Determining Contact Data for Rigid Body Dynamics with Convex Polyhedral Geometries

Bjoern Cheng Yi[1] and Evan M. Drumwright[2]

***Abstract*— Dynamic simulation of multi-rigid bodies is a key tool in robotics. Over the past two decades, such simulations have moved from modeling purely motion to simulating contact interactions like locomotion and grasping. Contact data—points of contact, surface normals, and signed distance—have proven straightforward to compute for multi-bodies modeled using primitive shapes, but there is no accepted procedure for determining such data for bodies with convex polyhedral bodies. This paper surveys the algorithmic state of the art that can be used to determine such data, paying special attention to numerical issues.**

I. INTRODUCTION

Multi-rigid body simulation is a key tool used for modeling the behavior of many robotic systems. Importing kinematic and dynamic parameters is somewhat tedious, but can be accomplished for, e.g., typical legged and manipulator robots, within a few hours. Importing geometry for governing contact interactions is much more challenging.

The dominant representation is that of polygon-based geometries, which are ubiquitous in geometric modeling for both computer animation and computer aided engineering. In the former context, triangle meshes, also known as "triangle soups", provide the ability to approximate shapes, both solid and otherwise, to geometric precision that scales with the number of triangles. In the latter context, geometries comprised of interconnected tetrahedra or hexahedra, among other possible polyhedra, are used to compute stresses on materials and predict their deformation; these collections of interconnected polyhedra are readily exported into a boundary representation for a rigid link. Since polygon-based geometries are supported by many 3D modeling and engineering tools and are sufficiently flexible to approximate any shape, effective incorporation of this representation into rigid body simulations is critical.

The community of physics-based simulation developers focused on computer games has developed many approaches for working with polygon-based geometries. These approaches typically consider polyhedra and triangle mesh representations separately, for good reason: since triangle meshes need not describe a solid, existing rigid body simulation techniques are generally unable to correct the inevitable interpenetration in a plausible manner (even as the integration step tends to zero). In any case, we characterize without prejudice that community's approaches as "better always fast than always correct" (it is acceptable for game developers to implement ad hoc solutions to work around artifacts). *This paper takes the alternative approach: the focus is on correct, numerically robust techniques.* Speed is a secondary priority. This paper works to define "correctness" since the use of both floating point arithmetic (which can admit interpenetration for bodies in kissing configurations even when constraints are solved to numerical precision) and popular and fast rigid body dynamics time stepping techniques (which explicitly allow interpenetration) preclude a simple definition.

We focus on rigid body dynamics and consider both rigid and quasi-rigid contact. Multiple quasi-rigid contact models exist, but the one that we consider assumes that a (typically) thin compliant layer exists around *one* rigid body. We have found anecdotally that this model provides advantages of both rigid contact models (better modeling of sticking friction, lower formulation stiffness) and other compliant models (avoiding the need to compute the time of contact).

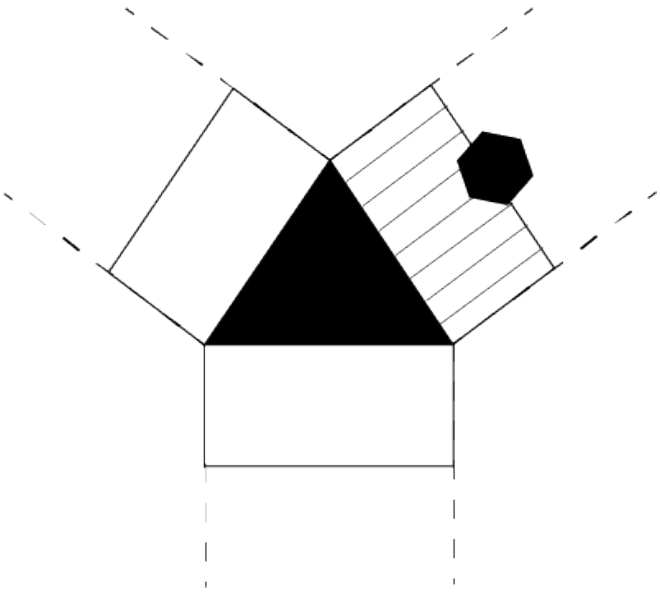


Fig. 1. Depictions of two rigid bodies (the hexagon and the triangle); contact with the triangle is modeled as quasi-rigid which is comprised of a rigid core (the black triangle) and a deformable outer layer (drawn as three boxes). The Voronoi regions of the rigid core are drawn using lines with dotted stroke. The lines drawn within the right-most deformable layer represent virtual springs and dampers that act to push the bodies apart. We do not consider the triangle to have a deformable outer layer in the Voronoi regions corresponding to the triangle vertices because the surface normal at those points are indeterminate. A surface normal could be selected arbitrarily, but care must be taken to ensure that the differential equations remain continuous.

In addition to surveying the necessary algorithms, this paper focuses on practical and robust floating point imple-

[1] George Washington University, Washington, DC 20052, USA
[2] Toyota Research Institute, Palo Alto, CA 94306, USA

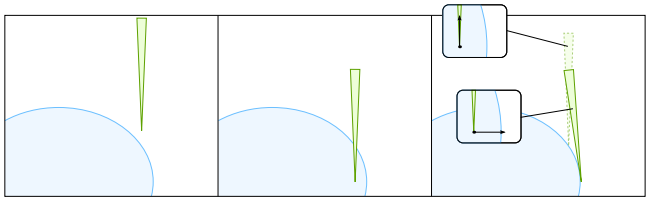


Fig. 2. Interpenetration violates an invariant of rigid bodies. Nevertheless, preventing interpenetration is technically challenging and constraint stabilization approaches like [2] use the notion of signed distance to minimize interpenetration. Every metric, including *minimum translational distance* [6], can cause bodies to be pushed apart in a different direction than they took while interpenetrating, as this illustration shows.

mentations. To that end, we point to sources and libraries for necessary routines.

II. BACKGROUND

This section discusses the state of the art in multi-rigid body dynamics with contact (§II-A); the relationship between vertices, edges, and faces in polyhedron (§II-B); and the minimum translational distance metric (§II-C).

A. Multi-body dynamics with contact

We now describe several topics pertinent to simulating multi-rigid body dynamics with contact, including the rigid contact model, the quasi-rigid contact model, and contact constraint stabilization.

1) Rigid body dynamics with rigid contact: Rigid contact can be modeled as differential variational inequalities (DVI) [22] or as differential algebraic equations. Rigid contact is conceptually straightforward to consider; one need not consider undeformed and deformed shapes, nor attempt to approximate the interconnected nodes in a finite element simulation with a computationally fast model. Literature on the rigid contact model uses the signed distance function, often denoted $\phi(.)$, extensively. The signed distance function is defined to be positive if the bodies are separated, zero if they are in a kissing configuration, and negative if they are interpenetrating. Note that while the signed distance between two rigid bodies *should* always be non-negative, time stepping methods like [26], [3] generally allow the distance between bodies to become negative—to be corrected using constraint stabilization, as will be discussed below—for computational efficiency.

2) Rigid body dynamics with quasi-rigid contact: The earliest works in multi-body dynamics used a quasi-rigid contact model [19]. Not only is this model simple to implement, it is more consistent with reality; however, the model can suffer from a number of drawbacks, the most prominent of which is stiff differential equations. DVI approaches were later extended to quasi-rigid contacts [15], which combine the advantages of both approaches (compliance, removal of stiffness from the differential equations).

3) Contact constraint stabilization: One challenge with all of these approaches is how to separate interpenetrating bodies; although the rigid body model precludes interpenetration, existing approaches for modeling rigid body dynamics are generally unable to maintain this invariant. Our recent work [9] is an exception, but maintaining non-interpenetration requires significant computation. Standard time stepping based simulation approaches must use *constraint stabilization* methods to correct interpenetration. Convergence proofs for DVI-based time stepping approaches with constraint stabilization typically assume the existence of a continuous signed distance function (see, e.g., [2]); polyhedral geometries cannot admit such a distance function, but the popularity of the representation obliges the present study. Nevertheless, correcting interpenetration impacts accuracy even when the signed distance function is continuous, as Figure 2 illustrates.

B. Polyhedral features

The Descartes-Euler polyhedral formula provides a relationship between the number of vertices (V), edges (E), and faces (F) of a polyhedron.

$$V + F - E = 2 \tag{1}$$

This paper leverages this relationship to reason about asymptotic time complexity of polyhedral *features*, rather than considering numbers of vertices, edges, and faces separately. The formula applies to all convex polyhedra and many non-convex polyhedra as well. We assume each polyhedron input to any algorithm referenced in this paper contains n features.

C. Minimum translational distance

There exist a number of metrics for degree of rigid body interpenetration. One of the most popular, judging by citations in literature, is *minimum translational distance* (MTLD). MTLD is the minimum Euclidean distance a pair of polyhedra must be translated in Cartesian space so that no feature of either polyhedron lies strictly inside the other. The fastest algorithms for computing MTLD on convex polyhedra currently run in quadratic time in the best and worst cases [7]. These algorithms include those based on the Separating Axis Theorem and the Minkowski "difference" [10]. The complexity of the Minkowski difference operation on non-convex polyhedra is $O(n^6)$ in the feature inputs [23].

III. QUESTIONS

A. How should contact points be selected?

Existing algorithms for computing contact forces presume a number of discrete point samples are selected from the contact manifold (in the case when the contact manifold does not correspond to a single point). As we will discuss, *the use of point samples does not necessarily imply loss of solution accuracy*. We assume for now that the surface normal is known.

For rigid contact: The combination of the rigid contact model and the Coulomb friction model has long been known to be subject to *indeterminism* [17] for rigid bodies contacting simultaneously at many points. It is currently unknown whether these indeterminate configurations—for which multiple, equally valid possible contact wrenches exist—can result in non-unique accelerations. As intimated above, this

finding shows that providing an increasingly dense sampling of the contact manifold does not yield an increase in solution accuracy.

With that caveat in mind, an insufficient number of samples can be problematic. A single point of contact to represent the surface between two cubes contacting at a face from each would result in interpenetration. Therefore, we suggest requiring that **the point samples be chosen such that, if the relative normal velocities between the bodies are non-negative at all point contact samples, the time derivative of the signed distance between the bodies is non-negative**. Here, as elsewhere in the paper, we use the convention that a positive normal velocity indicates impending separation (and therefore, that a negative normal velocity indicates impending interpenetration). For the ensuing discussion, we also assume that the rigid bodies are in "kissing" contact, i.e., that the signed distance between the two bodies is zero.

$$\dot{\phi} \geq 0 \text{ if } \hat{\boldsymbol{n}}^{\mathsf{T}} \boldsymbol{v}(\boldsymbol{p}_i) \geq 0, \forall i \tag{2}$$

where

$$\boldsymbol{v}(\boldsymbol{p}_i) \equiv \dot{\boldsymbol{x}}_A - \dot{\boldsymbol{x}}_B + \boldsymbol{\omega}_A \times (\boldsymbol{p}_i - \boldsymbol{x}_A) - \boldsymbol{\omega}_B \times (\boldsymbol{p}_i - \boldsymbol{x}_B) \tag{3}$$

For convex polyhedra, this requirement is straightforward to meet. The intersection between the two polyhedra will be convex [20], so—by convexity—if the relative normal velocities at all vertices of the convex hull of intersection are non-negative, the relative normal velocities at all points on the convex hull of intersection must be non-negative. In other words, *contact samples should be selected from the vertices of the convex hull of intersection for rigid bodies with convex polyhedral geometries*.

For rigid contact, point samples should only be generated for rigid bodies that are in kissing contact (i.e., $\phi = 0$). Floating point arithmetic makes such a condition generally infeasible. We now consider the cases of disjoint and interpenetrating bodies. In both cases, we must translate, rotate, or both one (or both) of the two bodies until they are in kissing contact. At that point, contact points can be sampled as already described above. Exactly how the bodies should be translated or rotated is not clear, particularly as the goal kissing configuration is unknown (implied by Figure 2). There is no obvious metric that should be optimized either: for example, moving the bodies along the path that minimizes squared work is not generally correct, which can be seen by examining the case where the velocity of the rigid bodies are initially zero (which would allow *any* ending configuration to be chosen, as long the velocities were to remain zero). Since no path is necessarily "right", we use the surface normal (with its determination described in the following section) to translate the bodies apart. The distance that the bodies should be translated apart to yield a kissing configuration is hereafter denoted δ.

For quasi-rigid contact: For the case of quasi-rigid contact, the issue of rigid contact indeterminacy no longer applies, meaning that sampling points from the contact manifold is not generally sufficient. However, the three cases— disjoint, kissing, and interpenetrating—need no longer be considered. The cases are instead disjoint and contacting. Surveying existing approaches, the possible choices for selecting contact points are (1) using the deepest point of interpenetration (see, e.g., [19]), (2) tracking points of contact from the "kissing manifold" by introducing new ODE state variables (see, e.g., [25], and (3) selecting points from the volume of intersection (see, e.g., [12], [13], [28], [8]). Approach (2) suffers from needing to find the first time of contact (using a root finding method, for example), which negates one of the key advantages of quasi-rigid contact. We avoid further consideration of that method for this reason and now seek to know whether Approach (1) or (3) is appropriate. Appendix I depicts Approach (3) and also proves that the approach computes the result as if contact forces were applied to the surface of the intersection volume, which fits with our model depicted in Figure 1.

Testing the hypothesis that using the deepest point of interpenetration is equivalent to the approach of sampling from the intersection volume: We hypothesized that the two are equivalent *for sufficiently small integration steps*. We tested this hypothesis using a box embedded into a halfspace with a random initial orientation and zero initial velocity. Spring and damper constants for the quasi-rigid contact model were tuned carefully and set identically for both approaches (deepest-point and volumetric). We plotted the pitch and roll of the box over time as it stabilized to a resting configuration on the half-space. As Figure 3 shows, the two approaches yield markedly different behavior, even for small integration steps (10^{-4}). Thus, the hypothesis is false and the deepest point of interpenetration should not be used.

B. How should the surface normal be selected?

This section will refer collectively to the vector between two closest features on a pair of disjoint bodies, the vector normal to the intersecting surface between kissing bodies, and the vector pointing along the direction that two intersecting bodies (with non-zero intersecting volume) should be pushed apart as the *surface normal*. The convention in all existing simulation literature (of which we are aware) is for the vector to point toward the first body, and we retain that convention in this work.

For rigid contact: Following the discussion in §III-A, there is not a "correct" surface normal for bodies that are not kissing but are to be treated as rigidly contacting. Even when polyhedral bodies are kissing, the contact normal will not be uniquely defined in cases of vertex-vertex, vertex-edge, or edge-edge contact. This particular issue is described further and addressed in [27], though it appears that obtaining robust, fast solutions to such scenarios remains an open problem. *This paper and our software implementations avoid this issue for bodies in kissing contact by returning no result in that case*; in other words, we allow (generally small) interpenetration to occur.

This issue aside, the selection of the surface normal is important to keep bodies from (unrealistically) passing

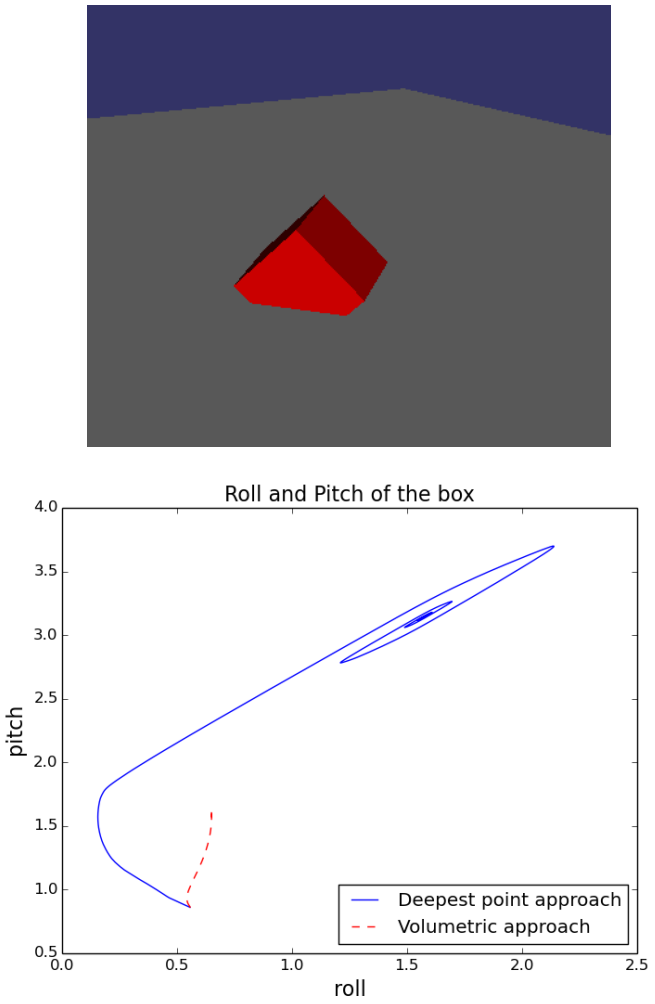


Fig. 3. The roll and pitch (bottom) of a box initially embedded into a half-space in 3D (top), for which the quasi-rigid contact model applies forces that act to separate the bodies; the initial orientation of the box is random and its initial velocity is zero. This plot shows that the pitch and roll of the volumetric approach—which we show applies forces over the surface of the volume of intersection—produces a different result than that obtained using the deepest point of interpenetration. We argue in §III-A that the former approach is the correct one.

through one another or becoming inexplicably locked together. We can specify a few necessary conditions and one recommended condition for selecting the surface normal for contacting rigid bodies.

Necessary condition: The surface normal must point in a direction $\hat{\boldsymbol{d}}$, such that the two bodies translated along $\hat{\boldsymbol{d}}$ and $-\hat{\boldsymbol{d}}$ would increase the Euclidean distance between disjoint bodies and decrease the minimum translational distance between interpenetrating bodies.

For bodies with convex geometries, the necessary condition points to a simple possible choice of surface normal selection: the vector from the center-of-mass of the second body to the center-of-mass of the first body. This selection appears *plausible* [4] but may result in the appearance of artifacts when a contacting body is particularly long, as Figure 5 depicts. The condition would also need to be modified for non-convex polyhedra, as multiple surface normals might be required.

For disjoint bodies, a candidate for the surface normal that is nearly as fast to compute is *a* vector from two closest features (there may be multiple such pairs, as illustrated in Figure 6). As will be discussed in §IV-A, this computation can be performed in "near" constant time [17]. For intersecting bodies, an analogous choice for the surface normal is the direction of minimum translational distance. The asymptotic time complexity computing MTLD is much greater: the current best algorithm runs in time $O(m^{\frac{3}{4}+\epsilon}n^{\frac{3}{4}+\epsilon} + m^{1+\epsilon} + n^{1+\epsilon})$ for some $\epsilon > 0$ [1]. Consistent with the discussion in §III-A, we again note that there is no "right" choice of surface normal selection for disjoint or interpenetrating bodies with rigid contact.

Necessary condition: The surface normal should be piecewise continuous.

Better than piecewise continuity is not achievable since the geometries are polyhedral; surface normals may change discontinuously by moving from one face to another. Accurate solutions to the differential algebraic equations can be obtained only by switching constraints as the surface normal changes; such accurate solutions will not be possible if the surface normal is not piecewise continuous.

Recommended condition: The surface normal should be normal to the contact manifold at $\phi = 0$.

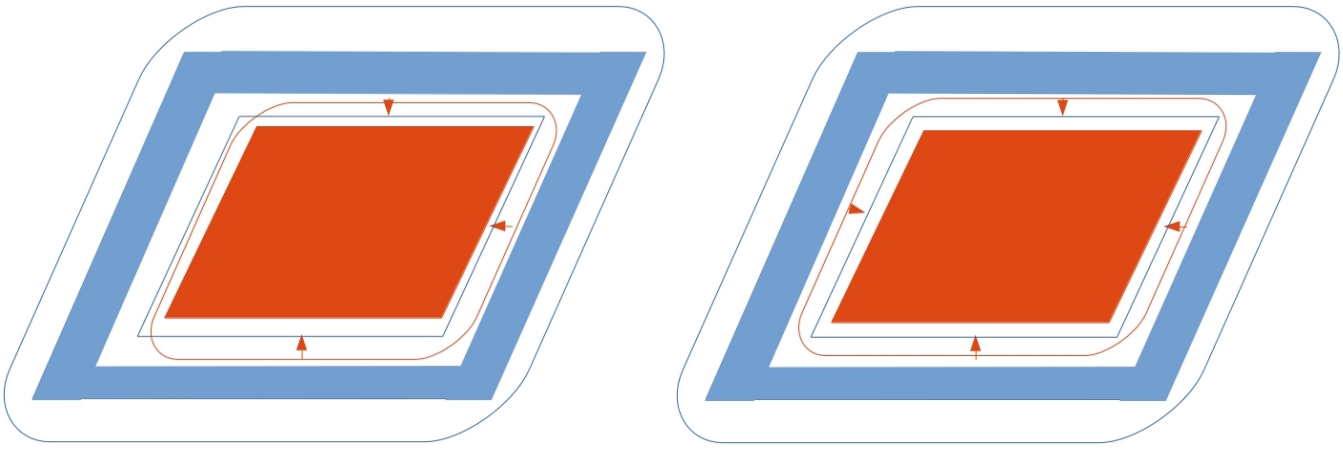
Given floating point arithmetic, the case of truly kissing contact (i.e., $\phi = 0$ rather than $\phi \approx 0$) may never be encountered in many simulations, and the condition of piecewise continuity makes it challenging to identify the downside of neglecting this recommendation. In fact, we believe that further research is necessary to identify further necessary conditions, and possibly recommended conditions as well. However, we can already point to one artifact (depicted in Figure 4) from violating the second necessary condition described above, albeit for a body with non-convex geometry.

For quasi-rigid contact: For quasi-rigid contact, the surface normal is well-defined by using the vector between a pair of closest features, as described above.

IV. TECHNICAL DETAILS

A. Querying whether polyhedra are intersecting

While it is possible to use one algorithm to compute both Euclidean distance and minimum translational distance, it is computationally more efficient to use separate algorithms for the two cases. For convex polyhedra, we recommend one of two algorithms, which are fast and take advantage of convexity: V-CLIP [18] or GJK [11]. Cameron provides uncommented source code for an efficient GJK implementation [5], but geometric data is copied, not referenced, thereby giving up some memory efficiency. Otherwise, numerically robust GJK implementations are rare. V-CLIP increases the numerical robustness of the Lin-Canny Algorithm [16]. Both GJK and V-CLIP run in worst-case linear time of the total features of the queried polyhedra and can leverage temporal coherence between simulation steps to attain near constant time performance [18].



(a) configuration that drives the bodies to the local minimum

(b) configuration at the local minimum

Fig. 4. An example of how improper surface normal selection can fail to give the desired result. The diagrams show two rigid bodies: a parallelogram with a hole (blue) and a solid parallelogram (red). Using the procedure described in [14], the Minkowski sums of each body and a circle are computed and depicted; these are the curved and straight blue lines and the curved red line. [14] indicates that the surface normals are selected as shown with the red arrows. Selecting the surface normals in this way causes the parallelograms in (a) to move to an incorrect, interlocking configuration from which separation would not be possible (b): the arrows cause all relative movement to be cancelled.

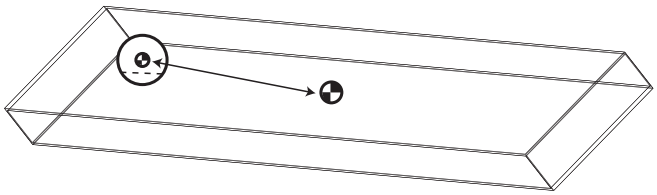


Fig. 5. A small sphere interpenetrating a long box. The points of intersection with the sphere on the box are drawn with dotted stroke. Centers-of-mass of both objects are depicted. If the surface normal is chosen as described in §III-B—pushing the centers-of-mass apart, as depicted—the sphere will counterintuitively not be pushed in a direction normal to the face of the box that the sphere is intersecting.

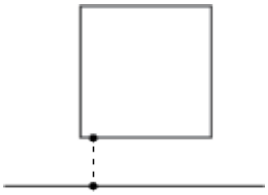


Fig. 6. When convex bodies are disjoint, a reasonable candidate for the surface normal is the vector between a pair of closest features on the two polyhedra. Multiple such closest features generally exist, as is the case for the geometries shown in the given configuration.

B. Computing closest features for disjoint polyhedra

V-CLIP computes a pair of closest features for disjoint polyhedra (the "closest features" returned are nonsensical when polyhedra are intersecting polyhedra). V-CLIP can return vertex/vertex, vertex/edge, vertex/face, and edge/edge types; edge/face and face/face cases are not optional return types (the features returned correspond to the Voronoi region iterates [18]: V-CLIP does not return the most general pair of features). Cameron's GJK implementation returns a pair of closest points, one from each convex polyhedron.

Again assuming that the polyhedra are disjoint, the closest features yield two pieces of information: $\boldsymbol{d}$, a vector from the feature on the second polyhedron to the feature on the first polyhedron, and $\delta = ||\boldsymbol{d}||$, i.e., the Euclidean distance.

C. Computing MTLD (for intersecting bodies)

For computing the minimum translational distance between bodies with convex geometries, we use the separating axis theorem, which requires computing the projection of the bodies onto a quadratic number of axes (a normal from each face of both polyhedra, and the cross products of each edge from one polyhedron and each edge from the other). $O(n^2 \lg n)$ operations are required, if using a Dobkin-Kirkpatrick hierarchy data structure to speed extremal distance queries. The axis that yields the minimum absolute overlap of the projections of the two polyhedra's vertices is the direction of minimum translational distance, and the absolute overlap is the minimum translational distance, δ. The separating axis theorem-based process is described in [10] in detail.

D. Computing the surface normal direction for rigid bodies in a kissing configuration

For a pair of kissing rigid bodies, we define the *contact plane* as the plane defined by the surface normal and a point interior to both polyhedra (which can be calculated using the procedure in §IV-G), as depicted in Figure 7. The plane can be defined by two such normals; this section describes how to find the one that points toward the first body. We compute the distance of the plane defined by each face of each polyhedron from the interior point. If that distance is zero and the face belongs to the second body, a candidate surface normal has been located. If that distance is zero and the face belongs to the first body, *the negation* of a candidate surface normal has been located. For face/vertex, face/edge, and face/face contact types—the only ones we consider—at

least one such surface normal must be found; zero tolerances can be adjusted until the condition holds. Multiple candidate normals that do not point in the same direction indicate that an indeterminate normal has been located (refer back to §III-B).

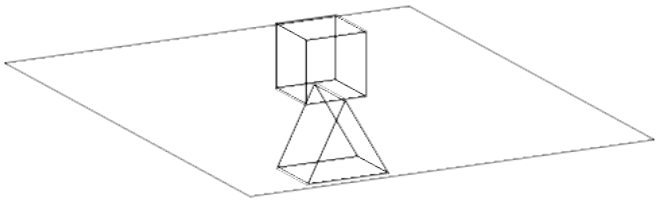


Fig. 7. Depiction of the contact plane for two rigid bodies in a kissing contact configuration. The contact plane is coplanar with the bottom face of the box; the top edge of the prism also lies on the plane.

E. Computing the two 2D convex hulls

When the two bodies are translated along $\boldsymbol{d}$ by distance δ, the two bodies are in a kissing configuration and contact features can be computed. Since the contact plane will be touching only the surface of each polyhedron, one or more vertices from each polyhedron can completely describe the contact surface. The set of these vertices—this set can be determined by calculating the distance of each vertex of the polyhedron from the contact plane—will comprise either a point, an edge, or a convex polygon on the contact manifold.

The number of operations required is $O(n)$ to compute the distances of the polyhedral vertices from the contact plane and $O(n \lg n)$ to compute the convex hull (if necessary). We compute distances of vertices from the contact plane using only dot product operations:

$$\sigma(\boldsymbol{p}) \equiv \hat{\boldsymbol{n}}^{\mathsf{T}}\boldsymbol{p} - \alpha \tag{4}$$

where the plane is defined by:

$$\hat{\boldsymbol{n}}^{\mathsf{T}}\boldsymbol{x} - \alpha = 0 \tag{5}$$

for unit vector $\hat{\boldsymbol{n}}$ and scalar α.

We use `qhull` to compute the 2D convex hull operation.

F. Intersecting the polygons/line segments/points

The contact manifold can be characterized for convex polyhedra using a point, a line segment, or a convex polygon. The manifold is computed using intersection operations.

The number of operations required is $O(n)$ to compute the intersection of two polygons or a polygon and a line segment. Intersecting two line segments, a line segment and a vertex, or check that two vertices are collocated is a constant time operation. For intersecting polygons, we use a floating-point version of an algorithm from [21]. For intersecting a line segment and a polygon, we also use a floating-point version of an algorithm from [21]. For intersecting two line segments in 2D, we use an implementation from `http://www.geometrictools.com`. For checking whether a point lies on a line segment, we compute the distance of the point from the line segment, also using an implementation from `http://www.geometrictools.com`.

G. Intersection volumes

For constructing the intersection between two intersecting convex polyhedra, we use the following approach: (1) find a point, $\boldsymbol{p}$ that maximizes the distance from all faces of both polyhedra and is inside both polyhedra (we use this operation instead of the interior-point finding approach described in [20]); (2) translate both polyhedra by $\boldsymbol{p}$; (3) compute the duals of the two translated polyhedra as described in [20]; (4) compute the convex hulls of the duals of the polyhedra; and (5) compute the dual of the resulting convex hull and translate it by $\boldsymbol{p}$. Step (1) can be computed in linear time using Seidel's linear programming approach [24]; Steps (2), (3), and (5) can be computed in linear time as well. Step (4) requires $O(n \lg n)$ time. Linear time convex polyhedron intersection algorithms exist, but the approach described above is relatively simple to implement, robust (assuming `qhull` or software of similar quality is used to compute convex hulls), and is relatively fast.

V. CONCLUSION

This paper has worked to transform concepts that have been formulated for exact arithmetic to algorithms in floating point arithmetic, while retaining physical correctness to the greatest possible degree. These algorithms have been implemented and tested extensively in `Moby`. These implementations have demonstrated numerically robust and reasonably fast performance, even for polyhedra with thousands of vertices, although running time would not appear impressive compared to gaming or animation codes. Additionally, the lack of a stress-testing suite of contact simulation problems makes it impossible to quantify performance improvements (or even to demonstrate correctness). In the presence of such a void, we have worked to motivate our solutions using proofs, conceptual models, and diagrams, and hope for keen readers to identify and publish flaws in our reasoning.

ACKNOWLEDGMENT

This work was funded by a GWU Undergraduate Research Fellowship for Bjoern Cheng Yi and ARO grant W911NF-16-1-0118.

APPENDIX I
CORRECTNESS PROOF FOR THE VOLUMETRIC QUASI-RIGID CONTACT APPROACH

This section proves that the volumetric quasi-rigid contact approach computes the result as if contact forces were applied to the surface of the intersection volume.

We do this by proving that the mean force exerted on the two vertices of an edge is equivalent the mean force exerted all along the edge. We consider an edge with vertices $\boldsymbol{p}_t$ and $\boldsymbol{p}_h$. The force on a single point $\boldsymbol{p}$ on the edge will be calculated by the formula, $f_p = k_p\phi(\boldsymbol{p}) + k_v\dot{\phi}(\boldsymbol{p})$, where $\phi(\boldsymbol{p})$ is the interpenetration depth of $\boldsymbol{p}$ and $\dot{\phi}(\boldsymbol{p})$is the relative normal velocity at $\boldsymbol{p}$; this is a "standard" quasi-rigid contact formula.

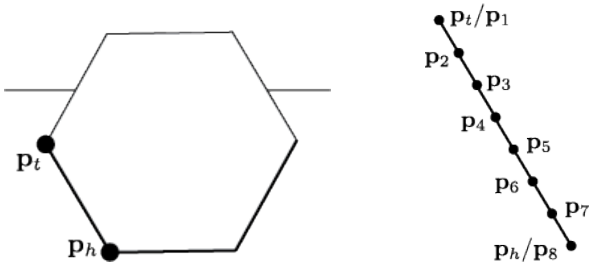


Fig. 8. A line segment of a hexagon interpenetrating the compliant layer of a half-space with quasi-rigid contact (left) and a depiction of dividing the line segment when $n = 8$. The parts of the hexagon to which contact forces are applied are outlined in heavier stroke.

The total force on the edge using the volumetric method is computed using the following formula:

$$f = \frac{1}{n}\sum_{i=1}^{n} f_{\boldsymbol{p}_i} \tag{6}$$

where $\boldsymbol{p}_i$ are n points spaced equally from $\boldsymbol{p}_t$ to $\boldsymbol{p}_h$, resulting in $n-1$ segments. Since $\overline{\boldsymbol{p}_h\boldsymbol{p}_t}$ is a line segment, any point $\boldsymbol{p}$ on $\overline{\boldsymbol{p}_h\boldsymbol{p}_t}$ can be represented as $\boldsymbol{p}_h+(\boldsymbol{p}_t-\boldsymbol{p}_h)s$ where $1 \geq s \geq 0$, i.e., $\boldsymbol{p}$ is a linear function of s. For simplicity, we assume that the global frame is rotated so that only one dimension contributes to the penetration depth. This coordinate of $\boldsymbol{p}$ can then be written in the form $p_{h_y} + (p_{t_y} - p_{h_y})s$ because ϕ is a linear function of the vertex depth. Therefore, $\phi(\boldsymbol{p}) = a \cdot \text{depth}(\boldsymbol{p}) + b = ap_{h_y} + (p_{t_y} - p_{h_y})s + b$ where $a, b \in \mathbb{R}$. Let $\boldsymbol{o}$ be the center of rotation and assume without loss of generality that one of the bodies is fixed; then $\dot{\phi}(\boldsymbol{p}) = \dot{x}_y + (\boldsymbol{\omega} \times (\boldsymbol{p} - \boldsymbol{o}))_y$, where $\dot{\boldsymbol{x}}$ and $\boldsymbol{\omega}$ are the dynamic body's linear and angular velocities, respectively. Plugging in the two previous results into the equation $f = (f_h+f_t)/2$ yields:

$$f = \frac{f_h + f_t}{2} \tag{7}$$

$$= \frac{a\ k_p\ (p_{h_y} + p_{t_y}) + k_v(2\dot{x}_y + (\boldsymbol{\omega} \times (\boldsymbol{p}_t + \boldsymbol{p}_h - 2\boldsymbol{o}))_y)}{2} \tag{8}$$

$$= \frac{ak_p(p_{h_y} + p_{t_y}) + k_v(\boldsymbol{\omega} \times (\boldsymbol{p}_t + \boldsymbol{p}_h))_y}{2} + k_p b + k_v\dot{x}_y - k_v(\boldsymbol{\omega} \times \boldsymbol{o})_y \tag{9}$$

Similarly,

$$\frac{1}{n}\sum_{i=1}^{n} f_{\boldsymbol{p}_i} = \frac{1}{n}(ak_p\sum_{i=1}^{n}(y_{\boldsymbol{p}_i}) - k_v(\boldsymbol{\omega}\sum_{i=1}^{n}\boldsymbol{p}_i)_y + k_p b + \dot{x}_y - k_v(\boldsymbol{\omega} \times \boldsymbol{o})_y \tag{10}$$

because there are $n-1$ uniform cuts on $\overline{\boldsymbol{p}_n\boldsymbol{p}_t}$. Therefore, $s_i = \frac{i-1}{n-1}$, where s_i is the s corresponding to each $\boldsymbol{p}_i$. Then,

$$\sum_{i=1}^{n}\boldsymbol{p}_i = \sum_{i=1}^{n}\boldsymbol{p}_h + \frac{\boldsymbol{p}_t - \boldsymbol{p}_h}{n-1}(i-1) \tag{11}$$

$$= n\boldsymbol{p}_h + \frac{\boldsymbol{p}_t - \boldsymbol{p}_h}{n-1}\frac{n(n-1)}{2} \tag{12}$$

$$= n\boldsymbol{p}_h + \frac{n\boldsymbol{p}_t - n\boldsymbol{p}_h}{2} \tag{13}$$

$$= \frac{n\boldsymbol{p}_t + n\boldsymbol{p}_h}{2} \tag{14}$$

Similarly,

$$\sum_{i=1}^{n} p_{iy} = \frac{np_{t_y} + np_{h_y}}{2} \tag{15}$$

Plugging in the result to the original equation, we get:

$$\frac{1}{n}(ak_p\frac{np_{t_y} + np_{h_y}}{2} + k_v(\boldsymbol{\omega} \times \frac{n\boldsymbol{p}_t + n\boldsymbol{p}_h}{2})_y) + k_v\dot{x}_y + k_p b - k_v(\boldsymbol{\omega} \times \boldsymbol{o})_y \tag{16}$$

This is equivalent to:

$$\frac{ak_p(p_{h_y} + p_{t_y}) + k_v(\boldsymbol{\omega} \times (\boldsymbol{p}_t + \boldsymbol{p}_h))_y}{2} + k_p b + k_v\dot{x}_y - k_v(\boldsymbol{\omega} \times \boldsymbol{O})_y \tag{17}$$

Therefore, $f = \frac{1}{n}\sum_{i=1}^{n} f_{\boldsymbol{p}_i}$.

REFERENCES

[1] P. K. Agarwal, L. J. Guibas, S. Har-Peled, A. Rabinovitch, and M. Sharir. Penetration depth of two convex polytopes in 3D. *Nord. J. Comput.*, 7(3):227–240, 2000.

[2] M. Anitescu and G. D. Hart. A constraint-stabilized time-stepping approach for rigid multibody dynamics with joints, contacts, and friction. *Intl. Journal for Numerical Methods in Engineering*, 60(14):2335–2371, 2004.

[3] M. Anitescu and F. A. Potra. Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems. *Nonlinear Dynamics*, 14:231–247, 1997.

[4] R. Barzel, J. F. Hughes, and D. N. Wood. Plausible motion simulation for computer graphics animation. In R. Boulic and G. Hégron, editors, *Computer Animation and Simulation (Proc. Eurographics Workshop)*, pages 183–197, 1996.

[5] S. Cameron. Enhancing GJK: Computing minimum and penetration distances between convex polyhedra. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Albuquerque, NM, USA, April 1997.

[6] S. A. Cameron and R. Culley. Determining the minimum translational distance between two convex polyhedra. In *Proc. IEEE Intl. Conf. Robotics Automation (ICRA)*, pages 591–596, 1986.

[7] D. Dobkin, J. Hershberger, D. Kirkpatrick, and S. Suri. Computing the intersection-depth of polyhedra. *Algorithmica*, 9:518–533, 1993.

[8] E. Drumwright. A fast and stable penalty method for rigid body simulation. *IEEE Trans. on Visualization and Computer Graphics*, 14(1):231–240, Jan/Feb 2008.

[9] E. M. Drumwright and J. C. Trinkle. Contact simulation. In *Handbook of Humanoid Robotics*. Springer, (under review).

[10] C. Ericson. *Real-Time Collision Detection*. Morgan Kaufmann, 2005.

[11] E. G. Gilbert, D. W. Johnson, and S. S. Keerthi. A fast procedure for computing the distance between complex objects in three-dimensional space. *IEEE J. of Robotics and Automation*, 4(2):193–203, April 1988.

[12] S. Hasegawa and N. Fujii. Real-time rigid body simulation based on volumetric penalty method. In *Proc. of the 11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS)*, 2003.

[13] S. Hasegawa and M. Sato. Real-time rigid body simulation for haptic interactions based on contact volume of polygonal objects. In *Proc. of Eurographics*, 2004.

[14] K. Hauser. Robust contact generation for robot simulation with unstructured meshes. In *Proc. Intl. Symp. Robotics Research (ISRR)*, 2013.

[15] C. Lacoursière. *Ghosts and Machines: Regularized Variational Methods for Interactive Simulations of Multibodies with Dry Frictional Contacts*. PhD thesis, Umeå University, 2007.

[16] M. C. Lin and J. F. Canny. Efficient collision detection for animation. In *Proc. Eurographics Workshop on Animation and Simulation*, 1992.

[17] B. Mirtich. *Impulse-based Dynamic Simulation of Rigid Body Systems*. PhD thesis, University of California, Berkeley, 1996.

[18] B. Mirtich. V-Clip: fast and robust polyhedral collision detection. *ACM Trans. on Graphics*, 17(3):177–208, 1998.

[19] M. Moore and J. Wilhelms. Collision detection and response for computer animation. In *Proc. of Intl. Conf. on Computer Graphics and Interactive Techniques*, pages 289–298, 1988.

[20] D. E. Muller and F. P. Preparata. Finding the intersection of two convex polyhedra. *Theoretical Computer Science*, 7:217–236, 1978.

[21] J. O'Rourke. *Computational Geometry in C*. Cambridge University Press, second edition, 2001.

[22] J.-S. Pang and D. E. Stewart. Differential variational inequalities. *Math. Program., Ser. A*, 113:345–424, 2008.

[23] S. Redon and M. Lin. A fast method for local penetration depth computation. *J. of Graphics Tools*, 11(2):37–50, 2006.

[24] R. Seidel. Linear programming and convex hulls made easy. In *Proc. of the sixth annual symposium on Computational Geometry*, pages 211–215, 1990.

[25] P. Song, P. Kraus, V. Kumar, and P. Dupont. Analysis of rigid body dynamic models for simulation of systems with frictional contacts. *J. Applied Mech.*, 68, 2000.

[26] D. E. Stewart and J. C. Trinkle. An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and Coulomb friction. *Intl. J. Numerical Methods in Engineering*, 39(15):2673–2691, 1996.

[27] J. Williams, Y. Lu, and J. Trinkle. A geometrically accurate contact model for polytopes in multi-rigid-body simulation. *ASME J. Computational and Nonlinear Dynamics*, 2016.

[28] K. Yamane and Y. Nakamura. Stable penalty-based model of frictional contacts. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Orlando, FL, USA, May 2006.