# High-Order Incremental Potential Contact for Elastodynamic Simulation on Curved Meshes

Zachary Ferguson
zfergus@nyu.edu
New York University
USA

Pranav Jain
pranav.jain@nyu.edu
New York University
USA

Denis Zorin
dzorin@cs.nyu.edu
New York University
USA

Teseo Schneider
teseo@uvic.ca
University of Victoria
Canada

Daniele Panozzo
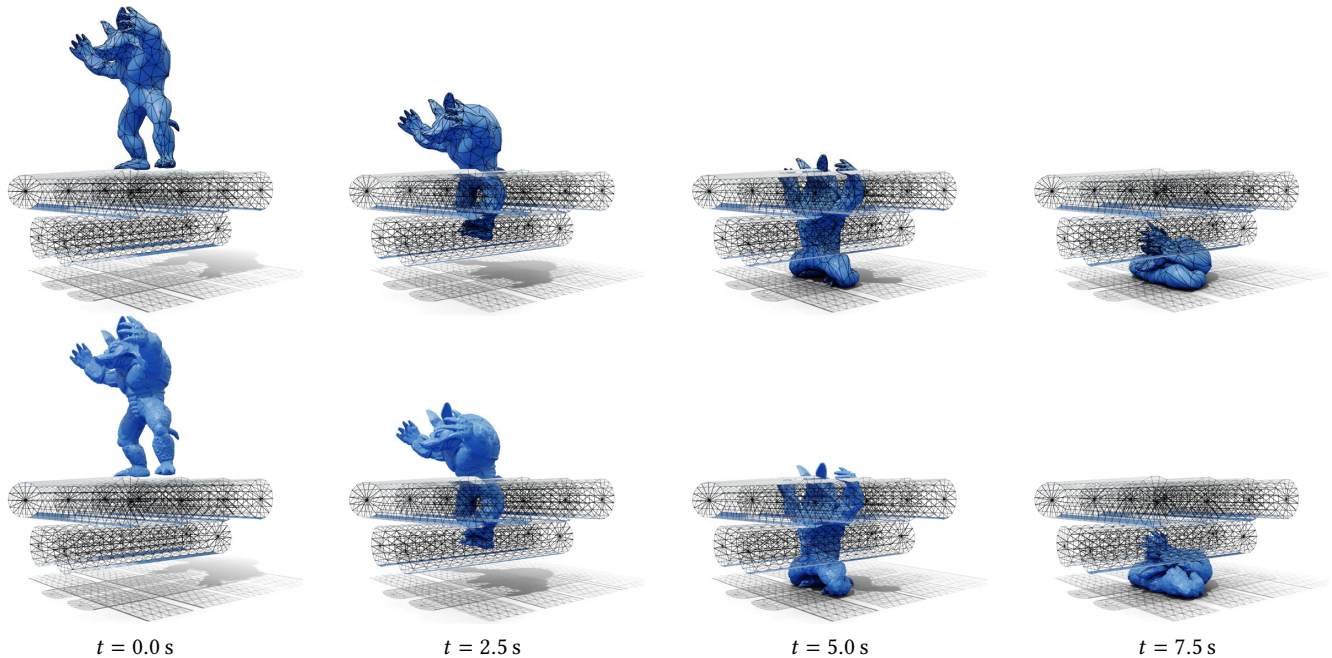panozzo@nyu.edu
New York University
USA

$t = 0.0\,\text{s}$      $t = 2.5\,\text{s}$      $t = 5.0\,\text{s}$      $t = 7.5\,\text{s}$

**Figure 1:** *High-order armadillo-rollers.* **A simulation of an armadillo squished by rollers. We use a high-order volumetric mesh (top row) and deform it with quadratic displacement. To solve collision and compute contact forces, we use a dense linear surface mesh (bottom row) and transfer the deformation and contact forces between the two meshes.**

## ABSTRACT

High-order bases provide major advantages over linear ones in terms of *efficiency*, as they provide (for the same physical model) higher accuracy for the same running time, and *reliability*, as they are less affected by locking artifacts and mesh quality. Thus, we introduce a high-order finite element (FE) formulation (high-order bases) for elastodynamic simulation on high-order (curved) meshes with contact handling based on the recently proposed Incremental Potential Contact (IPC) model.

Our approach is based on the observation that each IPC optimization step used to minimize the elasticity, contact, and friction potentials leads to linear trajectories even in the presence of nonlinear meshes or nonlinear FE bases. It is thus possible to retain the strong non-penetration guarantees and large time steps of the original formulation while benefiting from the high-order bases and high-order geometry. We accomplish this by mapping displacements and resulting contact forces between a linear collision proxy and the underlying high-order representation.

We demonstrate the effectiveness of our approach in a selection of problems from graphics, computational fabrication, and scientific computing.

## CCS CONCEPTS

• **Computing methodologies → Physical simulation**.

## KEYWORDS

Finite element method, Elastodynamics, Frictional contact

## 1 INTRODUCTION

Elastodynamic simulation of deformable and rigid objects is used in countless algorithms and applications in graphics, robotics, mechanical engineering, scientific computing, and biomechanics. While the elastodynamic formulations used in these fields are similar, the accuracy requirements differ: while graphics and robotics applications usually favor high efficiency to fit within strict time budgets, other fields require higher accuracy. In both regimes, finite element (FE) approaches based on a conforming mesh to explicitly partition the object volume are a popular choice due to their maturity, flexibility in handling non-linear material models and contact/friction forces, and convergence guarantees under refinement.

In a FE simulation, a set of elements is used to represent the computational domain and a set of basis functions are used within each element to represent the physical quantities of interest (e.g., the displacement in an elastodynamic simulation). Many options exist for both elements and bases. Due to the simplicity of their creation, linear tetrahedral elements are a common choice for the element shape. Similarly, linear Lagrangian functions (often called the hat functions) are often used to represent the displacement field. The linearity in both shape and basis leads to a major and crucial benefit for dynamic simulations: after the displacement is applied to the rest shape, the resulting mesh remains a piece-wise linear mesh. This is an essential property in order to robustly and efficiently detect and resolve collisions [Wang et al. 2021]. Collisions between arbitrary curved meshes or between linear meshes over curved trajectories are computationally expensive, especially if done in a conservative way [Ferguson et al. 2021].

However, these two choices are restrictive: meshes with curved edges represent shapes, at a given accuracy, with a lower number of elements than linear meshes, especially if tight geometric tolerances are required. Curved meshes are often favored over linear meshes in mechanical engineering [Hughes et al. 2005]. The use of linear bases, especially on simplicial meshes, is problematic as it introduces arbitrary stiffness (a phenomenon known as locking [Schneider et al. 2018]). Additionally, high-order bases are more efficient, in the sense that they provide the same accuracy (compared to a reference solution) as linear bases for a lower running time [Babuška and Guo 1992; Schneider et al. 2022]. Elasto-static problems in computational fabrication (e.g., [Panetta et al. 2015]), mechanics, and biomechanics [Maas et al. 2012] often use high-order bases, but their use for dynamic problems with contact is

very limited or the high-order displacements are ignored for contact purposes.

*Contribution.* We propose a novel elastodynamic formulation supporting both high-order geometry and high-order bases (Figure 1). Our key observation is that a linear transformation of the displacements degrees of freedom leads to linear trajectories of a carefully designed collision proxy. We use this observation to extend the recently proposed Incremental Potential Contact (IPC) formulation, enabling us to use both high-order geometry and high-order bases. Additionally, we can now use arbitrary collision proxies in lieu of the boundary of the FE mesh, a feature that is useful, for example, for the simulation of nearly rigid materials. To evaluate the effectiveness of our approach, we explore its use in graphics applications, where we use the additional flexibility to efficiently simulate complex scenes with a low error tolerance, and we show that our approach can be used to capture complex buckling behaviors with a fraction of the computational cost of traditional approaches. Note that in this work we focus on tetrahedral meshes, but there are no theoretical limitations to applying our method to hexahedral or other polyhedral elements.

*Reproducibility.* To foster further adoption of our method we release an open-source implementation based on PolyFEM [Schneider et al. 2019b] which can be found at polyfem.github.io.

## 2 RELATED WORK

*High-Order Contacts.* Contact between curved geometries has been investigated in multiple communities, as the benefits of *p*-refinement (i.e., refinement of the basis order) for elasticity have been shown to transfer to problems with contact in cases where an analytic solution is known, such as Hertzian contact [Aldakheel et al. 2020; Franke et al. 2010, 2008; Konyukhov and Schweizerhof 2009].

One of the simplest forms of handling contact, penalty methods [Moore and Wilhelms 1988; Terzopoulos et al. 1987] apply penalty force when objects contact and intersect. However, despite their simplicity and computational advantages, it is well known that the behavior of penalty methods strongly depends on the choice of penalty stiffness (and a global and constant in-time choice ensuring stability may not be possible). Li et al. [2020] propose IPC to address these issues, and we choose to use their formulation and benefit from their strong robustness guarantees.

Mortar methods [Belgacem et al. 1998; Hüeber and Wohlmuth 2006; Puso and Laursen 2004] are also a popular choice for contact handling, especially in engineering [Krause and Zulian 2016] and biomechanics [Maas et al. 2012]. Extensions to high-order non-uniform rational B-spline (NURBS) surfaces have also been proposed [Seitz et al. 2016]. Mortar methods require to (a priori) mark the contacting surfaces. A clear limitation of this method is that they cannot handle collisions in regions with more than two contacting surfaces or self-collisions. Li et al. [2020] provide a didactic comparison of the IPC method and one such mortar method ([Krause and Zulian 2016]). They show such methods enforce contact constraints weakly and therefore allow intersections (especially at large timesteps and/or velocities). Nitsche's method is a method for soft Dirichlet boundary conditions (eliminating

the need to tune the penalty stiffness) [Nitsche 1971]. Stenberg [1998] and recent work [Chouly et al. 2022; Gustafsson et al. 2020] extend Nitsche's method to handle contacts through a penalty or mortaring method. While this eliminates the need to tune penalty stiffnesses, these methods still suffer from the same limitations as mortaring methods.

Another way to overcome the challenges with high-order contact is the use of a *third medium* mesh to fill the empty space between objects [Wriggers et al. 2013]. This mesh is handled as a deformable material with carefully specified material properties and internal forces which act in lieu of the contact forces. In this setting, high-order formulations using *p*-refinement have been shown to be very effective [Bog et al. 2015]. Similar methods have been used in graphics (referred to as an "air mesh"), as a replacement for traditional collision detection and response methods [Jiang et al. 2017; Müller et al. 2015]. The challenge for these approaches is the maintenance of a high-quality tetrahedral mesh in the thin contact regions, a problem that is solved in 2D, but still open for tetrahedral meshes.

The detection and response to collisions between spline surfaces are major open problems in isogeometric analysis, where over a hundred papers have been published on this topic (we refer to Temizer et al. [2011] and Cardoso and Adetoro [2017] for an overview). However, automatic mesh generation for isogeometric analysis (IGA) is still an open issue [Schneider et al. 2021], limiting the applicability of these methods to simple geometries manually modeled, and often to surface-only problems.

In comparison, we introduce the first technique using the IPC formulation to solve elastodynamic problems with contact and friction forces on curved meshes using high-order elements. We also show that an automatic high-order meshing and simulation pipeline is possible when our algorithm is paired with [Jiang et al. 2021].

*High-Order Collision Detection.* IPC utilizes continuous collision detection (CCD) to ensure that every step taken is intersection-free. The numerical exactness of CCD can make or break the guarantees provided by the IPC algorithm [Wang et al. 2021]. While several authors have proposed methods for collision detection between curved surfaces and nonlinear trajectories [Ferguson et al. 2021; Kry and Pai 2003; Nelson and Cohen 1998; Nelson et al. 2005; Snyder et al. 1993; Von Herzen et al. 1990], there still does not exist a method that is computationally efficient while being conservative (i.e. never misses collisions). Therefore, we are unable to utilize existing methods and instead, propose a method of coupling linear surface representations with curved volumetric geometry.

*High-Order Bases.* Linear FE bases are overwhelmingly used in graphics applications, as they have the smallest number of degrees of freedom (DOF) per element and are simpler to implement. High-order bases have been shown to be beneficial to animate deformable bodies [Bargteil and Cohen 2014], to accelerate approximate elastic deformations [Mezger et al. 2009], and to compute displacements for embedded deformations [Longva et al. 2020]. Higher-order bases have also been used in meshless methods for improved accuracy and faster convergence [Faure et al. 2011; Martin et al. 2010]. High-order bases are routinely used in engineering analysis [Jameson et al. 2002] where *p*-refinement is often favored over *h*-refinement

(i.e., refinement of the number of elements) as it reduces the geometric discretization error faster and using fewer degrees of freedom [Babuska and Guo 1988; Babuška and Guo 1992; Bassi and Rebay 1997; Luo et al. 2001; Oden 1994].

We propose a method that allows using high-order bases within the IPC framework, thus enabling us to resolve the IPC contact model at a higher efficiency for elastodynamic problems with complex geometry, i.e. we can obtain similar accuracy as with linear bases with a lower computation budget. Additionally, our method allows us to explicitly control the accuracy of the collision approximation by changing the collision mesh sampling (Section 4).

High-order bases can be used as a reduced representation and the high-order displacements can be transferred to higher resolution meshes for visualization purposes [Suwelack et al. 2013]. We use this approach to extend our method to support arbitrary collision proxies, which enables us to utilize our method to accelerate elastodynamic simulations by sacrificing accuracy in the elastic forces.

*Physically-Based Simulation.* There is a large literature on the simulation of deformable and rigid bodies in graphics [Bargteil and Shinar 2018; Kim and Eberle 2022], mechanics, and robotics [Choi et al. 2021]. In particular, a large emphasis is on the modeling of contact and friction forces [Brogliato 1999; Kikuchi and Oden 1988; Stewart 2001; Wriggers 1995].

Longva et al. [2020] propose a method for embedding geometries in coarser FE meshes. By doing so they can reduce the complexity while utilizing higher-order elements to generate accurate elastic deformations. To apply Dirichlet boundary conditions they design the spaces such that they share a common boundary. This scheme, however, cannot capture self-contacts without resorting to using the full mesh. As such they do not consider the handling of contacts. They do, however, suggest a variant of the Mortar method could be future work, but this has known limitations as outlined above. We do not provide a comparison against this method as it does not support contact, and adding contact to it is a major research project on its own, as discussed by the authors.

In our work, we build upon the recently introduced IPC [Li et al. 2020] approach, as it offers higher robustness and automation compared to traditional formulations allowing interpenetrations between objects. We review only papers using the IPC formulation in this section, and we refer to [Li et al. 2020] for a detailed overview of the state of the art.

Li et al. [2020] proposes to use a linear FE method to model the elastic potential, and an interior point formulation to ensure that the entire trajectory is free of collisions. While the approach leads to accurate results when dense meshes are used, the computational cost is high, thus stemming a series of works proposing to use reduced models to accelerate the computation. Li et al. [2021] propose Codimensional IPC (C-IPC), a new formulation for codimensional objects is introduced that optionally avoids using volumetric elements to model thin sheets and rod-like objects. An acceleration of multiple orders of magnitude is possible for specific scenes where the majority of objects are codimensional. Ferguson et al. [2021] propose a formulation of IPC for rigid body dynamics, dramatically

reducing the number of DOF but adding a major cost and complexity to the collision detection stage, as the trajectories spanned by rigid objects are curved.

Longva et al. [2020] demonstrate their ability to approximately model a rigid body using a single stiff element. This idea is further expanded upon by Lan et al. [2022] who propose to relax the rigidity assumption: they use an affine transformation to approximate the rigid ones, thus reducing the problem of collision detection to a much more tractable linear CCD. Massive speedups are possible for rigid scenes, up to three orders of magnitude compared to the original formulation. While these methods provide major acceleration for specific types of scenes, they are not directly usable for scenes with deformable objects.

Lan et al. [2021] proposes to use medial elastics [Lan et al. 2020], a family of reduced models tailored for real-time graphics applications. In their work, the shape is approximated by a medial skeleton which is used to model both the elastic behavior and as a proxy for collision detection. The approach can simulate deformable objects, however, it cannot reproduce a given polyhedral mesh and it is also specialized for medial elasticity simulations.

In our work, we enable the use of high-order meshes and high-order elements in a standard FE framework. Our approach decouples the mesh used to model the elastic potential from the mesh used for the contact and friction potentials, thus providing finer-grained control between efficiency and accuracy.

*Convergence and use of $C^0$ Lagrangian Elements.* Studies compare $C^0$ (p-finite element method (FEM)) and IGA bases' convergence under p-refinement [Sevilla et al. 2011], in the presence of contact [Seitz et al. 2016; Temizer et al. 2011] and in other settings such as electromechanics [Poya et al. 2018]. IGA bases have been shown, in specific problems with simple geometries, to have slightly higher accuracy compared to Lagrangian $C^0$ elements. In this work, we favor Lagrangian $C^0$ elements as IGA meshes are hard to generate for complex geometries and, additionally, some of their benefits are lost when non-regular grid meshes are required to represent complex geometry [Schneider et al. 2019a, 2022]. Our paper does not study the convergence of the method, we leave a convergence (h and p) study as future work jointly with a convergence study for the IPC contact model. Our goal is restricted to show that elastodynamic simulations with high-order geometry and bases are possible on complex geometry and provide a practical speedup over the linear geometry representation and linear bases that are commonly used in graphics applications.
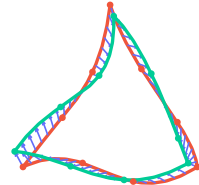
## 3 IPC OVERVIEW

Our approach builds upon the IPC solver introduced in [Li et al. 2020]. In this section, we review the original formulation and introduce the notation.

Li et al. [2020] computes the updated displacements $u^{t+1}$ of the objects at the next time step by solving an *unconstrained* non-linear energy minimization:

$$u^{t+1} = \operatorname*{argmin}_{u} E(u, u^t, v^t) + B(x + u, \hat{d}) + D(x + u, \epsilon_v), \quad (1)$$

where $x$ is the vertex coordinates of the rest position, $u^t$ is the displacement at the current step, $v^t$ the velocities, $E(u, u^t, v^t)$ is a time-stepping incremental potential (IP) [Kane et al. 2000], $B$



$$\left( \sum_{i=1}^{n} u_i^{t+1} \varphi_i \right) - \left( \sum_{i=1}^{n} u_i^t \varphi_i \right)$$
$$= \sum_{i=1}^{n} (u_i^{t+1} - u_i^t) \varphi_i = \sum_{i=1}^{n} \Delta u_i \varphi_i$$

**Figure 2:** *Linearity of displacement update.* **Even with non-linear bases $\varphi_i$, the update to displacement still constitutes a linear combination of nodal displacements. Therefore from a starting position (in red), the update to displacements of any point on the surface (in blue) is linear, and as such we need not use expensive nonlinear CCD.**

is the barrier potential, and $D$ is the lagged dissipative potential for friction [Li et al. 2020]. The user-defined geometric accuracy $\hat{d}$ controls the maximal distance at which the barrier potential will have an effect. Similarly, the smooth friction parameter $\epsilon_v$ controls the smooth transition between static and dynamic friction. We refer to Li et al. [2020] for a complete description of the potentials, as for our purposes we will not need to modify them.

*Solver and Line Search CCD.* The advantage of the IPC formulation is that it is possible to prevent intersections from happening by using a custom Newton solver with a line-search that explicitly checks for collisions using a continuous collision detection algorithm [Provot 1997; Wang et al. 2021], while keeping the overall simulation cost comparable to the more established linear complementarity problem (LCP) based contact solvers [Li et al. 2020].

## 4 METHOD

We introduce an extension of IPC for a curved mesh $\mathcal{M} = (V_{\mathcal{M}}, T_{\mathcal{M}})$ where $V_{\mathcal{M}}$ and $T_{\mathcal{M}}$ are the nodes and volumetric elements of $\mathcal{M}$, respectivly. The formulation reduces to standard IPC when linear meshes and linear bases are used, but other combinations are also possible: for example, it is possible to use high-order bases on standard piece-wise linear meshes, as we demonstrate in Section 5.

We first introduce explicit definitions for functions defined on the volume and the contact surface corresponding to its boundary. Let $f_{\mathcal{M}} \colon \mathcal{M} \to \mathbb{R}^3$ be a volumetric function (in our case the volumetric displacement $u$) defined as

$$f_{\mathcal{M}} = \sum_{i=1}^{n} f_{\mathcal{M}}^i \varphi_{\mathcal{M}}^i, \quad (2)$$

where $\varphi_{\mathcal{M}}^i$ are the $n$ FE bases defined on $\mathcal{M}$ and $f_{\mathcal{M}}^i$ their coefficient.

Similarly on the surface $\mathcal{S} = (V_S, T_S)$ used for collision, with vertices $V_S$ and triangular faces $T_S$, we define $f_S \colon \mathcal{S} \to \mathbb{R}^3$ (in our case the displacement $u$ restricted to the surface) as

$$f_S = \sum_{j=1}^{m} f_S^j \varphi_S^j, \quad (3)$$

where $\varphi_S^j$ are the $m$ FE bases defined on $\mathcal{S}$ and $f_S^j$ their coefficient. We can now rewrite Equation (1) to make explicit that the potential

$E$ depends on $\mathcal{M}$, while $B$ and $D$ only depend on $\mathcal{S}$:

$$u^{t+1} = \underset{u}{\operatorname{argmin}}\, E_{\mathcal{M}}(u, u^t, v^t) + B_{\mathcal{S}}(V_{\mathcal{S}} + \Phi(u), \hat{d})$$
$$+ D_{\mathcal{S}}(V_{\mathcal{S}} + \Phi(u), \epsilon_v), \qquad (4)$$

where $\Phi\colon \Theta_{\mathcal{M}} \to \Theta_{\mathcal{S}}$ is an operator where $\Theta_{\mathcal{M}} = \operatorname{span}\{\varphi_{\mathcal{M}}^i\}$ and $\Theta_{\mathcal{S}} = \operatorname{span}\{\varphi_{\mathcal{S}}^j\}$ that transfers volumetric functions on $\mathcal{M}$ to $\mathcal{S}$. In the context of [Li et al. 2020] (i.e., Equation (1)), $\Phi$ is a restriction of the volumetric function to its surface. While in general, $\Phi$ could be an arbitrary operator, IPC takes advantage of its linearity: if $\Phi$ is linear, then the trajectories of surface vertices in one optimization step of Equation (4) will be linear (Figure 2), and it is thus possible to use standard continuous collision detection methods [Provot 1997; Wang et al. 2021]. If $\Phi$ is nonlinear, for example in the rigid-body formulation introduced by Ferguson et al. [2021], the collision detection becomes considerably more expensive [Lan et al. 2022].

We observe that arbitrary *linear* operators can be used for $\Phi$, and note that increasing the order of the bases used to represent $f_{\mathcal{M}}$ and $f_{\mathcal{S}}$ does not affect the linearity of the operator. An additional advantage of this reformulation is that the space $\Theta_{\mathcal{S}}$ does not have to be a subspace of $\Theta_{\mathcal{M}}$. For example, the collision mesh can be at a much higher resolution than the volumetric mesh used to resolve the elastic forces (Section 5).

We first discuss how to build a linear operator $\Phi$ for high-order meshes, high-order elements, and arbitrary collision proxies, and we postpone the discussion on how to adapt the IPC algorithm to work with arbitrary $\Phi$ to Section 4.2.

## 4.1 Construction of $\Phi$

We present two methods for constructing $\Phi$: upsampling the surface of $\mathcal{M}$ to obtain a dense piecewise linear approximation of its boundary, which we use as $\mathcal{S}$ (Section 4.1.1), or using an arbitrary surface triangle mesh as $\mathcal{S}$ and determining closest point correspondences used to evaluate bases (Section 4.1.2). Our results in Section 5 show a mix of both approaches: Figures 3 to 5, 10, and 11 use an upsampling while Figures 1, 4, 6 to 9, and 11 use an arbitrary triangle mesh proxy.

Since $\Phi$ is a linear operator, a discrete function $f_{\mathcal{M}} \in \Theta_{\mathcal{M}}$ with coefficients $f_{\mathcal{M}}^i$ can be transferred to $f_{\mathcal{S}} \in \Theta_{\mathcal{S}}$ using its $m$ coefficients $f_{\mathcal{S}}^j$ as

$$\mathbf{f}_{\mathcal{S}} = W \mathbf{f}_{\mathcal{M}},$$

where $\mathbf{f}_{\mathcal{M}}$ and $\mathbf{f}_{\mathcal{S}}$ are the stacked coefficients $f_{\mathcal{M}}^i$ and $f_{\mathcal{S}}^j$, respectively. The tetrahedron $t_{\mathcal{M}}^i \in T_{\mathcal{M}}$ of a high-order mesh $\mathcal{M}$ is defined as the image of the *geometric mapping* $g^i$ applied to reference right-angle tetrahedron $\hat{t}$; that is

$$t_{\mathcal{M}}^i = g^i(\hat{t}).$$

On $\mathcal{S}$, the geometric map is a vectorial function and has the same form as Equation (3).

*4.1.1 Upsampled linear boundary.* To construct $\mathcal{S}$ we need to use the *geometric map* to find the initial vertex positions, while to define the operator to transfer functions from the volumetric mesh to $\mathcal{S}$ we will use the *basis functions* of $\mathcal{M}$.

*Vertex Positions.* Every vertex of the piece-wise linear approximation $v_{\mathcal{S}}^j \in V_{\mathcal{S}}$ has coordinates $\hat{v}^j$ in the reference tetrahedron of

---

**Algorithm 1** Construct $\Phi = W \mathbf{f}_{\mathcal{M}}$ for arbitrary triangle mesh

1: $W \leftarrow \mathbf{0}$                                     $\triangleright\; W \in \mathbb{R}^{m \times n}$
2: $\widetilde{\mathcal{M}} \leftarrow \operatorname{linearize}(\mathcal{M}, 4)$    $\triangleright$ 4 linear tetrahedra per curved tet
3: **for** $v_{\mathcal{S}}^j \in V_{\mathcal{S}}$ **do**
4:      $\boxdot \leftarrow \operatorname{inflate}(\operatorname{AABB}(v_{\mathcal{S}}^j), 10^{-3})$
5:      **while** $|\boxdot \cap \widetilde{\mathcal{M}}| < n$ **do**       $\triangleright\; n = 3$ in our examples
6:          $\boxdot \leftarrow \operatorname{inflate}(\boxdot, 10\%)$
7:      **end while**
8:      **for** $t_{\mathcal{M}}^i \in (\boxdot \cap \mathcal{M})$ **do**
9:          $\tilde{b}_i \leftarrow \operatorname{BC}(v_{\mathcal{S}}^j, \operatorname{linearize}(t_{\mathcal{M}}^i))$     $\triangleright$ barycentric coords.
10:          $\hat{v}_i^j \leftarrow \operatorname{argmin}_{\hat{v}} \|g^i(\hat{v}) - v_{\mathcal{S}}^j\|_2^2$   $\triangleright$ L-BFGS with $\hat{v}_0 = \tilde{b}_i$
11:      **end for**
12:      $i^* \leftarrow \operatorname{argmin}_i \|\hat{v}_i^j\|_1$          $\triangleright$ Closest to the interior
13:      $\hat{v}^j = \hat{v}_{i^*}^j$                       $\triangleright$ pre-image of $v_{\mathcal{S}}^j$
14:      $W_{ji} = \varphi_{\mathcal{M}}^i(\hat{v}^j)$
15: **end for**
16: **return** $W$

---

$t_{\mathcal{M}}^i$, so its global coordinates can be computed as

$$v_{\mathcal{S}}^j = g^i(\hat{v}^j),$$

and stacked into the vector $V_{\mathcal{S}}$ used in Equation (4).

*Transfer.* To construct the linear operator $\Phi$ encoded with the matrix $W$ transferring from a higher-order polynomial basis on the boundary of $\mathcal{M}$ to the piecewise linear approximation $\mathcal{S}$, we observe that, since $\mathcal{S}$ is an upsampling of $\mathcal{M}$, we can use $\hat{v}^j$ to directly evaluate the bases of $\mathcal{M}$ (for all non-zero bases) and use them as a weight to transfer the function from $\mathcal{M}$ to $\mathcal{S}$ and define

$$W_{ji} = \varphi_{\mathcal{M}}^i(\hat{v}^j),$$

which is a linear operator, independent of the degree of the basis functions.

*4.1.2 Arbitrary Triangle Mesh Proxy.* The same construction applies to arbitrary mesh proxies (e.g., Figure 1), but we need to compute $\hat{v}^j$ for every vertex. When $\mathcal{M}$ is linear we can simply compute $\hat{v}^j$ as the barycentric coordinates of the closest tetrahedron in $\mathcal{M}$, but when $\mathcal{M}$ is nonlinear we use an optimization to invert $g^i$ [Suwelack et al. 2013]. However, unlike Suwelack et al. [2013], we found that using a normal field to define correspondences is fragile when the surfaces have a very different geometric shape, so we opt for a simpler formulation based on distances.

Algorithm 1 outlines our method for computing $\Phi$ for an arbitrary triangle proxy. Namely, given a volumetric mesh $\mathcal{M}$ and an arbitrary triangle mesh $\mathcal{S}$ we do not have the pre-image under the geometric mapping of the vertices $v_{\mathcal{S}}^j \in V_{\mathcal{S}}$, so we compute one by determining the closest element in $\mathcal{M}$ to $v_{\mathcal{S}}^j$ and use an optimization to compute the inverse geometric mapping to obtain the coordinates $\hat{v}^j$. This procedure only needs to be performed once because $W$ depends only on the rest geometry.

## 4.2 Gradient and Hessian of Surface Terms

Adapting IPC to work with arbitrary linear $\Phi$ mapping requires only changing the *assembly* phase, which requires gradients and Hessian of the surface potentials. Similar to IPC, we use Newton's
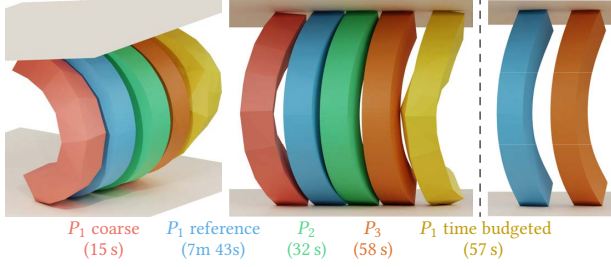
**Figure 3:** *Bending beam.* Squared-section coarse beam pressed by two planes. Linear elements exhibit artificial stiffness as they cannot bend. The reference $P_1$ solution and $P_3$ are rendered in isolation on the right. The results are indistinguishable, but $P_3$ is an order of magnitude faster.

method to minimize the newly formulated potential in Equation (4), and we thus need its gradient and Hessian.

For a surface potential $B_S(V_S + \Phi(u), \hat{d})$ and transfer

$$\Phi(u) = \Phi\left(\sum_{i=1}^{n} u_i \varphi_M^i\right) = \sum_{j=1}^{m} (W\mathbf{u})_j \varphi_S^j,$$

where $\mathbf{u}$ is the *vector* containing all the coefficients $u_i$; we use the definition of $W$ to express the gradient of the barrier (or the friction) potential as

$$\nabla_u B_S(V_S + \Phi(u), \hat{d}) = \nabla_u(V_S + \Phi(u)))^\top \nabla_{S_u} B_S(S_u, \hat{d})$$

$$= \nabla_u(V_S + (W\mathbf{u}))^\top \nabla_{S_u} B_S(S_u, \hat{d}) = W^\top \nabla_{S_u} B_S(S_u, \hat{d}),$$

where $S_u = V_S + \Phi(u)$. The Hessian is computed similarly

$$\nabla_u^2 B_S(V_S + \Phi(u), \hat{d}) = W^\top [\nabla_{S_u}^2 B_S(S_u, \hat{d})] W.$$

The formulas for $\nabla_{S_u} B_S(S_u, \hat{d})$, $\nabla_{S_u} D_S(S_u, \epsilon_v)$, and their Hessians are the same as in [Li et al. 2020], thus requiring minimal modifications to an existing implementation. As in [Li et al. 2020], we mollify the edge-edge distance computation to avoid numerical issues with nearly parallel edges.

## 5 RESULTS

All experiments are run on individual nodes of an high performance computing (HPC) cluster each using two Intel Xeon Platinum 8268 24C 205W 2.9 GHz Processors and 16 threads. All results are generated using the PolyFEM library [Schneider et al. 2019b] coupled with the IPC Toolkit [Ferguson et al. 2020], and use the direct linear solver Pardiso [Alappat et al. 2020; Bollhöfer et al. 2019, 2020]. We use the notation $P_n$ to define the FE bases order (e.g., $P_2$ indicates quadratic Lagrange bases) and all our curved meshes are quartic. All simulation parameters and a summary of the results can be found in Tables 1 and 2.

### 5.1 Test Cases

*Bending beam.* We first showcase the advantages of high-order bases and meshes. Figure 3 shows that linear bases on a coarse mesh introduce artificial stiffness and the result is far from the reference (a dense $P_1$ mesh). As we increase the order, the beam bends more. Using $P_3$ on such a coarse mesh leads to results indistinguishable from the reference at a fraction of the cost. We also compare the
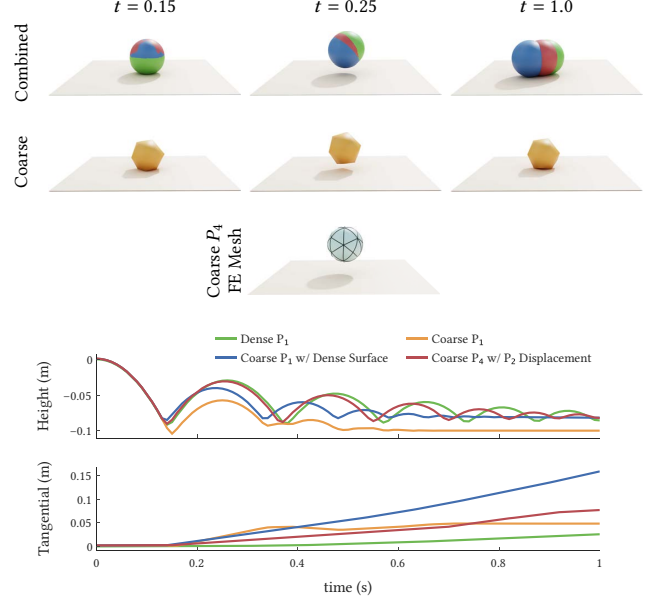


**Figure 4:** *Bouncing ball.* Simulation of a bouncing sphere on a plane. The yellow image and line are the baseline, a coarse linear mesh with linear displacement. The results can be improved using our method and replacing $S$ with a dense sphere in blue. When using a high-order mesh with $P_2$ displacement, red, the results are similar to the dense linear simulation in green.

results of a higher resolution $P_1$ mesh with a limited time budget. That is, the number of elements is chosen to produce a similar running time as the $P_3$ results (1,124 tetrahedra compared to 48 in the coarse version). Even in this case, the differences are obvious and far from the expected results.

*Bouncing ball.* Figure 4 shows the movement of the barycenter of a coarse bouncing sphere on a plane. When using linear bases on the coarse mesh, the ball tips over and starts rolling as the geometry is poorly approximated (yellow line). Replacing the coarse collision mesh using our method (blue line) improves the results for a small cost (125 frames/s versus 83.3 frames/s); however, since the sphere boundary is poorly approximated and the bases are linear, the results are still far from the accurate trajectory (green line). Finally, replacing $M$ with a curved mesh and using $P_2$ bases leads almost to the correct dynamics (red line) while maintaining a real-time simulation (38.4 frames/s). As a reference, the dense $P_1$ linear mesh (green line) runs at 3.9 frames/s.

*Rolling ball.* Figure 7 shows our method is able to maintain purely tangential friction forces on the FE mesh while rolling a ball down a slope. The baseline spherical FE mesh (8.8K $P_1$ tetrahedra) and our method using a cube FE mesh (26 $P_1$ tetrahedra), both using the same collision geometry, produce very similar dynamics, but our method is 7.5× faster. However, while the ball's material is stiff ($E = 10^9$ Pa), it is not rigid, so the baseline model deforms slightly at the point of contact. Our model exhibits extra numerical stiffness from the large linear elements and so deforms less. This results in a
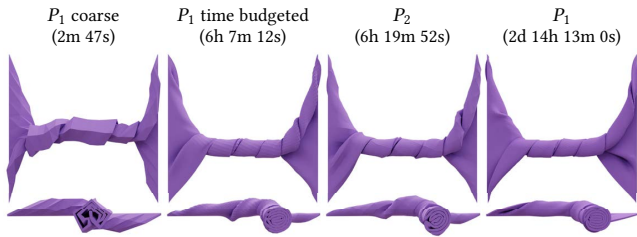
Figure 5: *Mat-twist.* Simulation of twisting for different bases' order and mesh resolutions. The cross-section (bottom row) shows that the coarse linear mesh (left) has huge artifacts. The coarse $P_2$ bases (middle-right) produce smooth results similar to a dense mesh (right) for a tenth of the time. A "time-budgeted" version shows similar results but exhibits checker patterns around the folds.

5% difference on average in the minimum distance which translates to a normal force (and ultimately friction force) that is 2× greater. This inaccuracy is a limitation of using such a course FE mesh within our framework.

## 5.2 Examples

*Mat twist.* We reproduce the mat twist example in [Li et al. 2020] using a thin linear mesh $\mathcal{M}$ with 2K tetrahedra and simulate the self-collisions arising from rotating the two sides using a collision mesh $\mathcal{S}$ with 65K vertices (Figure 5). Simulating this result using standard IPC on the coarse (left) is fast but leads to visible artifacts; by using $P_2$ bases for displacements the results are smooth and the simulation is faster (91 s/frame). For reference, a finer linear solution with more elements, to get a result similar to ours but only using linear elements, requires 230K elements and a runtime 10× higher.

We find a $P_1$ mesh with 51K tetrahedra (25× the number used in the $P_2$ variant) that produces a similar running time. The $P_2$ collision mesh uses 3.5× more triangles leading to 3.1× slower collision detection while the linear solver for the $P_1$ mesh is only 2.2× slower. This results in similar dynamics and final state (see Figure 5) with some notable differences around the folds of the mat.

*Microstructure.* In Figure 10, we simulate the compression of an extremely coarse (6K $P_4$ tetrahedra) curved microstructure mesh from [Jiang et al. 2021]. We upsample its surface to generate a collision mesh with 143K triangles. We demonstrate our method's ability to simulate *anisoparametric* scenarios (i.e., the shape and basis functions differ) by using $P_1$ and $P_2$ bases. In this case, both simulations take a similar amount of time (6h 34m 9s versus 6h 4m 48s).

*Armadillo on a Roller.* In Figure 11, we replicate the armadillo roller from [Verschoor and Jalba 2019] and use fTetWild [Hu et al. 2020] to generate $\mathcal{M}$ with 1.8K tetrahedra (original mesh has 386K). With our method, we combine $\mathcal{M}$ with the original surface with 21K faces with linear element and obtain a speedup of 60× (row*). We used [Jiang et al. 2021] to generate a coarse curved mesh (with only 4.7K tetrahedra) and use an optimization to invert the geometric mapping and simulate the result using $P_2$, this leads to a simulation 30× faster (row†). Finally, we upsampled the surface of
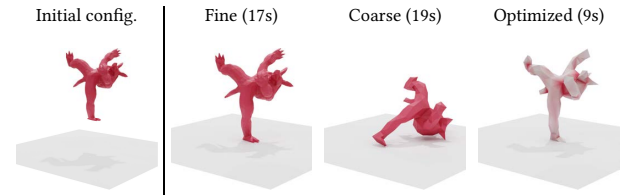


Figure 6: *Balancing armadillo.* We simulate the dancing armadillo from [Prévost et al. 2013] falling on a plane (left). The coarse model (middle) tips over because the center of mass falls outside the foot. We optimize the density (shown in red) to match the input center of mass and the armadillo is balanced (right). Differences in running time can be attributed to the different dynamics (i.e., the coarse model experiences more contacts when it falls over).

the curved mesh to generate a new collision mesh $\mathcal{S}$ with 20K faces, this simulation is only 8× faster (row‡).

*Trash-compactor.* We reproduce the trash compactor from [Li et al. 2020] using a coarse mesh $\mathcal{M}$ with 21K tetrahedra and compress it with five planes. Since the input mesh is already coarse and the models have thin features in the tentacles, we use fTetWild to generate a coarser mesh with 3.5K tetrahedra. Using this mesh with $P_1$ displacements while using the same surface mesh for collisions provides a 2.5× speedup. Since both coarse and input meshes have similar resolution, using $P_2$ leads to a more accurate but much slower (around 10×) result as the number of DOF for $P_2$ is similar to the denser mesh but with 5× the number of surface triangles.

## 5.3 Extreme coarsening

*Nut and Bolt.* As mentioned in Section 4, our method can be used with linear meshes and linear bases. This is best suited to stiff objects where the deformation is minimal. Figure 8 shows an example of a nut sliding inside a bolt, since both materials are stiff ($E = 200$ GPa), we coarsen $\mathcal{M}$ using fTetWild [Hu et al. 2020] from 6K tetrahedra and 1.7K vertices to 492 and 186, respectively. This change allows our method to be twice as fast without visible differences.

*Balancing Armadillo.* When generating a coarse mesh $\mathcal{M}$ the center of mass and mass of the object might change dramatically. Figure 6 shows that the coarse mesh cannot balance anymore as the center of mass is outside the contact area. To prevent this artifact, similarly to [Prévost et al. 2013], we modify the density (in red in the third figure) of the material to move the center of mass.

## 6 CONCLUDING REMARKS

We introduce a robust and efficient simulator for deformable objects with contact supporting high-order meshes and high-order bases to simulate geometrically complex scenes. We show that there are major computational advantages in increasing the order of the geometric map and bases and that they can be used in the IPC formulation with modest code changes.

*Limitations.* At a high level, we are proposing to use $p$-refinement for elasticity, coupled with $h$-refinement approach for contacts, to sidestep the high computational cost of curved continuous collision

SIGGRAPH '23 Conference Proceedings, August 06–10, 2023, Los Angeles, CA, USA        Zachary Ferguson, Pranav Jain, Denis Zorin, Teseo Schneider, and Daniele Panozzo

detection. The downside of our approach is that our contact surface is still an approximation of the curved geometry, and while we can reduce the error by further refinement, we cannot reduce it to zero. While for graphics applications this is an acceptable compromise, as the scene we use for collision is guaranteed to be collision-free and we inherit the robustness properties of the original IPC formulation, there could be engineering applications where it is important to model a high-order surface *exactly*. In this case, our approach could not be used as we might miss the collisions of the curved FE mesh.

A second limitation of our approach is that the definition of a robust, guaranteed positivity check for high-order elements is still an open research problem [Johnen et al. 2013]. In our implementation, we check positivity only at the quadrature points, which is a reasonable approximation but might still lead to unphysical results as the element might have a negative determinant in other interior points.

While our method for mapping between an arbitrary triangle mesh proxy and the curved tetrahedral mesh works well enough for the examples shown in this paper, it is not a robust implementation, as the closest point query can lead to wrong correspondences. In the future, it will be interesting to explore the use of bijective maps between the two geometries to avoid this issue (for example by using the work of Jiang et al. [2020]).

Our choice of $\Phi$ is not unique as there are a large number of basis functions to choose from. We explored other options such as mean value coordinates and linearized L2-projection, but we found their global mappings produce dense weight matrices. This results in slower running times with only minor quality improvements. A future direction might be the exploration of more localized operators such as bounded bi-harmonic weights [Jacobson et al. 2011].

*Future Work.* Beyond these limitations, we see three major avenues for future work: (1) existing curved mesh generators are still not as reliable in producing high-quality meshes as their linear counterparts: more work is needed in this direction, and our approach can be used as a testbed for evaluating the benefits curved mesh provides in the context of elastodynamic simulations, (2) our approach could be modified to work with hexahedral elements, spline bases, and isogeometric analysis simulation frameworks, and (3) we speculate that integrating our approach with high-order time integrators could provide additional benefits for further reducing numerical damping and we believe this is a promising direction for a future study.

Our approach is a first step toward the introduction of high-order meshes and high-order FEM in elastodynamic simulation with the IPC contact model, and we believe that our reference implementation will reduce the entry barrier for the use of these approaches in industry and academia.

## ACKNOWLEDGMENTS

This work was supported in part through the NYU IT High Performance Computing resources, services, and staff expertise. This work was also partially supported by the NSF CAREER award under Grant No. 1652515, the NSF grants OAC-1835712, OIA-1937043, CHS-1908767, CHS-1901091, NSERC DGECR-2021-00461 and RG-PIN 2021-03707, a Sloan Fellowship, a gift from Adobe Research and a gift from Advanced Micro Devices, Inc.</ant|cr

## REFERENCES

Christie Alappat, Achim Basermann, Alan R. Bishop, Holger Fehske, Georg Hager, Olaf Schenk, Jonas Thies, and Gerhard Wellein. 2020. A Recursive Algebraic Coloring Technique for Hardware-Efficient Symmetric Sparse Matrix-Vector Multiplication. *ACM Transactions on Parallel Computing* 7, 3, Article 19 (June 2020), 37 pages.

Fadi Aldakheel, Blaž Hudobivnik, Edoardo Artioli, Lourenço Beirão da Veiga, and Peter Wriggers. 2020. Curvilinear virtual elements for contact mechanics. *Computer Methods in Applied Mechanics and Engineering* 372 (2020), 113394.

I. Babuska and B. Q. Guo. 1988. The h-p Version of the Finite Element Method for Domains with Curved Boundaries. *SIAM J. Numer. Anal.* 25, 4 (1988), 837–861.

I. Babuška and B.Q. Guo. 1992. The h, p and h-p version of the finite element method; basis theory and applications. *Advances in Engineering Software* 15, 3 (1992), 159–174.

Adam Bargteil and Tamar Shinar. 2018. An Introduction to Physics-Based Animation. In *ACM SIGGRAPH 2018 Courses* (Vancouver, British Columbia, Canada). ACM, New York, NY, Article 6, 1 pages.

Adam W Bargteil and Elaine Cohen. 2014. Animation of deformable bodies with quadratic Bézier finite elements. *ACM Transactions on Graphics* 33, 3 (2014), 27.

F. Bassi and S. Rebay. 1997. High-Order Accurate Discontinuous Finite Element Solution of the 2D Euler Equations. *J. Comput. Phys.* 138, 2 (1997), 251–285.

F.B. Belgacem, P. Hild, and P. Laborde. 1998. The mortar finite element method for contact problems. *Mathematical and Computer Modelling* 28, 4 (1998), 263–271. Recent Advances in Contact Mechanics.

Tino Bog, Nils Zander, Stefan Kollmannsberger, and Ernst Rank. 2015. Normal contact with high order finite elements and a fictitious contact material. *Computers & Mathematics with Applications* 70, 7 (2015), 1370–1390. High-Order Finite Element and Isogeometric Methods.

Matthias Bollhöfer, Aryan Eftekhari, Simon Scheidegger, and Olaf Schenk. 2019. Large-scale Sparse Inverse Covariance Matrix Estimation. *SIAM Journal on Scientific Computing* 41, 1 (2019), 380–401.

Matthias Bollhöfer, Olaf Schenk, Radim Janalik, Steve Hamm, and Kiran Gullapalli. 2020. State-of-the-Art Sparse Direct Solvers. *Parallel Algorithms in Computational Science and Engineering* (2020), 3–33.

Bernard Brogliato. 1999. *Nonsmooth Mechanics*. Springer-Verlag.

R. P. R. Cardoso and O. B. Adetoro. 2017. On contact modelling in isogeometric analysis. *European Journal of Computational Mechanics* 26, 5-6 (2017), 443–472.

HeeSun Choi, Cindy Crump, Christian Duriez, Asher Elmquist, Gregory Hager, David Han, Frank Hearl, Jessica Hodgins, Abhinandan Jain, Frederick Leve, Chen Li, Franziska Meier, Dan Negrut, Ludovic Righetti, Alberto Rodriguez, Jie Tan, and Jeff Trinkle. 2021. On the use of simulation in robotics: Opportunities, challenges, and suggestions for moving forward. *Proceedings of the National Academy of Sciences* 118, 1 (2021).

Franz Chouly, Patrick Hild, Vanessa Lleras, and Yves Renard. 2022. Nitsche method for contact with Coulomb friction: Existence results for the static and dynamic finite element formulations. *J. Comput. Appl. Math.* 416 (2022), 114557.

François Faure, Benjamin Gilles, Guillaume Bousquet, and Dinesh K. Pai. 2011. Sparse Meshless Models of Complex Deformable Solids. *ACM Transactions on Graphics* 30, 4, Article 73 (July 2011), 10 pages.

Zachary Ferguson et al. 2020. IPC Toolkit. https://ipc-sim.github.io/ipc-toolkit/. https://ipc-sim.github.io/ipc-toolkit/

Zachary Ferguson, Minchen Li, Teseo Schneider, Francisca Gil-Ureta, Timothy Langlois, Chenfanfu Jiang, Denis Zorin, Danny M. Kaufman, and Daniele Panozzo. 2021. Intersection-Free Rigid Body Dynamics. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 40, 4, Article 183 (July 2021), 16 pages.

David Franke, A. Düster, V. Nübel, and E. Rank. 2010. A comparison of the h-, p-, hp-, and rp-version of the FEM for the solution of the 2D Hertzian contact problem. *Computational Mechanics* 45, 5 (April 2010), 513–522.

David Franke, Alexander Düster, and Ernst Rank. 2008. The p-version of the FEM for computational contact mechanics. *Pamm* 8, 1 (2008), 10271–10272.

Tom Gustafsson, Rolf Stenberg, and Juha Videman. 2020. On Nitsche's Method for Elastic Contact Problems. *SIAM Journal on Scientific Computing* 42, 2 (2020), B425–B446.

Yixin Hu, Teseo Schneider, Bolun Wang, Denis Zorin, and Daniele Panozzo. 2020. Fast Tetrahedral Meshing in the Wild. *ACM Transactions on Graphics* 39, 4, Article 117 (July 2020), 18 pages.

S. Hüeber and B.I. Wohlmuth. 2006. *Mortar methods for contact problems*. Springer Berlin Heidelberg, Berlin, Heidelberg, 39–47.

T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. 2005. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering* 194, 39 (2005), 4135–4195.

Alec Jacobson, Ilya Baran, Jovan Popović, and Olga Sorkine. 2011. Bounded Biharmonic Weights for Real-Time Deformation. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 30, 4 (2011), 78:1–78:8.

A. Jameson, J. Alonso, and M. McMullen. 2002. Application of a non-linear frequency domain solver to the Euler and Navier-Stokes equations. In *40th AIAA Aerospace Sciences Meeting & Exhibit*.

Zhongshi Jiang, Scott Schaefer, and Daniele Panozzo. 2017. Simplicial Complex Augmentation Framework for Bijective Maps. *ACM Transactions on Graphics* 36, 6, Article 186 (Nov. 2017), 9 pages.

Zhongshi Jiang, Teseo Schneider, Denis Zorin, and Daniele Panozzo. 2020. Bijective Projection in a Shell. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 39, 6, Article 247 (Nov. 2020), 18 pages.

Zhongshi Jiang, Ziyi Zhang, Yixin Hu, Teseo Schneider, Denis Zorin, and Daniele Panozzo. 2021. Bijective and Coarse High-Order Tetrahedral Meshes. *ACM Transactions on Graphics* 40, 4, Article 157 (July 2021), 16 pages.

Amaury Johnen, J-F Remacle, and Christophe Geuzaine. 2013. Geometrical validity of curvilinear finite elements. *J. Comput. Phys.* 233 (2013), 359–372.

Couro Kane, Jerrold E Marsden, Michael Ortiz, and Matthew West. 2000. Variational integrators and the Newmark algorithm for conservative and dissipative mechanical systems. *Internat. J. Numer. Methods Engrg.* 49, 10 (Dec. 2000), 1295–1325.

Noboru Kikuchi and John Tinsley Oden. 1988. *Contact Problems in Elasticity: A Study of Variational Inequalities and Finite Element Methods*. SIAM Studies in App. and Numer. Math., Vol. 8. Society for Industrial and Applied Mathematics.

Theodore Kim and David Eberle. 2022. Dynamic Deformables: Implementation and Production Practicalities (Now with Code!). In *ACM SIGGRAPH 2022 Courses* (Vancouver, British Columbia, Canada). ACM, New York, NY, Article 7, 259 pages.

Alexander Konyukhov and Karl Schweizerhof. 2009. Incorporation of contact for high-order finite elements in covariant form. *Computer Methods in Applied Mechanics and Engineering* 198, 13 (2009), 1213–1223.

Rolf Krause and Patrick. Zulian. 2016. A Parallel Approach to the Variational Transfer of Discrete Fields between Arbitrarily Distributed Unstructured Finite Element Meshes. *SIAM Journal on Scientific Computing* 38, 3 (2016).

Paul G Kry and Dinesh K Pai. 2003. Continuous contact simulation for smooth surfaces. *ACM Transactions on Graphics* 22, 1 (2003), 106–129.

Lei Lan, Danny M. Kaufman, Minchen Li, Chenfanfu Jiang, and Yin Yang. 2022. Affine Body Dynamics: Fast, Stable and Intersection-Free Simulation of Stiff Materials. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 41, 4, Article 67 (July 2022), 14 pages.

Lei Lan, Ran Luo, Marco Fratarcangeli, Weiwei Xu, Huamin Wang, Xiaohu Guo, Junfeng Yao, and Yin Yang. 2020. Medial Elastics: Efficient and Collision-Ready Deformation via Medial Axis Transform. *ACM Transactions on Graphics* 39, 3, Article 20 (April 2020), 17 pages.

Lei Lan, Yin Yang, Danny Kaufman, Junfeng Yao, Minchen Li, and Chenfanfu Jiang. 2021. Medial IPC: Accelerated Incremental Potential Contact with Medial Elastics. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 40, 4, Article 158 (July 2021), 16 pages.

Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, and Danny M. Kaufman. 2020. Incremental Potential Contact: Intersection- and Inversion-free Large Deformation Dynamics. *ACM Transactions on Graphics* 39, 4, Article 49 (July 2020), 20 pages.

Minchen Li, Danny M. Kaufman, and Chenfanfu Jiang. 2021. Codimensional Incremental Potential Contact. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 40, 4, Article 170 (2021).

Andreas Longva, Fabian Löschner, Tassilo Kugelstadt, José Antonio Fernández-Fernández, and Jan Bender. 2020. Higher-Order Finite Elements for Embedded Simulation. *ACM Transactions on Graphics* 39, 6, Article 181 (Nov. 2020), 14 pages.

Xiaojuan Luo, Mark S Shephard, and Jean-Francois Remacle. 2001. The influence of geometric approximation on the accuracy of high order methods. *Rensselaer SCOREC report* 1 (2001).

Steve A. Maas, Benjamin J. Ellis, Gerard A. Ateshian, and Jeffrey A. Weiss. 2012. FEBio: Finite Elements for Biomechanics. *Journal of Biomechanical Engineering* 134, 1 (Feb. 2012).

Sebastian Martin, Peter Kaufmann, Mario Botsch, Eitan Grinspun, and Markus Gross. 2010. Unified Simulation of Elastic Rods, Shells, and Solids. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 29, 3 (2010), 39:1–39:10.

Johannes Mezger, Bernhard Thomaszewski, Simon Pabst, and Wolfgang Straßer. 2009. Interactive physically-based shape editing. *Computer Aided Geometric Design* 26, 6 (2009), 680–694. Solid and Physical Modeling 2008.

Matthew Moore and Jane Wilhelms. 1988. Collision Detection and Response for Computer Animation. *Computer Graphics (Proceedings of SIGGRAPH)* 22, 4 (June 1988), 289–298.

Matthias Müller, Nuttapong Chentanez, Tae-Yong Kim, and Miles Macklin. 2015. Air Meshes for Robust Collision Handling. *ACM Transactions on Graphics* 34, 4, Article 133 (July 2015), 9 pages.

Donald D Nelson and Elaine Cohen. 1998. User interaction with CAD models with nonholonomic parametric surface constraints. In *ASME International Mechanical Engineering Congress and Exposition*, Vol. 15861. American Society of Mechanical Engineers, 235–242.

Donald D. Nelson, David E. Johnson, and Elaine Cohen. 2005. Haptic Rendering of Surface-to-Surface Sculpted Model Interaction. In *ACM SIGGRAPH 2005 Courses* (Los Angeles, California). ACM, New York, NY, 97–es.

Johannes C. C. Nitsche. 1971. Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg* 36 (1971), 9–15.

J. Tinsley Oden. 1994. Optimal h-p finite element methods. *Computer Methods in Applied Mechanics and Engineering* 112, 1 (1994), 309–331.

Julian Panetta, Qingnan Zhou, Luigi Malomo, Nico Pietroni, Paolo Cignoni, and Denis Zorin. 2015. Elastic Textures for Additive Fabrication. *ACM Transactions on Graphics* 34, 4, Article 135 (July 2015), 12 pages.

Roman Poya, Antonio J. Gil, Rogelio Ortigosa, Ruben Sevilla, Javier Bonet, and Wolfgang A. Wall. 2018. A curvilinear high order finite element framework for electromechanics: From linearised electro-elasticity to massively deformable dielectric elastomers. *Computer Methods in Applied Mechanics and Engineering* 329 (2018), 75–117.

Romain Prévost, Emily Whiting, Sylvain Lefebvre, and Olga Sorkine-Hornung. 2013. Make It Stand: Balancing Shapes for 3D Fabrication. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 32, 4 (2013), 81:1–81:10.

Xavier Provot. 1997. Collision and Self-Collision Handling in Cloth Model Dedicated to Design Garments. In *Computer Animation and Simulation*. Springer, 177–189.

Michael A Puso and Tod A Laursen. 2004. A mortar segment-to-segment contact method for large deformation solid mechanics. *Computer Methods in Applied Mechanics and Engineering* 193, 6-8 (2004), 601–629.

Teseo Schneider, Jérémie Dumas, Xifeng Gao, Mario Botsch, Daniele Panozzo, and Denis Zorin. 2019a. Poly-Spline Finite-Element Method. *ACM Transactions on Graphics* 38, 3, Article 19 (March 2019), 16 pages.

Teseo Schneider, Jérémie Dumas, Xifeng Gao, Denis Zorin, and Daniele Panozzo. 2019b. PolyFEM. https://polyfem.github.io/.

Teseo Schneider, Yixin Hu, Jérémie Dumas, Xifeng Gao, Daniele Panozzo, and Denis Zorin. 2018. Decoupling Simulation Accuracy from Mesh Quality. *ACM Transactions on Graphics* 37, 6 (Oct. 2018).

Teseo Schneider, Yixin Hu, Xifeng Gao, Jérémie Dumas, Denis Zorin, and Daniele Panozzo. 2022. A Large-Scale Comparison of Tetrahedral and Hexahedral Elements for Solving Elliptic PDEs with the Finite Element Method. *ACM Transactions on Graphics* 41, 3, Article 23 (March 2022), 14 pages.

Teseo Schneider, Daniele Panozzo, and Xianlian Zhou. 2021. Isogeometric high order mesh generation. *Computer Methods in Applied Mechanics and Engineering* 386 (2021), 114104.

Alexander Seitz, Philipp Farah, Johannes Kremheller, Barbara I. Wohlmuth, Wolfgang A. Wall, and Alexander Popp. 2016. Isogeometric dual mortar methods for computational contact mechanics. *Computer Methods in Applied Mechanics and Engineering* 301 (2016), 259–280.

Ruben Sevilla, Sonia Fernández-Méndez, and Antonio Huerta. 2011. Comparison of high-order curved finite elements. *Internat. J. Numer. Methods Engrg.* 87, 8 (2011), 719–734.

John M. Snyder, Adam R. Woodbury, Kurt Fleischer, Bena Currin, and Alan H. Barr. 1993. Interval Methods for Multi-Point Collisions between Time-Dependent Curved Surfaces, In Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques (Anaheim, CA). *Annual Conference Series (Proceedings of SIGGRAPH)*, 321–334.

Rolf Stenberg. 1998. Mortaring by a method of J. A. Nitsche. *Computational Mechanics* (Jan. 1998).

David E Stewart. 2001. Finite-dimensional contact mechanics. *Philosophical Transactions of the Royal Society A* 359 (2001), 2467–2482.

Stefan Suwelack, Dimitar Lukarski, Vincent Heuveline, Rüdiger Dillmann, and Stefanie Speidel. 2013. Accurate Surface Embedding for Higher Order Finite Elements. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Anaheim, California) *(SCA '13)*. ACM, New York, NY, 187–192.

I. Temizer, P. Wriggers, and T.J.R. Hughes. 2011. Contact treatment in isogeometric analysis with NURBS. *Computer Methods in Applied Mechanics and Engineering* 200, 9 (2011), 1100–1112.

Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. 1987. Elastically Deformable Models. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '87)*. ACM, New York, NY, 205–214.

Mickeal Verschoor and Andrei C Jalba. 2019. Efficient and accurate collision response for elastically deformable models. *ACM Transactions on Graphics* 38, 2, Article 17 (March 2019), 20 pages.

Brian Von Herzen, Alan H. Barr, and Harold R. Zatz. 1990. Geometric Collisions for Time-Dependent Parametric Surfaces. *Computer Graphics (Proceedings of SIGGRAPH)* 24, 4 (Sept. 1990), 39–48.

Bolun Wang, Zachary Ferguson, Teseo Schneider, Xin Jiang, Marco Attene, and Daniele Panozzo. 2021. A Large Scale Benchmark and an Inclusion-Based Algorithm for Continuous Collision Detection. *ACM Transactions on Graphics* 40, 5, Article 188 (Oct. 2021), 16 pages.

Peter Wriggers. 1995. Finite Element Algorithms for Contact Problems. *Archives of Computational Methods in Engineering* 2 (Dec. 1995), 1–49.

P. Wriggers, J. Schröder, and A. Schwarz. 2013. A finite element method for contact using a third medium. *Computational Mechanics* 52, 4 (Oct. 2013), 837–847.
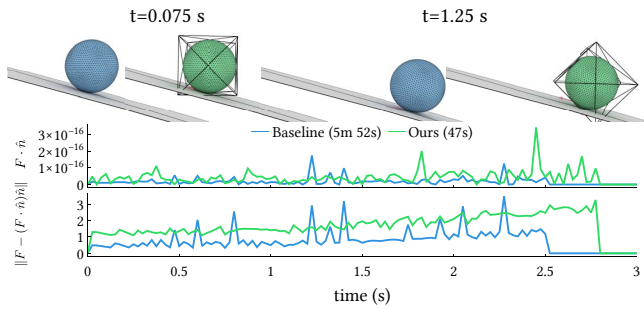
**Figure 7: *Rolling-ball.*** We demonstrate a ball rolling down a slope, while maintaining non-slip rolling contact, produces purely tangential friction forces on the FEM mesh. Our method uses a symmetric cube mesh (black wireframe) as the FE mesh and a high-resolution sphere (green) as the collision mesh. The friction forces on the FE mesh are shown as pink arrows. We plot the out-of-plane friction force ($F \cdot \hat{n}$) and norm of the in-plane friction force ($\|F - (F \cdot \hat{n})\hat{n}\|$). Compared to a high-resolution baseline, the out-of-plane error shows negligible differences but the in-plane force is around $2\times$ greater. This is due to the increased numerical stiffness of our course mesh leading to less localized deformation, smaller distances, and, ultimately, a larger normal force.
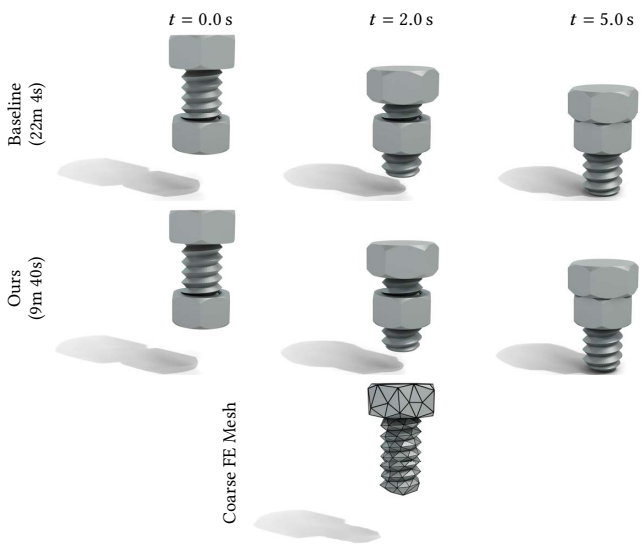


Model ©YSoft be3D under CC BY-SA 3.0.

**Figure 8: *Nut-and-bolt.*** Simulation of a bolt rotating into a bolt under gravity. Directly meshing the input mesh (top) generate similar results as using our method with a coarse simulation mesh (right).
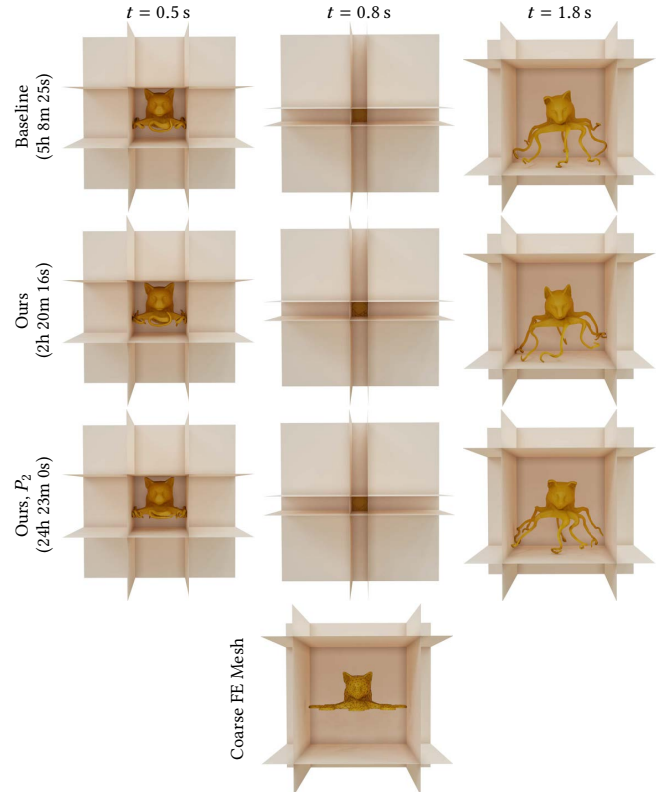


Model ©Brian Enigma under CC BY-SA 3.0.

**Figure 9: *Trash compactor.*** The Octocat model is compressed by five planes. Using the original input mesh (top) is two times slower than using our method with linear elements (middle). Since we cannot coarsen the input too much without losing the tentacles, using $P_2$ leads to longer running times and similar results (bottom).
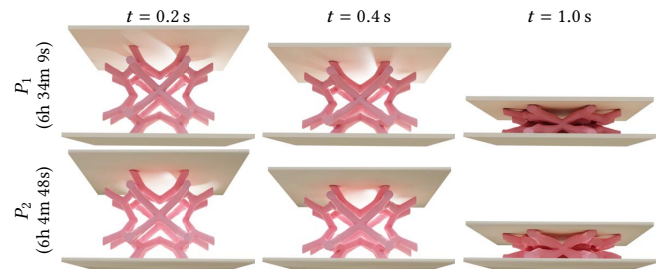


**Figure 10: *Microstructure.*** Compression of a curved microstructure using linear and quadratic bases. While the choice of bases only leads to marginal running time savings, it demonstrates our method's ability to simulate anisoparametric scenarios where the $P_4$ shape functions differ from the $P_1/P_2$ bases.
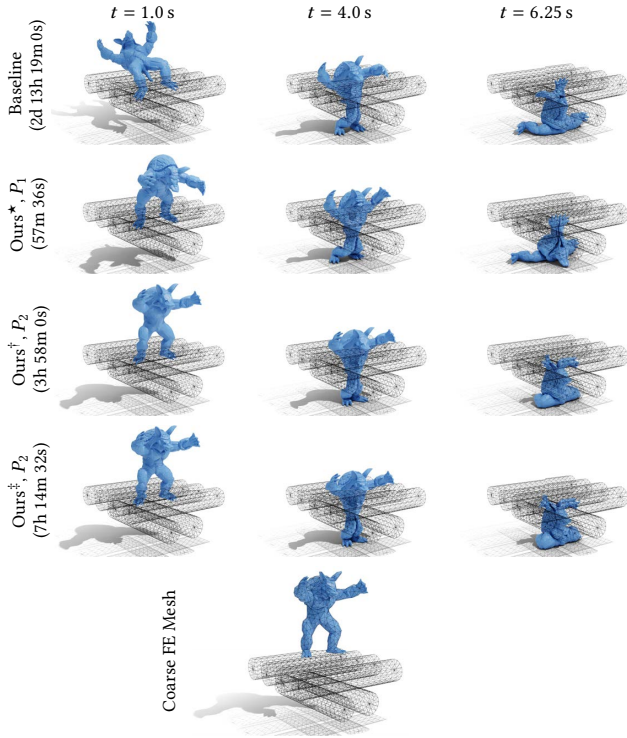
**Figure 11:** *Armadillo-rollers.* Armadillo roller simulation for the different variants of our method. Ours$^\star$ uses a coarse linear mesh with linear displacement and the original geometry for the collision. Ours$^\dagger$ uses a curved mesh with $P_2$ displacement and an upsampled geometry for the collision. Ours$^\ddagger$ uses a curved mesh with $P_2$ displacement and the original geometry for the collision.

**Table 1: Simulation parameters used in the results. For each example, we report the time step size ($h$), density ($\rho$ with $\star$ indicating multi-density), Young's modulus ($E$), Poisson ratio ($\nu$), barrier activation distance ($\hat{d}$), coefficient of friction ($\mu$), friction smoothing parameter ($\epsilon_v$), maximum friction iterations, and Newton tolerance. For all examples, we use implicit Euler time integration and the Neo-Hookean material model.**

| Scene | $h$ (s) | $\rho$ (kg/m$^3$), $E$ (Pa), $\nu$ | $\hat{d}$ (m) | $\mu, \epsilon_v$ (m/s), friction iters. | Newton tol. (m) |
|---|---|---|---|---|---|
| Armadillo-rollers (Figures 1 and 11) | 0.025 | 1e3, 5e5, 0.2 | 1e-3 | 0.5, 1e-3, 1 | 1e-3 |
| Bending beam (Figure 3) | 0.1 | 1e3, 1e7, 0.4 | 1e-3 | 0.5, 1e-3, 10 | 1e-5 |
| Bouncing ball (Figure 4) | 0.001 | 700, 5.91e5, 0.45 | 1e-3 | 0.2, 1e-3, 1 | 1e-12 |
| Mat-twist (Figure 5) | 0.04 | 1e3, 2e4, 0.4 | 1e-3 | - | 1e-5 |
| Balancing armadillo (Figure 6) | 0.1 | 1e3$^\star$, 1e11, 0.2 | 1e-5 | 0.1, 1e-3, 20 | 1e-5 |
| Rolling ball (Figure 7) | 0.025 | 1e3, 1e9, 0.4 | 1e-3 | 1.0, 1e-3, $\infty$ | 1e-5 |
| Nut-and-bolt (Figure 8) | 0.01 | 8050, 2e11, 0.28 | 1e-4 | - | 1e-5 |
| Trash-compactor (Figure 9) | 0.01 | 1e3, 1e4, 0.4 | 1e-3 | - | 1e-5 |
| Microstructure (Figure 10) | 0.01 | 1030, 6e5, 0.48 | 1e-5 | 0.3, 1e-3, 1 | 1e-4 |

**Table 2: Summary of results shown in Section 5. For each example, we report the number of tetrahedra (#T) used for elasticity, the number of surface triangles (#F) used for collision processing, and the total running time of the simulation. Names correspond to the same given in each figure.**

| Scene | | #T | #F | Running Time |
|---|---|---|---|---|
| Armadillo-rollers (Figure 11) | Baseline | 386K | 24K | 2d 13h 19m 00s |
| | Ours$^\star$, $P_1$ | 1.8K | 24K | 57m 36s |
| | Ours$^\dagger$, $P_2$ | 4.7K | 23K | 3h 58m 00s |
| | Ours$^\ddagger$, $P_2$ | 4.7K | 24K | 7h 14m 32s |
| Bending beam (Figure 3) | $P_1$ coarse | 48 | 72 | 15s |
| | $P_1$ reference | 25K | 4.4K | 7m 43s |
| | $P_2$ | 48 | 5.5K | 32s |
| | $P_3$ | 48 | 5.5K | 58s |
| | $P_1$ time budgeted | 1.1K | 2.8K | 57s |
| Bouncing ball (Figure 4) | Dense $P_1$ | 8.8K | 5.1K | 4m 16s |
| | Coarse $P_1$ | 30 | 32 | 8s |
| | Dense Surface | 30 | 2.4K | 12s |
| | $P_4$ | 30 | 2.4K | 26s |
| Mat-twist (Figure 5) | $P_1$ coarse | 2.2K | 1.6K | 2m 47s |
| | $P_1$ time budgeted | 54K | 37K | 6h 7m 12s |
| | $P_2$ | 2.2K | 129K | 6h 19m 52s |
| | $P_1$ | 230K | 141K | 2d 14h 13m 00s |
| Balancing armadillo (Figure 6) | Fine | 5.9K | 3.7K | 17s |
| | Coarse | 585 | 486 | 19s |
| | Optimized | 585 | 486 | 9s |
| Rolling ball (Figure 7) | Baseline | 8.8K | 5.1K | 5m 52s |
| | Ours | 26 | 5.1K | 47s |
| Nut and bolt (Figure 8) | Baseline | 6.1K | 5.2K | 22m 04s |
| | Ours | 492 | 5.2K | 9m 40s |
| Trash-compactor (Figure 9) | Baseline | 21K | 8.6K | 5h 08m 25s |
| | Ours | 3.5K | 8.5K | 2h 20m 16s |
| | Ours, $P_2$ | 3.5K | 41K | 24h 23m 00s |
| Microstructure (Figure 10) | $P_1$ | 6.4K | 143K | 6h 34m 09s |
| | $P_2$ | 6.4K | 143K | 6h 04m 48s |