Implementation of Dynamics-Aware Coarsening for Fabrication Design

Desai Chen
MIT CSAIL & Adobe Research

David I.W. Levin
University of Toronto

Wojciech Matusik
MIT CSAIL

Danny M. Kaufman
Adobe Research

1 Dynamic Contact-Impact Solver

Implicit time stepping method At each time step our implicit contact method must jointly satisfy the discrete equations of motion

$$\begin{aligned}\mathsf{M}\delta^{t+1} - \mathsf{b}^t - \tfrac{h^2}{4}\mathsf{F}(\mathsf{q}^{t+1}) + \tfrac{h^2}{4}\mathsf{D}(\mathsf{q}^{t+1})\mathsf{v}^{t+1} \\ - \tfrac{h^2}{2}\mathsf{N}\alpha - \tfrac{h^2}{2}\mathsf{T}\beta = 0,\end{aligned} \tag{1}$$

updates

$$\begin{aligned}\mathsf{b}^t &= h\mathsf{M}\mathsf{v}^t + \tfrac{h^2}{4}\mathsf{F}(\mathsf{q}^t) - \tfrac{h^2}{4}\mathsf{D}(\mathsf{q}^t)\mathsf{v}^t, \\ \mathsf{q}^{t+1} &= \mathsf{q}^t + \delta^{t+1},\end{aligned} \tag{2}$$

contact conditions[1]

$$0 \le \alpha \perp \mathsf{N}^T \delta^{t+1} \ge 0, \tag{3}$$

variational maximal dissipation conditions

$$\min_{\beta}\{\beta^T \mathsf{T}^T(\tfrac{2}{h}\delta^{t+1} - \mathsf{v}^t) : \mu_k \bar{\alpha}_k \ge \|\bar{\beta}_k\|,\ \forall k \in \mathcal{C}\}, \tag{4}$$

and the impact projection

$$\begin{aligned}\mathsf{c} &= 2\delta^{t+1} - h\mathsf{v}^t, \\ \mathsf{A} &= \mathsf{M} + \frac{h^2}{4}\mathsf{K}(\mathsf{q}^{t+1}) + \frac{h}{2}\mathsf{D}(\mathsf{q}^{t+1}), \\ \mathsf{d}^* &= \underset{\mathsf{d}}{\operatorname{argmin}}\left\{\|\mathsf{d} - \mathsf{c}\|^2_{\mathsf{A}} : \mathsf{N}^T\mathsf{d} \ge 0\right\}, \\ \mathsf{v}^{t+1} &= \frac{1}{h}\mathsf{d}^*,\end{aligned} \tag{5}$$

to convergence with an iterative solver.

Modified Newton-Raphson with frictional contact To construct our solver we first consider time-stepping without contact forces. At each time step we seek a displacement update δ^{t+1} satisfying

$$\mathsf{f}(\delta^{t+1}) = \mathsf{M}\delta^{t+1} - \mathsf{b}^t - \tfrac{h^2}{4}\mathsf{F}(\mathsf{q}^{t+1}) + \tfrac{h^2}{4}\mathsf{D}(\mathsf{q}^{t+1})\mathsf{v}^{t+1} = 0, \tag{6}$$

with

$$\begin{aligned}\mathsf{q}^{t+1} &= \mathsf{q}^t + \delta^{t+1}, \\ \mathsf{v}^{t+1} &= \tfrac{2}{h}\delta^{t+1} - \mathsf{v}^t.\end{aligned} \tag{7}$$

Ignoring $\frac{\partial \mathsf{D}}{\partial \mathsf{q}}$ we set

$$\mathsf{H}(\delta) = \mathsf{M} + \frac{h^2}{4}\mathsf{K}(\mathsf{q}^t + \delta) + \frac{h}{2}\mathsf{D}(\mathsf{q}^t + \delta). \tag{8}$$

and have $\nabla \mathsf{f} \simeq \mathsf{H}$. In the following we will reserve superscripting with indexing t for time step increments and superscripting with indexing i for iteration increments. Modified Newton-Raphson then approximates the linearization of f, at iterate i, around δ^i as

$$\mathsf{f}(\delta^{i+1}) \simeq \mathsf{f}(\delta^i) + \mathsf{H}(\delta^i)(\delta^{i+1} - \delta^i). \tag{9}$$

We then find the improved estimate of displacement δ^{i+1} by line search on the descent direction

$$\delta^i - \mathsf{H}(\delta^i)^{-1}\mathsf{f}(\delta^i). \tag{10}$$

With contact, at each Newton iterate we are now solving for updated triples of both displacement and boundary contact and friction forces, (δ, α, β). Applying the same modified linearization of (1) about δ^{i-1} then gives

$$\mathsf{f}(\delta^{i-1}) + \mathsf{H}(\delta^{i-1})(\delta^i - \delta^{i-1}) - \tfrac{h^2}{2}\mathsf{N}\alpha - \tfrac{h^2}{2}\mathsf{T}\beta = 0, \tag{11}$$

Setting

$$\begin{aligned}\mathsf{r}(\delta) = \mathsf{b}^t + \tfrac{h^2}{4}\mathsf{F}(\mathsf{q}^t + \delta) - \tfrac{h^2}{4}\mathsf{D}(\mathsf{q}^t + \delta)(\frac{2}{h}\delta - \mathsf{v}^t) \\ + \big[\frac{h^2}{4}\mathsf{K}(\mathsf{q}^t + \delta) + \frac{h}{2}\mathsf{D}(\mathsf{q}^t + \delta)\big]\delta\end{aligned} \tag{12}$$

at each Newton iterate, the linearized contacting system we want to solve is then

$$\begin{aligned}&\mathsf{H}(\delta^{i-1})\delta^i = \mathsf{r}(\delta^{i-1}) + \tfrac{h^2}{2}\mathsf{N}\alpha + \tfrac{h^2}{2}\mathsf{T}\beta, \\ &0 \le \lambda \perp \mathsf{N}^T\delta_{i+1} \ge 0, \\ &\min_{\beta}\{\beta^T\mathsf{T}^T(\tfrac{2}{h}\delta^{t+1} - \mathsf{v}^t) : \mu_k\bar{\alpha}_k \ge \|\bar{\beta}_k\|,\ \forall k \in \mathcal{C}\}\end{aligned} \tag{13}$$

or, equivalently

$$\begin{aligned}&0 \le \lambda \perp \tfrac{h^2}{2}\mathsf{N}^T\mathsf{H}(\delta^{i-1})^{-1}\mathsf{N}\alpha \\ &\qquad + \mathsf{N}^T\mathsf{H}(\delta^{i-1})^{-1}\big[\mathsf{r}(\delta^{i-1}) + \tfrac{h^2}{2}\mathsf{T}\beta\big] \ge 0, \\ &\min_{\beta}\{\beta^T\mathsf{T}^T(\tfrac{2}{h}\delta^{t+1} - \mathsf{v}^t) : \mu_k\bar{\alpha}_k \ge \|\bar{\beta}_k\|,\ \forall k \in \mathcal{C}\}\end{aligned} \tag{14}$$

with the update $\delta^i = \mathsf{H}(\delta^{i-1})^{-1}\big[\mathsf{r}(\delta^{i-1}) + \tfrac{h^2}{2}\mathsf{N}\alpha + \tfrac{h^2}{2}\mathsf{T}\beta\big]$.

Inner-loop solve of Newton steps To solve each Newton step we first backsolve to get

$$\begin{aligned}\tilde{\mathsf{N}} &= \mathsf{H}(\delta^{i-1})^{-1}\mathsf{N}, \\ \tilde{\mathsf{D}} &= \mathsf{H}(\delta^{i-1})^{-1}\mathsf{D}, \\ \tilde{\mathsf{r}} &= \mathsf{H}(\delta^{i-1})^{-1}\mathsf{r}(\delta^{i-1})\end{aligned} \tag{15}$$

The solution then simplifies a bit further to finding

$$\begin{aligned}&0 \le \lambda \perp \mathsf{N}^T\tilde{\mathsf{N}}\alpha + \tilde{\mathsf{N}}^T\big[\tfrac{2}{h^2}\tilde{\mathsf{r}} + \tilde{\mathsf{T}}\beta\big] \ge 0, \\ &\min_{\beta}\{\beta^T\mathsf{T}^T\tilde{\mathsf{T}}\beta + \beta^T\mathsf{T}^T\big[\tilde{\mathsf{N}}\alpha + \tfrac{2}{h^2}\tilde{\mathsf{r}} - \tfrac{1}{h}\mathsf{v}^t\big] \\ &\qquad : \mu_k\bar{\alpha}_k \ge \|\bar{\beta}_k\|,\ \forall k \in \mathcal{C}\}.\end{aligned} \tag{16}$$

We then solve the Newton step Gauss-Seidel fashion. Each Gauss-Seidel pass iteratively holds all unknowns except for forces at a single contact $k \in \mathcal{C}$ fixed. We solve for the forces at k, update them and then move on to the next $k+1 \in \mathcal{C}$. We run multiple Gauss-Seidel passes through the system until convergence is reached satisfying (16).

---

[1] $\mathsf{x} \perp \mathsf{y}$ is the *complementarity condition* $\mathsf{x}_i\mathsf{y}_i = 0,\ \forall i$.

To solve for the updated forces $(\bar{\alpha}_k^{i+1}, \bar{\beta}_k^{i+1})$ for contact $k \in \mathcal{C}$ in Gauss-Seidel pass $i+1$ we compute

$$\begin{aligned} \mathsf{d}_k = &\sum_{j<k} \tilde{\mathsf{n}}_j \alpha_j^{i+1} + \sum_{j>k} \tilde{\mathsf{n}}_j \alpha_j^i \\ &+ \sum_{j<k} \tilde{\mathsf{T}}_j \beta_j^{i+1} + \sum_{j>k} \tilde{\mathsf{T}}_j \beta_j^i + \frac{2}{h^2} \tilde{\mathsf{r}}_k \in \mathbb{R}^n. \end{aligned} \tag{17}$$

Substituting in (16) we then solve the single-point frictional-contact problem at contact k. This is just the small, three-dimensional problem to find $(\bar{\alpha}_k^{i+1}, \bar{\beta}_k^{i+1}) \in \mathbb{R}^3$ satisfying

$$\begin{aligned} 0 \leq \bar{\alpha}_k^{i+1} \perp \mathsf{n}_k^T \tilde{\mathsf{n}}_k \bar{\alpha}_k^{i+1} + \mathsf{n}_k^T \tilde{\mathsf{T}}_k \bar{\beta}_k^{i+1} + \mathsf{n}_k^T \mathsf{d}_k \geq 0, \\ \bar{\beta}_k^{i+1} = \underset{\bar{\beta}_k}{\operatorname{argmin}} \{ \bar{\beta}_k \mathsf{T}_k^T \tilde{\mathsf{T}}_k \bar{\beta}_k + \bar{\beta}_k \mathsf{T}_k^T (\tilde{\mathsf{n}}_k \bar{\alpha}_k^{i+1} + \mathsf{d}_k - \tfrac{1}{h} \mathsf{v}^t) \\ : \ \mu_k \alpha_k^{i+1} \geq \|\bar{\beta}_k\| \}. \end{aligned} \tag{18}$$

We then update to $(\bar{\alpha}_k^{i+1}, \bar{\beta}_k^{i+1})$ and move on to contact $k+1$.

On convergence of this *inner* Gauss-Seidel iteration to optimal (α^*, β^*) we update to the next Newton step displacement estimate to

$$\delta^i = \tilde{\mathsf{r}} + \tfrac{h^2}{2} \tilde{\mathsf{N}} \lambda^* + \tfrac{h^2}{2} \tilde{\mathsf{T}} \beta^* \tag{19}$$

We form the new needed quantities for the next Newton step in (15) and then solve the next Newton step with Gauss-Seidel. On convergence of the *outer* Newton iteration to satisfying (1), (3), and (4) we then perform the Impact projection in (5) described below and then move to the next time step.

Impact model solve To solve the BBI impact projection step in (5) we can reuse the already computed *final* compliant term from the last iterate in (15). In dual form our BBI impact projection is equivalent to solving the system

$$0 \leq \lambda \perp \mathsf{N}^T \tilde{\mathsf{N}} \lambda + \tilde{\mathsf{N}}^T \mathsf{c} \geq 0, \tag{20}$$

and applying a final velocity update

$$\mathsf{v}^{t+1} \leftarrow \frac{1}{h} (\mathsf{c} + \tilde{\mathsf{N}} \lambda), \tag{21}$$

with c given from (5). We solve the linear-complementarity problem in (20) with projected Gauss-Seidel.

2 SE(3) Projections

During free-flight motion we can project each body's FE state to a fitted rigid body model equipped with rotational $\boldsymbol{R} \in SO(3)$ and translational $\boldsymbol{t} \in \mathbb{R}^3$ degrees of freedom. Per body we choose coordinates so that $\boldsymbol{R}$ rotates from principal-axis–aligned body frame to world frame and $\boldsymbol{t}$ gives the location of center of mass. Corresponding angular and linear momenta are $\boldsymbol{\pi}, \boldsymbol{l} \in \mathbb{R}^3$, with diagonal inertia tensor I and mass m. Nodal positions of material points $\boldsymbol{x}_i$ during rigid motion are then $\boldsymbol{x}_i^t = \boldsymbol{t}^t + \boldsymbol{R}^t(\boldsymbol{x}_i^0 - \boldsymbol{t}^0)$. We set corresponding momenta to $(\boldsymbol{p}_1^T, ..., \boldsymbol{p}_n^T)^T = \mathsf{M}\mathsf{v}^t \in \mathbb{R}^{3n}$, stack nodal vertices as $\mathsf{Q} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_n) \in \mathbb{R}^{3 \times n}$ and then project to rigid state with

$$\begin{aligned} &\boldsymbol{R}^t \leftarrow \underset{\boldsymbol{T} \in SO(3)}{\operatorname{argmin}} \ ||\boldsymbol{T}\mathsf{Q}^t - \mathsf{Q}^0||_F, \\ &\boldsymbol{\pi}^t \leftarrow \sum_{i=1}^n (\boldsymbol{x}_i^t - \boldsymbol{t}^t) \times \boldsymbol{p}_i^t, \quad \boldsymbol{l}^t \leftarrow \sum_{i=1}^n \boldsymbol{p}_i^t, \\ &\boldsymbol{I} \leftarrow \int_{\Omega^0} \rho(\boldsymbol{x})[\boldsymbol{x}][\boldsymbol{x}]^T dV, \quad m \leftarrow \int_{\Omega^0} \rho(\boldsymbol{x}) dV. \end{aligned} \tag{22}$$

We then timestep rigid body state forward through free-flight with DMV [Moser and Veselov 1991], an energy–momentum preserving rigid-body integrator, until the next collision is reached. Upon collision we again need to model elastic behavior and so project back to closest FE state with nodal positions and velocities

$$\begin{aligned} &\boldsymbol{x}_i^t \leftarrow \boldsymbol{t}^t + \boldsymbol{R}^t(\boldsymbol{x}_i^0 - \boldsymbol{t}^0), \\ &\boldsymbol{v}_i^t \leftarrow \tfrac{1}{m} \boldsymbol{l}^t - \boldsymbol{R}^t(\boldsymbol{x}_i^0 - \boldsymbol{t}^0) \times (\boldsymbol{I}^{-1} \boldsymbol{\pi}^t). \end{aligned} \tag{23}$$

Comparing the output trajectories between the full FEM simulation and our hybrid projection trajectory we find a tight match throughout.In our Design Supplemental we show the trajectories obtained from full FEM and the hybrid for our *jumping-over* example. Both simulations are initialized to the same configuration and terminate at first impact. Here the jumper traces out an approximately 20cm long trajectory while the two simulated trajectories differ in the L^∞-norm by 0.63 mm for the linear trajectory, i.e., center of mass, and 0.02 radians in rotational pose over the trajectory.

3 Static Contact Solver

Loading model We observe in experiment that initial loading processes are effectively quasistatic, with no-slip at contacts. We then model the loading phase with a custom static solver that finds equilibrium state subject to satisfying no-penetration contact constraints $\mathsf{p} \geq 0$ and an assumption of infinite (no-slip) friction. We solve for a δ that gives the constrained equilibrium system maximizing frictional work

$$\begin{aligned} F(\mathsf{q}^0 + \delta) + \mathsf{N}\alpha + \mathsf{T}\beta + \mathsf{F}_{load} = 0, \\ 0 \leq \alpha \perp \mathsf{p}(\mathsf{q}^0 + \delta) \geq 0, \\ \alpha \perp \mathsf{T}^T \delta, \ \forall k \in \mathcal{C}. \end{aligned} \tag{24}$$

Static solver Our static solver applies a direct solution approach to compute nodal positions q subject to external loads F_{ext} with infinite-friction (no-slip) unilateral breaking contact. At each iteration step i, we initialize a Newton-Raphson search direction δ^i

$$\mathsf{K}(\mathsf{q}^i)\delta^i = \mathsf{F}_{ext} - \mathsf{F}(\mathsf{q}^i).$$

subject to Dirichlet boundary conditions fixing a set of previously identified *active* contacting boundary vertices identified in the prior iterate. At each iteration we apply a projected line search. We first analyze the depth-component of all the previously identified active contacting points q_j. If q_j penetrates an obstacle, we half the step size of the search direction until the penetration depth for that point is less than $10^{-2} \times dx$ where dx is characteristic rest element size in our discretization. After applying the line search, we then update the set of active contacting vertices as follows and take the next iterate step. Initially, all vertices touching an obstacle boundary are treated as fixed vertices. We then update the active set of contacting vertices by examining force consistency on all vertices currently contacting the boundary. If the force on a contacting boundary vertex opposes the contacting normal direction, we free it by removing it from the active set; if a contacting boundary vertex was previously free and is now penetrating, we project it back to the contact boundary surface and add it to the active set. We run this solve to convergence satisfying equilibrium in (24).

We verify our static solver with respect to full dynamic FEM simulation modeling loading with frictional contact. We find that the relative geometric difference between solutions is 0.2% (Hausdorff distance) while the relative difference between internal energies is 0.4%, with an overall 15X speedup gain from the static solver over the dynamic FEM loading simulation.

4 Stiffness Consistent Mass Matrix

We assemble our full mass matrix from element mass matrices. Within each element e, we integrate

$$\mathbf{M}_e = \int_{\Omega_e} \sum_{i=1}^{8} \rho_e \mathbf{N}_i(\xi) \mathbf{N}_i^T(\xi) d\Omega$$

using 2-point Gauss quadrature. Here $\mathbf{N}_i$s are the tri-linear shape functions used for our FE calculations including stress computation. Thus the mass matrix is stiffness-consistent and consistently captures both linear and angular momentum of the system.

5 Constraint assembly

For each contact $k \in \mathcal{C}$, the relative acceleration between material points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ (at contact k) can be expressed via the map $\mathsf{\Gamma}_k : \dot{\mathsf{q}} \rightarrow \dot{\boldsymbol{x}}_i - \dot{\boldsymbol{x}}_j$. If $\boldsymbol{y} \in \mathbb{R}^3$ is a force applied to point $\boldsymbol{x}_i$, and an equal but opposite force is applied to point $\boldsymbol{x}_j$, $\mathsf{\Gamma}_k^T \boldsymbol{y}$ is the resulting generalized force applied to the contacting system. For contact k, the map $\mathsf{\Gamma}_k$ is the sparse matrix with non-zero entries corresponding to nodes participating in the contacting vertex or vertices. For nodal vertex-boundary contact, a single identity entry corresponding to the node's DoFs is sufficient. For vertex-quadrilateral contacts, we compute the signed identity entries for the five participating face node DoFs weighted by the bilinear weights of the contacting points in the face quadrilateral.

References

MOSER, J., AND VESELOV, A. P. 1991. Discrete versions of some classical integrable systems and factorization of matrix polynomials. *Communications in Mathematical Physics 139*, 2, 217–243.

Design Decisions for Dynamics-Aware Coarsening

Desai Chen
MIT CSAIL & Adobe Research
David I.W. Levin
University of Toronto
Wojciech Matusik
MIT CSAIL
Danny M. Kaufman
Adobe Research

1 Predictive Simulation for Design

Accurate physical modeling of transient dynamics with contact is validated by experiment and generally requires simulation at close to convergent spatial and temporal grid sizes. This makes even a single forward dynamic simulation run in 3D prohibitively expensive. On the other end of the spectrum physically based animation methods seek efficiency by pushing simulations towards maximally stable time-step sizes and coarsest possible spatial meshes to obtain visually appealing but generally highly inaccurate dynamics. Here we detail our experiments and investigations towards designing predictive and efficient simulation of high-speed dynamics under frictional contact, impact, loading and free-flight suitable for fabrication design.

Sources of Error Errors between simulated and real-world fabricated results can be attributed to three primary sources:

1. inaccuracies in simulation modeling,
2. inaccuracies in material modeling, and
3. fabrication tolerances.

We focus on how to efficiently target and address these sources of error in simulation. If all physical quantities must be accurately captured in simulation in order to design for dynamics, then our assessment is that high-performance FEM is currently unable to solve transient dynamics at speeds practical for even a single simulation, much less the many simulations required for design optimization iterations. If, on the other hand, there is a subset of critical quantities whose accuracy matters most in predictive design, then we are doing too much and likely the wrong kinds of work in applying traditional FEM simulation models that attempt to optimize away all errors. In identifying what properties matter there is then hope that a frugal and targeted simulation algorithm can be constructed to enable a tractable design optimization.

With desktop and numerical experiments we first identify critical physical properties in transient dynamics for which error has a large effect on design behavior. With analysis and simulation tests we then construct a discrete physical model and simulation framework that predictively resolves these properties. We begin by detailing our model problem below.

We observe that highly dynamic motion trajectories for fabricated mechanisms are generally decomposable into stages of

1. *Loading*, where we store energy via deformation—energy is loaded into the system by an applied external force;
2. *Launch*, where we transfer internal energy to kinetic energy while remaining in contact;
3. *Flight*, where an object loses contact with its environment and is carried into contact-free motion by its inertia; and
4. *Landing*, following *flight*, where an object regains contact—kinetic energy is transferred back into internal energy via collision induced deformation, after which an object may maintain contact while in dynamic motion or in rest.

See our main paper for an illustration of these stages.

Model problem We begin by examining a simple 3D-printed jumping model, *the forward flipper*, that captures the full range of dynamic stages from *loading* to *launch* to *flight* to *landing*. We load the jumping model by pressing down and then releasing. The design task for this flipper is to find a shape that, upon release, jumps forward into the air, flips once and then lands stably on its feet, see our main paper for illustrations of the flipper.

2 Critical Physical Properties

After numerous desktop and numerical experiments with a wide collection of 3D-printed jumping models we observe that the following properties are each critical to predicting the success or failure of a design in a subset of the above dynamic phases:

Deformation: to accurately resolve *loading*, *launching* and *landing*, deformation shape must be well-captured. Deformation state initiating *flight* is critical to determining trajectory. However, during the later prolonged *flight* stages, energy stored in deformation modes damps rapidly and we observe that these modes are generally unimportant in determining the overall flight path.

Energy: the predictive modeling of *loading* energy, *launching* time and *landing* stability all require accurate bookkeeping of the energy loaded into the system, its release into motion, its damping over time, and its absorption upon collision.

Contact in space-time: the dynamics of *loading*, *launching* and *landing* are highly sensitive to localizing the locations and times where surfaces separate and collide. Small changes in contact state often result in large changes to later trajectory. Finite friction models, consistency between surface tractions and boundary contacts, and proper impact modeling are key to getting this right.

Orientation: in *flight*, ballistic motions often involve extremely rapid rotations while small variations in orientation and angular momentum in *launch* will often give very different *flight* trajectories. To predictively avoid collisions with obstacles and correctly orient landing poses, rotational state and angular momentum need to be tightly tracked.

3 Simulation Algorithm Analysis

We begin with a standard FEA framework for 3D printed materials, see our main paper. To capture stiff elastic response of 3D-printed materials we use the neo-Hookean material model, augmented with a Rayleigh damping model to capture transient dissipation of vibrations, and discretize with second order, hexahedral finite elements.

As a baseline, we first attempt to simulate our model test problem, the *forward flipper*, with validated, nonlinear FEM. Validated, accurate simulation matching experimental loading, release, launch, flight, and landing behaviors in our experiments requires an average of 791 seconds of computation *per timestep*, with a step size of $h = 10^{-5}$, on an Amazon EC2 compute-optimized instance with 4 CPU threads (Intel Xeon E5-2666 v3, 2.9Ghz).

Consistent with our observations in the last section, we find that simply decreasing our overall discretization resolution in both space and time introduces unacceptable inaccuracies that make optimization unworkable. Specifically, as we proceeded to larger mesh and time step sizes we obtain inaccuracies in loading shapes and energies, launch break-away times and configurations, flight trajectories, landing times and configurations, as well as spurious instabilities, especially upon launch and impact.

With this confirmation that traditional FEM will not be practical, we next analyze and detail how we resolve our identified physical properties for the predictive optimization of our dynamic design task while avoiding the high cost of out-of-the-box high-resolution FEM simulation. We begin by re-visiting each property before describing the detailed construction in the following section.

Deformation To enable efficient simulation we want the coarsest possible spatial discretization of our material. Coarse-resolution discretizations suffer from numerical stiffness, where the numerically simulated material is stiffer than the modeled material. However, we observe that for a large regime of these coarsenings these same meshes are *not kinematically locked*—this range of coarse models captures deformation, just not stiffness. This is the starting observation for our Dynamics Aware Coarsening (DAC) method. DAC first determines and applies the coarsest discretization that captures deformation as given by primary mass-PCA deformation modes matched to the convergent FEM model. We then match the *numerical stiffness* of our coarsened model to a calibrated 3D-printed rig model. See our our main paper for details on DAC.

While deformation is critical to *loading*, *launch*, and *landing*, during prolonged *flight* phases we observe that the potential energy stored in the deformation modes of an object rapidly damp to have negligible effect in determining the next point of collision that initiates a landing phase. Thus we track the time-varying elastic potential energy stored in our simulated model. When damping causes this internal potential to fall to zero, during free-flight, we project our full DAC discretization to a rigid body model with SE(3) variables. We integrate our rigid model forward until the next collision is reached, at which point we map rigid-body state back to the DAC model to capture the new deformation dynamics at impact. See our Implementation Supplemental for details of the projection process. Figure 1 shows trajectories of a jumper computed using high-resolution finite element simulation with the hybrid DAC-SE(3) simulation where we note that the hybrid yields a trajectory that tightly tracks the gold-standard FE simulated result.

Energy Even if we employ quite small time-step sizes and high-resolution spatial meshes, numerical dissipation from implicit Euler still incorrectly models the transfer of energy to the material during *loading*, and likewise the transfer of the load energy to kinetic energy during *launching*. We similarly noted corresponding issues with higher-order BDF2. This leads to widely differing trajectories and the issue is only exacerbated if we seek to maximize time step sizes for efficiency. If not fixed we particularly notice incorrect breakaway times and locations as contact surfaces peel from the boundary, as well as large disparities in the linear and angular velocities as objects leave the ground during launch.

For predictive energy bookkeeping at larger time steps we choose an implicit, geometric time integrator—implicit Newmark [Kane et al. 2000]. Before adopting implicit Newmark we first experimented with less computationally expensive geometric integrators: Verlet and linearly implicit Newmark [Tao and Owhadi 2016]. When integrating our deformable models we observe that fully explicit, second-order Verlet is highly unstable in our setting even at remarkably small timesteps. Linearly implicit Newmark provides improved stability and is computationally affordable, with only a single linear solve per time step, but still becomes unstable once we add our contact and impact resolution to the simulation. Implicit Newmark provides the predictive energy tracking we seek, even as we scale to larger time steps we eventually employ.

Contact in space-time With our coarse discretization model in place, simulation becomes bottlenecked in our dynamic FEM simulation of *loading*. However, we note that the loading process is effectively quasistatic, and so, if we are careful, we should be able to replace the time stepped solves of loading with a single static solve. To do this, however, we next observe that we need to predictively capture the final loaded equilibrium pose *subject to frictional contact*. During loading portions of contacting surfaces peel away due to breaking contact with the unilateral boundary constraint.

To capture this breaking contact effect we need to efficiently model frictional contact in our static solve in order to correctly capture the loading pose and potential energy that determine initial *launch* conditions. Many static solvers treat boundary conditions with bilateral constraints and so would be unable to model breaking contact. In our Implementation Supplemental we detail a static loading solver using projected contact and an infinite-friction model that we validate against the full dynamic FEM frictional-contact solution for loading.

While geometric time integration with implicit Newmark gives us better energy bookkeeping for dynamic FE simulation, we still need to capture frictional contact and impact behavior during mode shifts from *launching* to *flight* and from flight to next collision accurately. We initially looked at inexpensive, explicit contact models and contact projection methods but confirmed that they are much too inaccurate in localizing contact times and locations. Fully implicit contact modeling is, as expected, much better but still fails to correctly model the breakaway and restitution behaviors that are critical to capturing *launching* and *landing*. In our main paper we detail our BBI method for impact modeling that allows us to match observed experimental behavior.

Orientation Accurate *flight* with large rotations and high angular speeds, such as those we deal with here, are not satisfied by standard SE(3) time integrators, e.g. exponential Euler. Instead we apply Discrete Moser-Veselov (DMV) [Moser and Veselov 1991] a second-order, energy–momentum preserving time integrator. DMV is inexpensive, and by tightly tracking angular momentum and rotational state allows us to efficiently and predictively track trajectory at larger time steps, while giving us an accurate accounting of kinetic energy to map back to our FE model upon collision. This enables us to predictively model if and when undesirable collisions occur during sample trajectories and likewise inexpensively set accurate collision orientations prior to modeling landings.

Finally, to correctly capture the angular momentum initiating *flight* we require good tracking of momenta during *launch* as well. While the implicit-Newmark model we employ is momentum-preserving, we additionally require a stiffness-consistent mass-matrix to accurately model the correct momenta quantity.

References

KANE, C., MARSDEN, J. E., ORTIZ, M., AND WEST, M. 2000. Variational integrators and the Newmark algorithm for conservative and dissipative mechanical systems. *International Journal for Numerical Methods in Engineering 49*, 10, 1295–1325.

MOSER, J., AND VESELOV, A. P. 1991. Discrete versions of some classical integrable systems and factorization of matrix polynomials. *Communications in Mathematical Physics 139*, 2, 217–243.

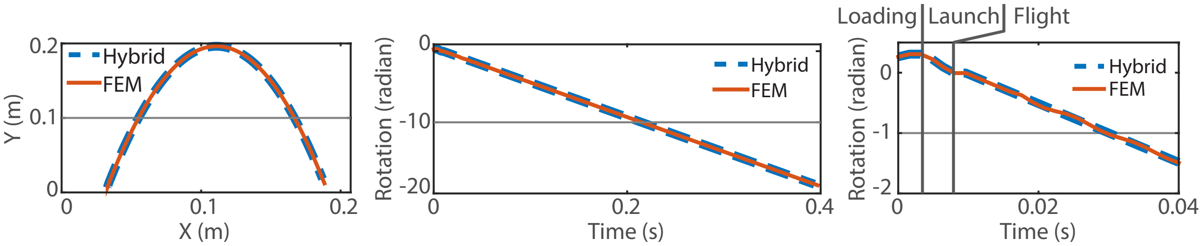

Figure 1: *Comparison between the hybrid simulation with projected rigid body simulation in free-flight and the trajectory obtained by full FEM simulation for our* jumping over *example. On the left, we compare center of mass trajectories. In the middle plot, we compare rotations around Z-axis throughout the jump trajectory. Finally, on the right, we zoom in to the initial trajectory up to 0.05 seconds. Here the oscillations of the FEM model quickly damp out, and throughout, the overall rotation matches the rigid body rotation with a maximal difference of 0.02 radians.*

TAO, M., AND OWHADI, H. 2016. Variational and linearly implicit integrators, with applications. *IMA Journal of Numerical Analysis 36*, 1 (Jan.), 80–107.