}{\sqrt{k}}$ was observed to be stable for all experiments described in Section 8.

The choice of thickness η likewise does not affect the method’s non-interpenetration guarantee, but does influence the trajectory, since shrinking η shrinks the distance over which the penalty layers are

permitted to act, approaching exact enforcement of the constraint $g_0 > 0$ as the thickness vanishes. Moreover, since the maximum potential energy V_l of a layer l is proportional to η^2 , for smaller η stiffer, deeper layers will be activated to resolve a given collision, carrying a performance cost.

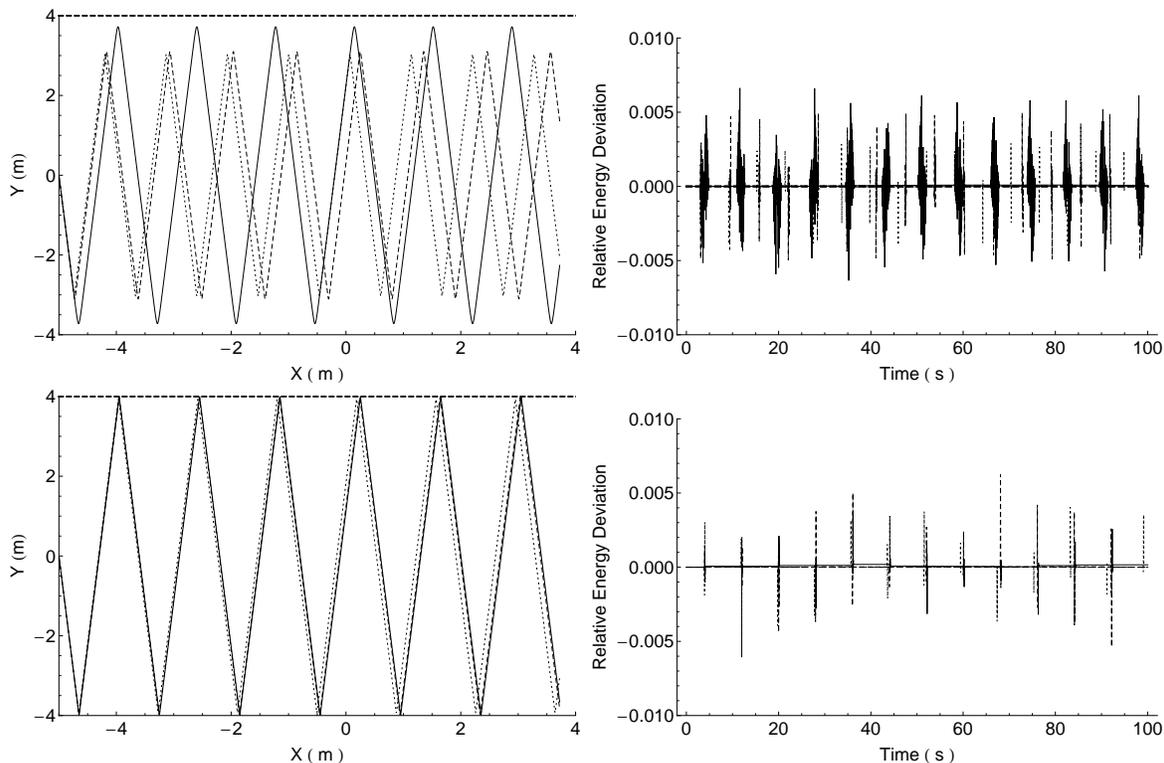


Figure 14: The trajectory (left) and energy behavior over time (right) of a single ball bouncing elastically between two parallel walls in two dimensions, for two different outer layer thicknesses (top row: $\eta = 1$ m; bottom row: $\eta = 0.1$ m) and three outer layer stiffnesses (solid line: $k = 1$; dashed line: $k = 0.1$; dotted line: $k = 0.01$). Although the choice of these parameters affects the trajectory of the system, good energy behavior is guaranteed for any such choice.

To explore the effects of k and η on a simple simulation, a particle in $2D$ of unit mass was simulated bouncing between two parallel walls 1 m apart (no gravity). The particle was initially positioned midway between the walls, with velocity 1 m/s at 85 degrees to the bottom wall. Figure 14 shows the trajectory and energy of the particle for various choices of k and η .

10. Conclusion and Future Work

A framework for asynchronous, structure-preserving handling of contact and impact has been presented. Provable guarantees were established for this framework: impact handling is robust, allowing no penetrations or tunneling; the good properties of AVIs are preserved, such as a discrete Noether's Theorem and discrete multisymplectic structure; and for well-posed problems, the amount of computation required to simulate the problem is bounded and in particular finite. Good long-time energy behavior,

conjectured to accompany multisymplectic structure, was confirmed empirically by both didactic and challenging, large-scale experiments. Modifications to allow for simple dissipative phenomena, such as a coefficient of restitution, were described. Data structures and algorithms to improve performance, such as the use of separating slabs to prune inactive penalty layers, were briefly discussed. Implementation details for these and other optimizations, as well as ideas for future improvements to the algorithms that promise to substantially decrease computation time, are treated more comprehensively by Harmon et al. [17].

Missing from the basic asynchronous contact framework described by this paper is comprehensive handling of friction, particularly static friction. Static friction conflicts fundamentally with asynchrony: in an asynchronous simulation, contact between a pair of elements is resolved piecemeal, by summing the impulses at many different times contributed by many different penalty layers. At any given moment of time it is unclear how to define a total normal force, an element necessary for the robust treatment of even the most elementary static friction models. Successfully merging the handling of friction with the asynchronous framework, to allow simulations of systems such as a standing house of cards, remains a challenging area for future research.

11. Acknowledgements

The authors thank David Mooy for modeling the knots, and Igor Boshoer, Matt Kushner, Kori Valz for lighting and rendering. We are grateful for the valuable feedback provided by Breannan Smith, Miklós Bergou, Rony Goldenthal, Bernhard Thomaszewski, Max Wardetzky, and the anonymous reviewers. This work was supported in part by the Sloan Foundation, the NSF (MSPA Award Nos. IIS-05-28402 and IIS-09-16129, CSR Award No. CNS-06-14770, CAREER Award No. CCF-06-43268), and the Amazon Elastic Compute Cloud. The Columbia authors are supported in part by generous gifts from Adobe, ATI, Autodesk, mental images, NVIDIA, Side Effects Software, the Walt Disney Company, and Weta Digital.

References

- [1] A. Lew, J. Marsden, M. Ortiz, M. West, Asynchronous variational integrators, *Archive for Rational Mechanics and Analysis* 167 (2003) 85–146.
- [2] Y. Suris, Hamiltonian methods of runge-kutta type and their variational interpretation, *Matematicheskoe Modelirovanie* 2 (1990) 78–87.
- [3] R. MacKay, Some aspects of the dynamics of hamiltonian systems, in: *The Dynamics of Numerics and the Numerics of Dynamics*, Oxford University Press, 1992, pp. 137–193.
- [4] J. Marsden, M. West, Discrete mechanics and variational integrators, *Acta Numerica* 10 (2001) 357–514.
- [5] M. West, Variational integrators, Ph.D. thesis, California Institute of Technology, 2004.
- [6] E. Hairer, C. Lubich, G. Wanner, *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*, Springer-Verlag, second edition, 2006.
- [7] E. Barth, B. Leimkuhler, S. Reich, A time-reversible variable-stepsize integrator for constrained dynamics, *SIAM Journal on Scientific Computing* 21 (1999) 1027–1044.
- [8] C. Kane, J. E. Marsden, M. Ortiz, M. West, Variational integrators and the newmark algorithm for conservative and dissipative mechanical systems, *International Journal for Numerical Methods in Engineering* 49 (2000) 1295–1325.
- [9] T. A. Laursen, G. R. Love, Improved implicit integrators for transient impact problems - geometric admissibility within the conserving framework, *International Journal for Numerical Methods in Engineering* 53 (2002) 245–274.
- [10] A. Pandolfi, C. Kane, J. E. Marsden, M. Ortiz, Time-discretized variational formulations of non-smooth frictional contact, *International Journal for Numerical Methods in Engineering* 53 (2002) 1801–1829.
- [11] R. Rangarajan, R. Ryckman, A. Lew, Towards long-time simulation of soft tissue simulant penetration, in: *Proceedings of the 26th Army Science Conference*, pp. 1–8.
- [12] R. Ryckman, A. Lew, An explicit asynchronous contact algorithm (in preparation).
- [13] P. Wriggers, *Computational Contact Mechanics*, John Wiley and Sons Ltd, 2002.
- [14] T. Laursen, *Computational Contact and Impact Mechanics*, 2002.
- [15] T. Belytschko, M. O. Neal, Contact-impact by the pinball algorithm with penalty and lagrangian methods, *International Journal for Numerical Methods in Engineering* 31 (1991) 547–572.
- [16] J. Nocedal, S. Wright, *Numerical Optimization*, Springer, 2000.
- [17] D. Harmon, E. Vouga, B. Smith, R. Tamstorf, E. Grinspun, Asynchronous contact mechanics, in: *ACM SIGGRAPH 2009 papers*, pp. 1–12.

- [18] L. J. Guibas, Kinetic data structures: a state of the art report, in: Proceedings of the Third Workshop on the Algorithmic Foundations of Robotics on Robotics : the Algorithmic Perspective, pp. 191–209.
- [19] H. Hertz, Study on the contact of elastic bodies, *Journal für die Reine und Angewandte Mathematik* 29 (1882) 156–171.
- [20] T. Belytschko, W. K. Liu, B. Moran, *Nonlinear Finite Elements for Continua and Structures*, John Wiley and Sons, 2006.
- [21] P. Wriggers, Finite element algorithms for contact problems, *Archives of Computational Methods in Engineering* 2 (1995) 1–49.
- [22] J. T. Oden, Exterior penalty methods for contact problems in elasticity, in: Bathe, Stein, Wunderlich (Eds.), *Europe-US Workshop: Nonlinear Finite Element Analysis in Structural Mechanics*, Springer, 1980.
- [23] N. Kikuchi, J. T. Oden, *Contact Problems in Elasticity: A Study of Variational Inequalities and Finite element Methods*, SIAM, 1988.
- [24] W. J. T. D. T. Belytschko, G. Ventura, A monolithic smoothing-gap algorithm for contact-impact based on the signed distance function, *International Journal for Numerical Methods in Engineering* 55 (2002) 101–125.
- [25] D. Perić, D. R. J. Owen, Computational model for 3-D contact problems with friction based on the penalty method, *International Journal for Numerical Methods in Engineering* 35 (1992) 1289–1309.
- [26] T. Hughes, R. L. Taylor, J. L. Sackman, A. Curnier, W. Kanoknukulchai, A finite element method for a class of contact-impact problems, *Computer Methods in Applied Mechanics and Engineering* 8 (1976) 249–276.
- [27] B. Nour-Omid, P. Wriggers, A two-level iteration method for solution of contact problems, *Computer Methods in Applied Mechanics and Engineering* 54 (1986) 131–144.
- [28] R. L. Taylor, P. Papadopoulos, On a finite element method for dynamic contact/impact problems, *International Journal for Numerical Methods in Engineering* 36 (1993) 2123–2140.
- [29] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, 1984.
- [30] P. Wriggers, J. C. Simo, R. L. Taylor, Penalty and augmented lagrangian formulations for contact problems, in: J. Middleton, G. N. Pande (Eds.), *Proceedings of NUMETA 85 Conference*, Balkema, 1985.
- [31] J. C. Simo, T. A. Laursen, An augmented lagrangian treatment of contact problems involving friction, *Computers and Structures* 42 (1992) 97–116.
- [32] J. C. Simo, P. Wriggers, R. L. Taylor, A perturbed lagrangian formulation for the finite element solution of contact problems, *Computer Methods in Applied Mechanics and Engineering* 50 (1985) 163–180.

- [33] C. Kane, E. A. Repetto, M. Ortiz, J. E. Marsden, Finite element analysis of nonsmooth contact, *Computer Methods in Applied Mechanics and Engineering* 180 (1999) 1–26.
- [34] F. Cirak, M. West, Decomposition contact response (DCR) for explicit finite element dynamics, *International Journal for Numerical Methods in Engineering* 64 (2005) 1078–1110.
- [35] C. Lanczos, *The Variational Principles of Mechanics*, Dover Publications, fourth edition, 1986.
- [36] A. Lew, *Variational Time Integrators in Computational Solid Mechanics*, Ph.D. thesis, California Institute of Technology, 2003.
- [37] A. P. Veselov, Integrable discrete-time systems and difference operators, *Functional Analysis and Its Applications* 22 (1988) 83–93.
- [38] J. Moser, A. P. Veselov, Discrete versions of some classical integrable systems and factorization of matrix polynomials, *Communications in Mathematical Physics* 139 (1991) 217–243.
- [39] J. M. Wendlandt, J. E. Marsden, Mechanical integrators derived from a discrete variational principle, *Physica D: Nonlinear Phenomena* 106 (1997) 223–246.
- [40] J. T. Klosowski, M. Held, J. S. B. Mitchell, H. Sowizral, K. Zikan, Efficient collision detection using bounding volume hierarchies of k-dops, *IEEE Transactions on Visualization and Computer Graphics* 4 (1998) 21–36.
- [41] R. Weller, G. Zachmann, Kinetic separation lists for continuous collision detection of deformable objects, in: *Third Workshop in Virtual Reality Interactions and Physical Simulation (Vriphys)*.
- [42] Y. Gingold, A. Secord, J. Y. Han, E. Grinspun, D. Zorin, *A Discrete Model for Inelastic Deformation of Thin Shells*, Technical Report, New York University, 2004.
- [43] E. Grinspun, A. Hirani, M. Desbrun, P. Schrder, Discrete shells, in: *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 62–67.
- [44] M. Bergou, S. Mathur, M. Wardetzky, E. Grinspun, Tracks: toward directable thin shells, in: *ACM SIGGRAPH 2007 papers*, p. 50.
- [45] W. Fong, E. Darve, A. Lew, Stability of asynchronous variational integrators, in: *Proceedings of the 21st International Workshop on Principles of Advanced and Distributed Simulation, PADS '07*, pp. 38–44.