Proceedings of the ASME 2014 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2014
August 17-20, 2014, Buffalo, New York, USA

DETC2014-35231

A COMPLEMENTARITY BASED CONTACT MODEL FOR GEOMETRICALLY ACCURATE TREATMENT OF POLYTOPES IN SIMULATION

Jedediyah Williams*
Computer Science Robotics Laboratory
Department of Computer Science
Rensselaer Polytechnic Institute
Troy, NY 12180
Email: jedediyah@gmail.com

Ying Lu
Department of Computer Science
Rensselaer Polytechnic Institute
Troy, New York, 12180
rosebudflyaway@gmail.com

Jeffrey C. Trinkle
Department of Computer Science
Rensselaer Polytechnic Institute
Troy, New York, 12180
trinkle@gmail.com

ABSTRACT

We present a contact model for rigid-body simulation that considers the local geometry at points of contact between convex polyhedra in order to improve physical fidelity and stability of simulation. This model formulates contact constraints as sets of complementarity problems in a novel way, avoiding or correcting the pitfalls of previous models. We begin by providing insight into the special considerations of collision detection needed to prevent interpenetration of bodies during time-stepping simulation. Then, three fundamental complementarity based contact constraints are presented which provide the foundation for our model. We then provide general formulations for 2D and 3D which accurately represent the complete set of physically feasible contact interactions in six unique configurations. Finally, experimental results are presented which demonstrate the improved accuracy of our model compared to four others.

NOMENCLATURE

a Scalar values are represented in plain face.
$\mathbf{a}$ Column vectors are represented as lower case in bold face.
$\hat{\mathbf{a}}$ The hat operator denotes a unit vector $i.e.$ $\hat{\mathbf{a}} = \frac{\mathbf{a}}{||\mathbf{a}||}$.
$\mathbf{A}$ Matrices are represented as upper case in bold face.
$\mathbf{A}^T$ The ^{T} operator denotes the transpose of a matrix or vector.
$\mathbf{a} \cdot \mathbf{b}$ The $\cdot$ operator denotes the dot product of two vectors.
$\mathbf{a} \times \mathbf{b}$ The $\times$ operator denotes the cross product of two vectors.
$\mathbf{a} \perp \mathbf{b}$ The $\perp$ operator denotes orthogonality $i.e.$ $\mathbf{a} \cdot \mathbf{b} = 0$.

INTRODUCTION

There is a wide spectrum of applications for simulation, but an understandable bias toward real-time interactivity has bred a culture that leaves concerns of physical accuracy in a dark corner. As a result, applications which require some level of physical fidelity from simulation are left with a relatively underdeveloped set of tools.

There is a classic trade-off of speed versus accuracy in simulation, and many applications simply do not gain from improved physical accuracy. The gaming community is naturally concerned with methods that are fast and reliable, but not necessarily accurate. Many games in fact intentionally incorporate non-realistic physics into their play. The graphics community, which is constantly pushing the boundaries of realistic visual effects, does not offer many improvements to physically accurate simulation. Although there are many examples in a long history of "physics based" methods in use in animation [1, 2], there is really no need for such methods to do more than approximate some convenient alternative version of reality. Indeed, a more pressing concern in many graphics applications is to present the viewer with something that "feels" correct.

Particularly in the field of robotics, physical fidelity is a pressing concern and is especially challenging with regard to intermittent frictional contact. Robotics experiments tend to be exceedingly costly and time consuming, so that the ability to perform reliable experiments in simulation would be a great benefit. Although some work has been done regarding validation of simu-

*Address all correspondence to this author.

lation [3–5], the problem of measuring physical accuracy of simulation is a difficult one. Certainly we can say that convergence of a constraint solver is not a good metric alone, since the underlying constraints may be flawed. Comparison of a simulation with that of a commercial simulator (an argument the authors have heard with unfortunate frequency) is meaningless without proof of the commercial simulator's accuracy. Even validation of a simulator with experiment can be misinterpreted when it takes the tuning of several simulation parameters to match a single experiment. Such an attempt at validation says little of the simulator's ability to predict a different experiment.

The contact model presented herein is a step toward physical fidelity in multibody dynamics. In the first section, we discuss the way in which we must approach collision detection if we hope to approach physically accurate interactions between bodies in simulation. In the second section, we present three fundamental contact constraints which form the mathematical foundation of our new contact model. These constraints are utilized in the third section where we construct general formulations based on a small set of possible configurations in both 2D and 3D. The fourth section presents experimental results of our method and contrasts performance with popular existing methods. Finally, the fifth section concludes this paper while offering outlook on future work.

Background

There is a vast body of research regarding complementarity problems and methods for their solution [6, 7]. Given a mapping $\mathbf{f}(\mathbf{z}) : \mathbb{R}^n \to \mathbb{R}^n$, the complementarity problem (CP) seeks to find a solution vector $\mathbf{z} \in \mathbb{R}^n$ to satisfy

$$\begin{aligned} \mathbf{z} &\geq \mathbf{0} \\ \mathbf{f}(\mathbf{z}) &\geq \mathbf{0} \\ \mathbf{z}^T \mathbf{f}(\mathbf{z}) &= \mathbf{0} \end{aligned} \tag{1}$$

which will be written compactly as

$$\mathbf{0} \leq \mathbf{f}(\mathbf{z}) \perp \mathbf{z} \geq 0 \tag{2}$$

There are many solvers available for CPs [8], and in particular the linear CP (LCP) [9, 10] where $\mathbf{f}(\mathbf{z})$ is a linear function, *i.e.*, $\mathbf{f}(\mathbf{z}) = \mathbf{A}\mathbf{z} + \mathbf{b}$. We will see in the following sections that the LCP offers a convenient way of modeling contact.

Time-stepping simulation methods utilizing the complementarity problem were introduced by Moreau [11]. Stewart and Trinkle (ST) introduced an implicit time-stepping method [12], which was followed shortly after by a method from Anitescu and Potra (AP) [13].

Let us divide time-stepping methods into two classes: 1. penalty methods (PM) which wait for inter-penetrations to occur and then introduce forces to either correct these penetrations [14] or prevent them from worsening, and 2. differential variational inequality (DVI) methods [15] such as ST which formulate a time-stepping subproblem of constraints to be solved in attempt to prevent penetrations at each step. PMs suffer from oscillatory instabilities that require parameter tuning and small time steps to correct. DVI methods require special collision detection since popular algorithms which determine nearest features or points of overlap may not return all the contact information needed to avoid penetration, or become degenerate for particular configurations (a uniform stack of identical polyhedra is one good test for degeneracy).

We define a contact as occurring between a pair of features of two bodies where a *feature* is either a vertex, edge, or face of a polyhedral body. Let us represent each contact as a function of two features g_a and g_b of two distinct bodies $\mathcal{A}$ and $\mathcal{B}$, and as a 5-tuple $C = \mathcal{C}(g_a, g_b) = \{\mathcal{A}_{id}, \mathcal{B}_{id}, \mathbf{P}_a, \hat{\mathbf{n}}, \psi\}$, where $\mathcal{A}_{id}$ and $\mathcal{B}_{id}$ are body identifiers for $\mathcal{A}$ and $\mathcal{B}$, $\mathbf{P}_a$ is the contact point on $\mathcal{A}$ in world coordinates, $\hat{\mathbf{n}}$ is the contact normal vector in the direction of $\mathcal{A}$ onto $\mathcal{B}$ by convention, and ψ is the signed gap distance between the two bodies. The point of contact $\mathbf{P}_b$ on $\mathcal{B}$ is available since $\mathbf{P}_b = \mathbf{P}_a + \psi\hat{\mathbf{n}}$. A contact may or may not have a normal force of magnitude λ along $\hat{\mathbf{n}}$ associated with it.

Contact constraints have been traditionally modeled as fundamentally unilateral, however representing features as infinite half-spaces at points of contact generates non-physical behaviour at corners where multiple finite features meet and terminate. Consider a vertex v near a corner as depicted in Figure 1. When the collision detection routine identifies both $C_1 = \mathcal{C}(v, e_1)$ and $C_2 = \mathcal{C}(v, e_2)$, the vertex becomes erroneously trapped in the convex free space at the corner if both contacts are enforced. This is a known consequence in ST, and we therefore refer to it as the *Stewart-Trinkle trap*.

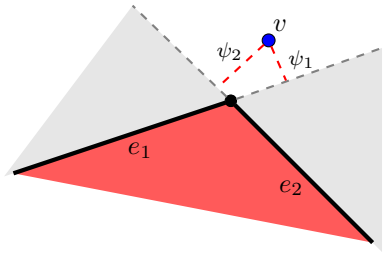


FIGURE 1: The Stewart-Trinkle trap. A vertex v approaching two edges e_1 and e_2 at a corner of a body. If unilateral constraints are enforced against both edges, the vertex becomes trapped by the sum of the edges' half-spaces (grey regions with body). The same trap can occur in 3D between a vertex and set of faces or an edge against multiple edges.

Much inspiration for this work came from Nguyen [16] who introduced the locally non-convex (LNC) model for avoiding the ST-trap in 2D for a vertex against a convex polygon.

SPECIAL CONSIDERATIONS IN COLLISION DETECTION FOR PREVENTATIVE METHODS

There is much to be said regarding collision detection and its intimate connection with constraint determination in simulation, however we will limit the scope of discussion here to those ideas necessary to support the construction of our contact model.

We start by defining a convex polyhedron in terms of features. We may refer to a vertex either as an object v or as the

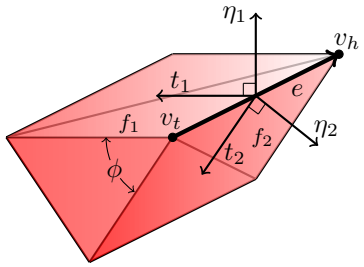


FIGURE 2: Geometric representation of a convex polyhedron.

vector $\mathbf{v}$ representing the vertex's position. Similarly, an edge object $e = (v_t, v_h)$ corresponds to a vector $\mathbf{e} = \mathbf{v}_h - \mathbf{v}_t$. A body is composed of a set vertices V, edges E, and triangle faces F defined by three vertices in counter-clockwise order. The normal vector for a given face $f = (v_i, v_j, v_k) \in F$ is given by

$$\hat{\boldsymbol{\eta}} = \frac{(\mathbf{v}_j - \mathbf{v}_i) \times (\mathbf{v}_k - \mathbf{v}_j)}{||(\mathbf{v}_j - \mathbf{v}_i) \times (\mathbf{v}_k - \mathbf{v}_j)||}. \tag{3}$$

Let us also define two vectors $\mathbf{t}_1$ and $\mathbf{t}_2$ for each edge that are planar with f_1 and f_2 respectively, and perpendicular to the vector $\mathbf{e} = \mathbf{v}_h - \mathbf{v}_t$. These vectors are defined as

$$\begin{aligned} \mathbf{t}_1 &= \hat{\boldsymbol{\eta}}_1 \times \mathbf{e} \\ \mathbf{t}_2 &= \mathbf{e} \times \hat{\boldsymbol{\eta}}_2 \end{aligned} \tag{4}$$

and are useful for geometric tests when determining contacts.

Relaxation of Applicability

The notion of contact applicability is well known in motion planning and collision detection [17–21], and is particularly useful when searching for a pair of nearest features between two convex polyhedra or when determining a boolean collision result. However, classical applicability is not directly useful when determining contacts for preventative methods since it determines only a single contact, and is numerically unstable in common configurations.

For the three definitions of applicability that follow, each has a dimensionless relaxation parameter $\epsilon_\theta = sin(\theta)$ which is a function of the angle θ by which each geometrically extends its domain of applicability.

2D applicabiltiy Let $\omega(v)$ be the set of all vertices that share an edge with v (of course this will be strictly 2 in 2D). We define vertex-edge applicability as a function of the configuration of bodies $\mathbf{q}$ and a vertex-edge pair where $\hat{\boldsymbol{\eta}}_b$ is the edge normal:

$$APPL^{\{ve\}}(\mathbf{q}, v_a, e_b) = \bigwedge_{v_k \in \omega(v_a)} \left[\hat{\boldsymbol{\eta}}_b \cdot \frac{\mathbf{v}_k - \mathbf{v}_a}{||\mathbf{v}_k - \mathbf{v}_a||} \geq -\epsilon_\theta^{\{ve\}} \right] \tag{5}$$

This differs from classical applicability in that we have relaxed the function by an angle θ where originally we would have checked for non-negativity, *i.e.*, $\epsilon_\theta = 0$. Essentially, (5) checks that the edges connected to v_a are pointed "toward" the edge e_b within a tolerance of θ radians.

3D applicabiltiy Vertex-face applicability is analogous to vertex-edge in 2D, and is written for vertex v_a near face f_b as

$$APPL^{\{vf\}}(\mathbf{q}, v_a, f_b) = \bigwedge_{v_k \in \omega(v_a)} \left[\hat{\boldsymbol{\eta}}_b \cdot \frac{\mathbf{v}_k - \mathbf{v}_a}{||\mathbf{v}_k - \mathbf{v}_a||} \geq -\epsilon_\theta^{\{vf\}} \right] \tag{6}$$

where $\hat{\boldsymbol{\eta}}_b$ is the face normal of f_b. We define edge-edge applicability as

$$APPL^{\{ee\}}(\mathbf{q}, e_a, e_b) = \min_{i,j} \ \hat{\boldsymbol{\eta}}_{bi} \cdot \hat{\mathbf{t}}_{aj} \tag{7}$$

where the face normals $\hat{\boldsymbol{\eta}}$ and $\hat{\mathbf{t}}$ vectors are those corresponding to e_a and e_b. Conceptually, edge-edge applicability tests if the edges of two bodies are "facing" one another.

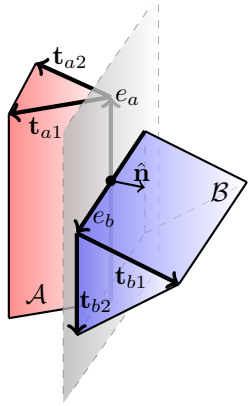


FIGURE 3: Edge-edge contact at the border of stability for classical applicability ($\epsilon_\theta^{\{ee\}} = 0$). The result in this configuration is dependent on machine precision error. Relaxation eliminates this dependency by increasing the domain of applicability.

CONTACT HEURISTICS

Heuristics based on body configuration or properties such as velocities are helpful for determining which contacts to include when constructing the time-stepping subproblem from CPs. Including more contacts results in a larger problem, so being able to reduce or eliminate contact constraints is essential to improving timing performance in simulation.

Applicability is a particularly useful heuristic for determining which contacts could feasibly yield a contact force. Consider the configuration in Figure 4 in which v_a is near two edges of

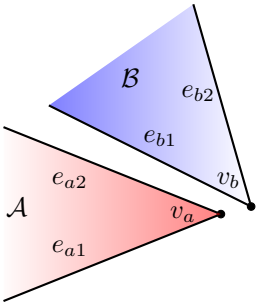


FIGURE 4: Two polygonal bodies with vertices near contact. Heuristics that consider body configuration are crucial when determining what contact information to include when formulating the time-stepping subproblem.

body $\mathcal{B}$. Given reasonable body velocities and time step size, it is clear that $C_1 = \mathcal{C}(v_a, e_{b1})$ could result in a contact force at the end of the current time step, but $C_2 = \mathcal{C}(v_a, e_{b2})$ could not. This is determined by the applicability of v_a against each of the edges

$$APPL^{\{ve\}}(\mathbf{q}, v_a, e_{b1}) = true$$
$$APPL^{\{ve\}}(\mathbf{q}, v_a, e_{b2}) = false$$

Although C_2 will not have a potential force associated with it, it is necessary to include both contacts in the active set in order to give the dynamics formulation information about the geometry for the contact between v_a and $\mathcal{B}$. We will see in the next section how this information is incorporated into the formulation of contact constraints.

CONTACT CONSTRAINTS AS COMPLEMENTARITY PROBLEMS

At each step of simulation, a set of contacts is used to formulate a time-stepping subproblem as a set of complementarity constraints which, when satisfied, determine the state of the system for the next time step. In this section, we define three fundamental constraint formulations as functions of subsets of contacts from the full set of potential contacts determined by collision detection.

Unilateral Constraint

The simplest of the fundamental constraints, a unilateral contact constraint ($\mathcal{U}$-constraint) is a function of a single contact C and may be represented by

$$\mathcal{U}(C) \tag{8}$$

or written in the form of a LCP as

$$0 \leq \psi \perp \lambda \geq 0 \tag{9}$$

Equation (9) is a common contact mode and should match our intuition regarding simple contact between bodies, as it states that if there is a positive gap distance ($\psi > 0$) between the bodies then there should be no mutual force experienced at that contact point ($\lambda = 0$), and if there is a positive force ($\lambda > 0$) being experienced then the bodies should be in contact ($\psi = 0$). Additionally, neither of these quantities should ever be negative. The degenerate case of $\psi = \lambda = 0$ must also be considered. Such a case may occur at the moment just before contact is broken.

Inter-contact Constraint

At a high level, the purpose of an inter-contact constraint ($\mathcal{I}$-constraint) is to avoid the Stewart-Trinkle trap. Given sets of contacts $\mathbf{C}_\iota$ and $\mathbf{C}_\kappa$, where all contacts in both sets are between the same feature of body $\mathcal{A}$ against adjacent features on body $\mathcal{B}$, an $\mathcal{I}$-constraint attempts to

- Constrain at least one gap distance $\psi_{\{\iota,\kappa\}}$ of $\{\mathbf{C}_\iota, \mathbf{C}_\kappa\}$ to be non-negative,
- Permit a positive contact force $\lambda_\iota > 0$ for at most one contact of $\mathbf{C}_\iota$, and
- Only allow $\lambda_\iota > 0$ for a contact with $\psi_\iota = 0$, and when all other $\psi_{\{\iota,\kappa\}}$ are non-positive.

We represent an $\mathcal{I}$-constraint by

$$\mathcal{I}(\mathbf{C}_\iota, \mathbf{C}_\kappa) \tag{10}$$

where $\mathbf{C}_\iota$ is the set of contacts for which there may be a contact force, and $\mathbf{C}_\kappa$ is the set of contacts for which there cannot be a corresponding contact force but with which we determine possible forces on one of the contacts in $\mathbf{C}_\iota$. In other words, the contacts in $\mathbf{C}_\kappa$ will not generate any contact forces between $\mathcal{A}$ and $\mathcal{B}$, but they contain information necessary for fully modeling the contact geometry in a way that prevents penetrations and other non-physical solutions.

The set of complementarity constraints for an $\mathcal{I}$-constraint given $m = |\mathbf{C}_\iota|$ and $n = |\mathbf{C}_\kappa|$ are

$$\begin{aligned} 0 \leq \quad & c_{\iota 2} + \psi_{\iota 1} - \psi_{\iota 2} && \perp \; c_{\iota 2} \geq 0 \\ 0 \leq \quad & c_{\iota 3} + c_{\iota 2} + \psi_{\iota 1} - \psi_{\iota 3} && \perp \; c_{\iota 3} \geq 0 \\ & \vdots \\ 0 \leq \; & c_{\iota m} + c_{\iota m-1} + ... + c_{\iota 2} + \psi_{\iota 1} - \psi_{\iota m} && \perp \; c_{\iota m} \geq 0 \end{aligned} \tag{11}$$

$$\begin{aligned} 0 \leq \quad & c_{\kappa 1} + c_{\iota m} + c_{\iota m-1} + ... + c_{\iota 2} + \psi_{\iota 1} - \psi_{\kappa 1} && \perp \; c_{\kappa 1} \geq 0 \\ & \vdots \\ 0 \leq \; & c_{\kappa n} + ... + c_{\kappa 1} + c_{\iota m} + ... + c_{\iota 2} + \psi_{\iota 1} - \psi_{\kappa n} && \perp \; c_{\kappa n} \geq 0 \end{aligned} \tag{12}$$

$$0 \leq c_{\kappa n} + ... + c_{\kappa 1} + c_{\iota m} + ... + c_{\iota 2} + \psi_{\iota 1} - \alpha e \perp e \geq 0 \tag{13}$$

$$\begin{aligned} 0 \leq \quad & d_{\iota 1} + \psi_{\iota 1} && \perp \; d_{\iota 1} \geq 0 \\ & \vdots \\ 0 \leq \; & d_{\iota m} + \psi_{\iota m} && \perp \; d_{\iota m} \geq 0 \end{aligned} \tag{14}$$

$$\begin{aligned} 0 \leq \quad & c_{\kappa n} + ... + c_{\kappa 1} + c_{\iota m} + ... + c_{\iota 2} + \psi_{\iota 1} + d_{\iota 1} && \perp \; \lambda_{\iota 1} \geq 0 \\ & \vdots \\ 0 \leq \; & c_{\kappa n} + ... + c_{\kappa 1} + c_{\iota m} + ... + c_{\iota 2} + \psi_{\iota 1} + d_{\iota m} && \perp \; \lambda_{\iota m} \geq 0 \end{aligned} \tag{15}$$

where c, d, and e are slack variables, and α is an arbitrary positive constant we will choose to be 1. Equations (11-13) enforce that at least one contact in $\{\mathbf{C}_\iota, \mathbf{C}_\kappa\}$ is non-negative. Equations (14-15) allow at most one contact force to be non-zero, and only permit this force to be non-zero when all other gap distances are non-positive. We unfortunately do not have space to include the lengthy derivation of these constraints here, but insight into the theory behind these constraints is provided by [16].

An $\mathcal{I}$-constraint requires there be at least one contact in $\mathbf{C}_\iota$. If there were not, then there would exist no non-trivial solution to the $\mathcal{I}$-constraint since these potential contact forces are necessary for preventing penetration.

Cross-contact Constraint

Given sets of contacts $\mathbf{C}_\iota$ and $\mathbf{C}_\kappa$, a cross-contact constraint ($\mathcal{X}$-constraint) attempts to constrain either

- At least one non-negative ψ_ι in $\mathbf{C}_\iota$ OR
- At least one non-negative ψ_κ in $\mathbf{C}_\kappa$

We represent a $\mathcal{X}$-constraint as

$$\mathcal{X}(\mathbf{C}_\iota, \mathbf{C}_\kappa) \tag{16}$$

and can write the LCP in multiple forms. Conveniently, $\mathcal{X}$-constraints can be written in the form of equations (11-13). That is, $\mathcal{X}(\mathbf{C}_\iota, \mathbf{C}_\kappa)$ is similar to $\mathcal{I}(\mathbf{C}_\iota, \mathbf{C}_\kappa)$, but excludes equations (14-15) which involve contact forces.

Although a $\mathcal{X}$-constraint necessarily deals with contacts between the same pair of bodies, the contacts need not share features. Subsequently, there is no argument for incorporating contact forces directly into a $\mathcal{X}$-constraint. It should be understood however that care must be taken, when constructing a time-stepping subproblem containing $\mathcal{X}$-constraints, not to construct a subproblem that is inherently unsatisfiable. This is accomplished by ensuring that there exist sufficient $\mathcal{U}$- or $\mathcal{I}$-constraints so as to allow generation of forces which may contribute to achieving configurations that will satisfy all constraints.

POLYGONALLY EXACT GEOMETRY

Given the contact constraints formulated in the previous section, we complete our model by identifying the possible feature combinations and composing for each a set of constraints that accurately represents the complete set of physically feasible contact interactions. We will find that there are two unique configurations in 2D, and four configurations in 3D. We dub our model Polygonally Exact Geometry (PEG).

2D PEG

Given the feature set in 2D to be vertices and edges, we will concern ourselves with the vertex-edge case and the vertex-vertex case, where edge-edge is not necessary since the first two cases prevent all possible penetrations that could result when edges come into contact. All forces in 2D are generated between a vertex of one body against the edge of another. We therefore have two configurations for which we wish to construct sets of contact constraint: vertex against a single edge, and vertex against multiple edges (vertex-vertex).

Vertex-Edge Given a vertex v_a of body $\mathcal{A}$ that is near an edge e_b of body $\mathcal{B}$, for which we determine a contact $C = \mathcal{C}(v_a, e_b)$, we enforce a single $\mathcal{U}$-constraint

$$\mathcal{U}(C)$$

in order to prevent v_a from penetrating e_b.

Vertex-Vertex Here we arrive at our first example with real insight into the usefulness of PEG. The set of contacts we expect from the configuration depicted in Figure 5 is

$$\begin{aligned} C_1 &= \mathcal{C}(v_a, e_{b1}) \\ C_2 &= \mathcal{C}(v_a, e_{b2}) \\ C_3 &= \mathcal{C}(v_b, e_{a1}) \\ C_4 &= \mathcal{C}(v_b, e_{a2}) \end{aligned}$$

However, if we were to simply include a unilateral constraint on each contact, we would be trapping both v_a and v_b against the convex free space defined by the half spaces of the other body's edges.

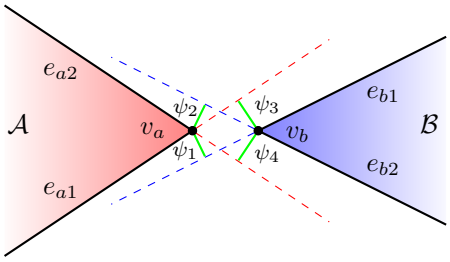


FIGURE 5: A dual case of a vertex near two edges in 2D.

Instead, we write the set of contact constraints as

$$\begin{aligned} &\mathcal{I}(\{C_1, C_2\}, \{\}) \\ &\mathcal{I}(\{C_3, C_4\}, \{\}) \\ &\mathcal{X}(\{C_1\}, \{C_4\}) \\ &\mathcal{X}(\{C_2\}, \{C_3\}) \end{aligned} \tag{17}$$

The $\mathcal{I}$-constraints prevent either vertex v_a or v_b from entering the Stewart-Trinkle trap on the other body. Note that for both $\mathcal{I}$-constraints we have included both contacts in $\mathbf{C}_\iota$ and left $\mathbf{C}_\kappa$ empty. Although this will not be generally true for the 2D vertex-vertex case, it is true for the example in Figure 5 where each contact could feasibly result in a force. The two $\mathcal{X}$-constraints ensure that neither pair of contacts (C_1, C_4) nor (C_2, C_3) can be simultaneously violated, since this would result in inter-penetration such as depicted in Figure 11d. In fact, we may refer to this second trap as the LNC-trap since it is a consequence introduced by the LNC method.

3D PEG

Considering the 3D features of vertices, edges, and faces, we are interested in the four configurations of vertex-face, vertex-edge, edge-edge, and vertex-vertex. Similar to the argument of edge-edge in 2D, face-face and edge-face are redundant.

Vertex-Face The vertex-face case is analogous to the 2D vertex-edge case. Given a vertex v_a near a face f_b as depicted in Figure 6, we include

$$\mathcal{U}(\mathcal{C}(v_a, f_b)) \tag{18}$$

in the set of constraints.

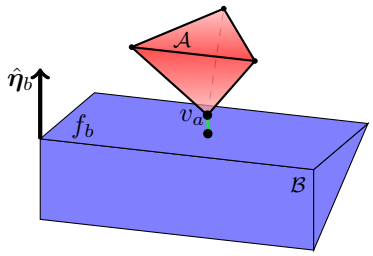


FIGURE 6: A well defined vertex-face contact between vertex v_a of body $\mathcal{A}$ and face f_b of body $\mathcal{B}$.

Edge-Edge Edge-edge contact is also represented by a single unilateral contact. For example, the configuration depicted in Figure 3 would include

$$\mathcal{U}(\mathcal{C}(e_a, e_b)) \tag{19}$$

in order to prevent e_a from penetrating across e_b.

Vertex-Edge Consider Figure 7 where the vertex v_a of a tetrahedron $\mathcal{A}$ approaches the edge of body $\mathcal{B}$.

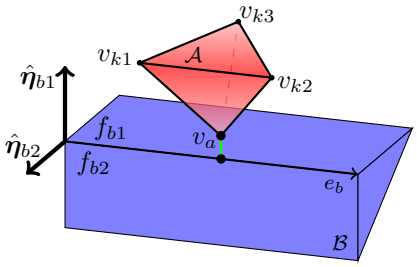


FIGURE 7: An example of vertex-edge configuration. Face normals are depicted, but edge-edge contact normals are excluded for clarity. Consider the relationship between the contacts $\mathcal{C}(v_a, f_{b1})$ and $\mathcal{C}((v_{k3}, v_a), e_b)$. If the first is not enforced, then the second must be, and vice versa.

In the case of vertex-edge there are two vertex-face contacts to consider, as well as $m = |\omega(v_a)|$ edge-edge contacts ($m = 3$

in our example of the tetrahedron). Letting $C_1 = \mathcal{C}(v_a, f_{b1})$ and $C_2 = \mathcal{C}(v_a, f_{b2})$, we first include an $\mathcal{I}$-constraint

$$\mathcal{I}(C_1, C_2) \tag{20}$$

in order to allow v_a to potentially pass one of the half spaces corresponding to faces f_{b1} or f_{b2}. Here we have chosen $\mathbf{C}_\iota = C_1$ and $\mathbf{C}_\kappa = C_2$, but note again that this will not be the case in general. The reason we set $\mathbf{C}_\kappa = C_2$ is to demonstrate that because of the edge $\mathbf{e}_a = \mathbf{v}_a - \mathbf{v}_{k3}$ in Figure 7, $APPL^{\{vf\}}(v_a, f_{b2}) = false$.

We next include $\mathcal{X}$-constraints in order to avoid the LNC-trap. For each edge $\mathbf{e}_{ai} = \mathbf{v}_a - \mathbf{v}_{ki}$, we include

$$\begin{cases} \mathcal{X}(C_1, \mathcal{C}(e_{ai}, e_b)) & \text{if } \hat{\mathbf{n}}_{ai} \cdot (\mathbf{e}_b \times -(\hat{\boldsymbol{\eta}}_{b1} + \hat{\boldsymbol{\eta}}_{b2})) < 0 \\ \mathcal{X}(C_2, \mathcal{C}(e_{ai}, e_b)) & \text{otherwise} \end{cases} \tag{21}$$

where the condition in (21) tests if the edge-edge contact geometrically "opposes" $\mathbf{C}_1$.

Vertex-Vertex In the case of a vertex v_a near another body's vertex v_b, we wish to avoid trapping v_a against the faces of $\mathcal{B}$, and similarly for v_b against $\mathcal{A}$. To achieve this, we include

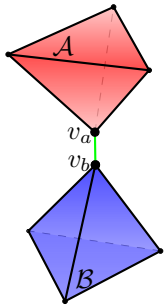


FIGURE 8: A vertex v_a of body $\mathcal{A}$ approaching a vertex v_b of body $\mathcal{B}$ from above. Vertex v_a should be permitted to break the half spaces represented by the falsely extended faces of $\mathcal{B}$, but only when doing so does not violate edge-edge contacts.

$\mathcal{I}$-constraints for v_a against all the faces connected to v_b,

$$\mathcal{I}(\{\mathcal{C}(v_a, f_{b1}), ..., \mathcal{C}(v_a, f_{bn})\}, \{\}) \tag{22}$$

where n is the number of faces connected to v_b. Next we include $\mathcal{I}$-constraints between v_b and all the faces connected to v_a

$$\mathcal{I}(\{\mathcal{C}(v_b, f_{a1}), ..., \mathcal{C}(v_b, f_{am})\}, \{\}) \tag{23}$$

where m is the number of faces connected to v_a.

We additionally require $\mathcal{X}$-constraints for each face pair (f_a, f_b) in the form

$$\mathcal{X}(\{\mathcal{C}(v_a, f_b), \mathcal{C}(v_b, f_a)\}, \{\mathcal{C}(e_{a1}, e_{b1}), \mathcal{C}(e_{a2}, e_{b2})\}) \tag{24}$$

where f_a has edges e_{a1}, e_{a2} connected to v_a, and f_b has edges e_{b1}, and e_{b2} connected to v_b. These $\mathcal{X}$-constraints are enforced only for edges e_{ar} and e_{bs} which have edge-edge applicability and feasibility. The constraint in equation (24) is similar to the $\mathcal{X}$-constraints in equation (21) for the vertex-edge case, but permits a vertex to pass on either side of a face instead of either side of an edge.

Time stepping with PEG

We demonstrate how to use PEG in time-stepping by incorporating it into the ST method. Beginning with the Newton-Euler equation

$$\mathbf{M}(\mathbf{q})\dot{\boldsymbol{\nu}} = \mathbf{G}(\mathbf{q})\boldsymbol{\lambda} + \boldsymbol{\lambda}_{ext} \tag{25}$$

where $\mathbf{M}$ is the mass-inertia matrix diagonally composed of $\mathbf{M}_i = \begin{bmatrix} m_i\mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_i \end{bmatrix}$ for each i^{th} body with mass m_i and inertia tensor $\mathbf{J}_i$, $\dot{\boldsymbol{\nu}}$ is the first derivative of the generalized velocities $\boldsymbol{\nu}_i = \begin{bmatrix} \mathbf{v}_i^T & \boldsymbol{\omega}_i^T \end{bmatrix}^T$, $\boldsymbol{\lambda}$ is a column vector of forces resulting from constraints imposed upon the bodies, $\mathbf{G}$ is the corresponding constraint Jacobian, and $\boldsymbol{\lambda}_{ext}$ is a column vector of the external forces applied to the bodies. $\mathbf{M}$ and $\mathbf{G}$ are expressed in the world frame and therefore functions of the body configurations $\mathbf{q}$, however we drop the parentheses to simplify notation. Given contact j on body i, the constraint Jacobian is given by

$$\mathbf{G}_{ij} = \begin{bmatrix} \hat{\mathbf{n}}_{ij} \\ \mathbf{r}_{ij} \times \hat{\mathbf{n}}_{ij} \end{bmatrix} \tag{26}$$

where $\mathbf{r}$ is the vector from the center of mass of body i to the point of contact j on body i. We discretize equation (25) from time step ℓ to $\ell + 1$ as

$$\mathbf{M}\boldsymbol{\nu}^{\ell+1} = \mathbf{M}\boldsymbol{\nu}^{\ell} + \mathbf{G}\mathbf{p}^{\ell+1} + \mathbf{p}_{ext}^{\ell+1} \tag{27}$$

where $\mathbf{p}$ and $\mathbf{p}_{ext}$ are impulses, *i.e.*, $\mathbf{p} = h\boldsymbol{\lambda}$ for time step size h.

Given the Taylor series approximation of a gap distance ψ at the end of the current time step as

$$\psi^{\ell+1} = \psi^{\ell} + \mathbf{G}^T\boldsymbol{\nu}^{\ell+1}h + \frac{\partial\psi^{\ell}}{\partial t}h \tag{28}$$

we can substitute (28) into our three contact constraints and rewrite them along with (27) in vector-matrix form as a mixed LCP (MLCP) to get

$$\begin{bmatrix} \mathbf{0} \\ \boldsymbol{\rho}_n^{\ell+1} \\ \boldsymbol{\gamma} \\ \boldsymbol{\delta} \\ \boldsymbol{\epsilon} \\ \boldsymbol{\rho}^{\ell+1} \end{bmatrix} = \begin{bmatrix} -\mathbf{M} & \mathbf{G}_n & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{G}_p \\ \mathbf{G}_n^T & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{G}_c^T & \mathbf{0} & \mathbf{E}_c & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{G}_d^T & \mathbf{0} & \mathbf{0} & \mathbf{E}_d & \mathbf{0} & \mathbf{0} \\ \mathbf{G}_e^T & \mathbf{0} & \mathbf{E}_{ec} & \mathbf{0} & \mathbf{E}_e & \mathbf{0} \\ \mathbf{G}_p^T & \mathbf{0} & \mathbf{E}_{pc} & \mathbf{E}_{pd} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\nu}^{\ell+1} \\ \mathbf{p}_n^{\ell+1} \\ \mathbf{c} \\ \mathbf{d} \\ \mathbf{e} \\ \mathbf{p}_p^{\ell+1} \end{bmatrix} + \begin{bmatrix} \mathbf{M}\boldsymbol{\nu}^{\ell} + \mathbf{p}_{ext} \\ \boldsymbol{\psi}_n^{\ell}/h + \partial\boldsymbol{\psi}_n^{\ell}/\partial t \\ \boldsymbol{\psi}_c^{\ell}/h + \partial\boldsymbol{\psi}_c^{\ell}/\partial t \\ \boldsymbol{\psi}_d^{\ell}/h + \partial\boldsymbol{\psi}_d^{\ell}/\partial t \\ \boldsymbol{\psi}_e^{\ell}/h + \partial\boldsymbol{\psi}_e^{\ell}/\partial t \\ \boldsymbol{\psi}_p^{\ell}/h + \partial\boldsymbol{\psi}_p^{\ell}/\partial t \end{bmatrix} \tag{29}$$

$$\mathbf{0} \leq \begin{bmatrix} \boldsymbol{\rho}_n^{\ell+1} \\ \boldsymbol{\gamma} \\ \boldsymbol{\delta} \\ \boldsymbol{\epsilon} \\ \boldsymbol{\rho}_p^{\ell+1} \end{bmatrix} \perp \begin{bmatrix} \mathbf{p}_n^{\ell+1} \\ \mathbf{c} \\ \mathbf{d} \\ \mathbf{e} \\ \mathbf{p}_p^{\ell+1} \end{bmatrix} \geq \mathbf{0} \tag{30}$$

where the rows of (29) respectively correspond to the discretized forms of equations (27), (9), (11-12), (14), (13), and (15), and each of the Jacobians $\mathbf{G}$ correspond to the contacts related to those constraints. The exception is $\mathbf{G}_c$ which is given for the i^{th} body and k^{th} index of a given constraint as

$$\mathbf{G}_{c_{ik}} = \mathbf{G}_{\iota_{i1}} - \mathbf{G}_{\{\iota,\kappa\}_{ik}} \tag{31}$$

The $\mathbf{E}$ sub-matrices are selection matrices, and are conveniently all identity matrices with the one exception that $\mathbf{E}_{pc} = diag(ones(m_j, 1))$ where m_j is the number of contacts associated with the j^{th} $\mathcal{I}$-constraint. Like the Jacobians, each gap distance in each $\boldsymbol{\psi}$ corresponds to those contacts that make up the constraints for that row, but again with an exception in the third row that

$$\psi_{c_{ik}} = \psi_{\iota_{i1}} - \psi_{\{\iota,\kappa\}_{ik}} \tag{32}$$

In (30), $\mathbf{p}_n^{\ell+1}$ represents the impulses corresponding to $\mathcal{U}$-constraints, and $\mathbf{p}_p^{\ell+1}$ represents impulses corresponding to $\mathcal{I}$- and $\mathcal{X}$-constraints. All remaining variables in (30) can be considered slack variables.

We have disregarded friction for the time being, but note that it may be included in exactly the same manner as with ST. Also note that if there are only $\mathcal{U}$-constraints, the time-stepping subproblem represented by (29-30) reduces to ST.

EXPERIMENTAL COMPARISON OF CONTACT MODELS

There are many body configurations in which PEG outperforms other contact models in terms of stability and physical fidelity. There are other simpler configurations for which PEG reduces to LNC or ST. Here we present the results of 2D and 3D benchmark comparisons. Although Figures 11 and 12 demonstrate simulation over relatively large time steps which would be impractical for most models, they give the reader a sense of how these other models fail while demonstrating the stability of PEG.

Consider the 2D vertex-vertex case depicted in Figure 9 where a blue triangle is moving right to left toward a similar red triangle at rest. Figure 11 depicts the results among five different simulation methods.

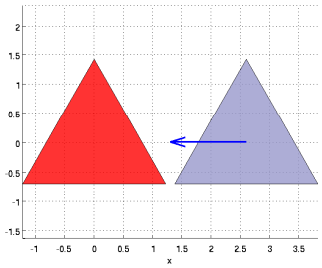


FIGURE 9: A blue triangle (right) approaching a stationary red triangle from right to left. Similar configurations are common where several polyhedra lie on the same surface.

For an example in 3D, consider the problem of stacking congruent objects. Figure 10 depicts a red block falling under the influence of gravity toward a static blue body (fixed with respect to the world frame). Figure 12 shows the results of this 3D example.

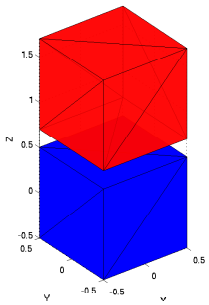
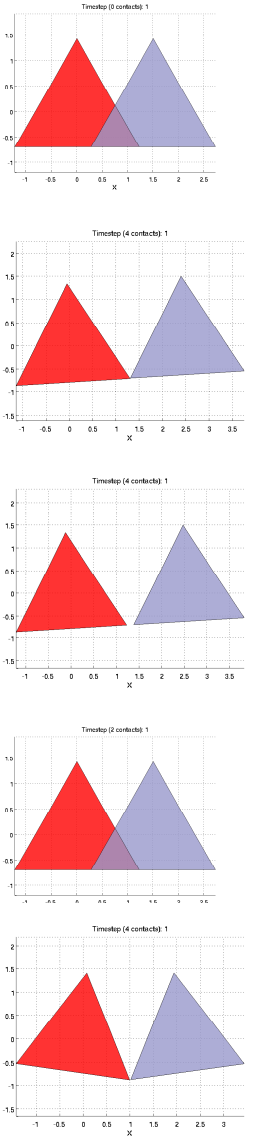


FIGURE 10: A red (top) block falling onto a static blue block. This configuration has four vertex-vertex cases.



(a) PM
Penalty methods don't prevent contact, so at the end of the first time step there is penetration that will need to be "corrected" over the next time step.

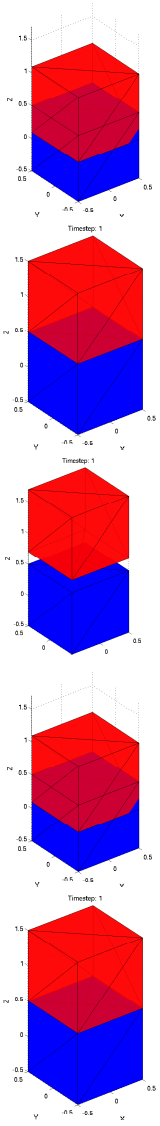
(b) ST
Stewart-Trinkle treats the four contacts in this configuration as four independent unilateral contacts. As a result, the nearest vertices of both bodies are trapped at the contact point, allowing the momentum of the blue body to rotate both bodies about this point.

(c) AP
We see the same result with Anitescu-Potra as we do with Stewart-Trinkle, except that in addition to the overly constrained solution, we also see the gap distance prevented from decreasing.

(d) LNC
Although physically feasible solutions do exist with the LNC model, this non-physical configuration is unfortunately also a valid solution when using LNC. Here the result is the same after the first time step as when using a PM.

(e) PEG
PEG successfully avoids the traps fallen into by the previous methods, even given the relatively large time step. In practice, we have found the PATH solver [8] is able to solve the PEG formulation, resulting in a more stable and accurate simulation.

FIGURE 11: Comparison of model behaviours over a single large time step of 2D simulation, $h = 0.5$ s. All simulations were implemented using RPIsim [22].



(a) PM
As we saw in the 2D case, we again note that after a single time step the penalty method has allowed a large penetration.

(b) ST
An interesting case in that it appears ST has accurately simulated the contact interaction. However, this is only because the correct behaviour happens to fit into the Stewart-Trinkle trap! Each vertex in contact is trapped by the three nearest faces of the other body.

(c) AP
Despite gravity, the contacts erroneously prevent the top body from falling. As was the case in 2D, this is in addition to also being caught in the Stewart-Trinkle trap.

(d) LNC
Since the vertex contacts with outside faces all have $\psi = 0$ at the end of the time step, LNC is perfectly happy to allow the top body to fall through the bottom. Again, the result is the same after the first time step as when using a PM.

(e) PEG
PEG successfully prevents penetration. Although the resulting body positions match those of ST above, the vertices are not trapped as they are with ST and the red body is free to move horizontally should it encounter an outside force.

FIGURE 12: Comparison of model behaviours in 3D with gravity over a single large time step of $h = 0.25$ s. A red block is dropped onto a static blue block.

DISCUSSION AND FUTURE WORK

A contact model, PEG, was presented in which contact constraints are formulated as configuration-dependent complementarity problems. These problems have solutions which prevent interpenetration of bodies and correspond to physically accurate behaviour. PEG may require more time to formulate and solve constraints than other models in certain configurations, but offers improved accuracy and stability for even very large time steps. Additionally, PEG requires no parameter tuning in order to achieve this stability. Table 1 summarizes the behavior of PEG along with four other methods.

PEG has been shown to be solvable in practice using the PATH solver, however, we currently offer no proof of the solvability of equation (29). Other solvers, particularly the LMMCP solver [23] offer promising solution approaches, yet there remain many interesting and unanswered questions regarding the process of solving the PEG formulation. For example, we have observed that when there are multiple distinct solutions, *e.g.* a vertex near an edge in 3D that will have active contact with either face f_1 or f_2, then re-ordering the constraints may affect a given

TABLE 1: Summary of model comparisons.

Attribute	PM	ST	AP	LNC	PEG
1. Applies only one impulse per contact manifold	✓			✓	✓
2. Prevents inter-penetrations for reasonable time steps		✓	✓		✓
3. Applies impulses only at body surfaces				✓	✓

solver's "choice" of active contact. If we find that the ordering of constraints deterministically gives priority to one possible outcome over another, then we may be consistently ignoring a significant region of the solution space. An alternative approach to solving PEG could divide the problem into smaller subproblems to be solved independently. Heuristics could then be utilized to choose either a subproblem to solve or a solution from among them.

REFERENCES

[1] Terzopoulos, D., Pltt, J., Barr, A., Zeltzer, D., Witkin, A., and Blinn, J., 1989. "Physically-based modeling: Past, present, and future". *SIGGRAPH Comput. Graph.,* ***23***(5), July, pp. 191–209.

[2] Witkin, A., and Baraff, D. Physically based modeling: Principles and practice. Tech. rep., SIGGRAPH 1997. Course Notes.

[3] Macaluso, W., 2012. "Exploring the domain of applicability of simulated 2d rigid body dynamical systems". Master's thesis, Rensselaer Polytechnic Institute, Department of Computer Science.

[4] Berard, S., 2009. "Using simulation for planning and design of robotic systems with intermittent contact". PhD thesis, Rensselaer Polytechnic Institute, Department of Computer Science.

[5] Berard, S., Nguyen, B., Anderson, K., and Trinkle, J., 2010. "Sources of error in a rigid body simulation of rigid parts on a vibrating rigid plate". *ASME Journal of Computational and Nonlinear Dynamcis*. to appear.

[6] Billups, S. C., and Murty, K. G., 2000. "Complementarity problems". *J. Comput. Appl. Math,* ***124***, pp. 303–318.

[7] Bender, J., Erleben, K., Trinkle, J., and Coumins, E., 2012. "Interactive simulation of rigid body dynamics in computer graphics". In Conference of the European Association for Computer Graphics, State of The Art Report (STAR).

[8] Ferris, M., and Munson, T., 2000. "Complementarity problems in gams and the path solver". *Journal of Economic Dynamics and Control,* ***24***(2), pp. 165 – 88.

[9] Cottle, R., Pang, J., and Stone, R., 1992. *The Linear Complementarity Problem*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104).

[10] Erleben, K., 2013. "Numerical methods for linear complementarity problems in physics-based animation". In ACM SIGGRAPH 2013 Courses, SIGGRAPH '13, ACM, pp. 8:1–8:42.

[11] Moreau, J., and Panagiotopoulos, P., 1988. "Unilateral contact and dry friction in finite freedom dynamics". In Nonsmooth Mechanics and Applications, Springer, pp. 1–82.

[12] Stewart, D., and Trinkle, J., 1996. "An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction". *International Journal for Numerical Methods in Engineering,* ***39***(15), August, pp. 2673 – 91.

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

[14] Moore, M., and Wilhelms, J., 1988. "Collision detection and response for computer animation". In Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '88, ACM, pp. 289–298.

[15] Pang, J.-S., and Stewart, D., 2008. "Differential variational inequalities". *Mathematical Programming,* ***113***(2), pp. 345–424.

[16] Nguyen, B., 2011. "Locally non-convex contact models and solution methods for accurate physical simulation in robotics". PhD thesis, Rensselaer Polytechnic Institute.

[17] Donald, B. R., 1984. Local and global techniques for motion planning. Tech. rep., Massachusetts Institute of Technology.

[18] Latombe, J.-C., 1991. *Robot Motion Planning*. Kluwer Academic Publishers, Norwell, MA, USA.

[19] Lin, M. C., 1993. Efficient collision detection for animation and robotics. Tech. rep., University of California, Berkeley.

[20] Jimnez, P., Thomas, F., and Torras, C., 1998. "Collision detection algorithms for motion planning". In Lecture notes in control and information sciences, 229, Springer, pp. 305–343.

[21] Jimnez, P., Thomas, F., and Torras, C., 2000. "3d collision detection: A survey". *Computers and Graphics,* ***25***, pp. 269–285.

[22] Williams, J., Lu, Y., Niebe, S., Andersen, M., Erleben, K., and Trinkle, J., 2013. "Rpi-matlab-simulator: A tool for efficient research and practical teaching in multibody dynamics". In Workshop on Virtual Reality Interaction and Physical Simulation (VRIPHYS).

[23] Kanzow, C., and Petra, S., 2005. Lmmcp a levenberg-marquardt-type matlab solver for mixed complementarity problems. `www.mathematik.uni-wuerzburg.de/~kanzow/`. Accessed: 2014-01-27.