Performance of a Method for Formulating Geometrically Exact Complementarity Constraints in Multibody Dynamic Simulation

Article *in* Journal of Computational and Nonlinear Dynamics · September 2014

DOI: 10.1115/1.4027314

CITATIONS
6

READS
99

3 authors:

Daniel Montrallo Flickinger
Sikorsky Aircraft
13 PUBLICATIONS **335** CITATIONS
SEE PROFILE

Jedediyah Williams
Nantucket High School
12 PUBLICATIONS **39** CITATIONS
SEE PROFILE

J.C. (Jeff) Trinkle
Rensselaer Polytechnic Institute
143 PUBLICATIONS **4,351** CITATIONS
SEE PROFILE

Performance of a Method for Formulating Geometrically Exact Complementarity Constraints in Multibody Dynamic Simulation

Daniel Montrallo Flickinger

Research Assistant
CS Robotics Laboratory
Department of Computer Science
Rensselaer Polytechnic Institute
Troy, New York 12180
Email: dmflickinger@gmail.com

Jedediyah Williams

Research Assistant
CS Robotics Laboratory
Department of Computer Science
Rensselaer Polytechnic Institute
Troy, New York 12180
Email: jedediyah@gmail.com

Jeffrey C. Trinkle

Professor
CS Robotics Laboratory
Department of Computer Science
Rensselaer Polytechnic Institute
Troy, New York 12180
Email: trinkle@gmail.com

Contemporary problem formulation methods used in the dynamic simulation of rigid bodies suffer from problems in accuracy, performance, and robustness. Significant allowances for parameter tuning, coupled with careful implementation of a broad phase collision detection scheme is required to make dynamic simulation useful for practical applications. A constraint formulation method is presented herein that is more robust, and not dependent on broad-phase collision detection or system tuning for its behavior.

Several uncomplicated benchmark examples are presented to give an analysis and make a comparison of the new Polyhedral Exact Geometry method with the well-known Stewart-Trinkle method. The behavior and performance for the two methods are discussed. This includes specific cases where contemporary methods fail to match theorized and observed system states in simulation, and how they are ameliorated by the new method presented here. The goal of this work is to complete the groundwork for further research into high performance simulation.

1 Introduction

Many engineering and scientific problems are researched through the simulation of the dynamics of rigid bodies. The demand for fast and accurate simulation is high, as dynamic problems grow larger and larger under limited computational power. The work presented herein is the analysis of a time-stepping method for dynamic simulation intended to push the boundaries of robustness, performance, and accuracy.

Multibody dynamic simulation is used in a broad range of engineering, research, and entertainment fields. Accurate simulation of the motion of machinery is crucial in mechanical design [1–3]. It is particularly useful in designing complex machinery that is expensive to prototype, such as internal combustion engines [4]. Accurate simulation is required for robotics applications, particularly where contact is expected, such as in grasp planning [5]. Virtual reality is more useful for research, and convincing when objects undergo more realistic motion in simulation [6]. Entertainment products such as motion pictures and video games benefit from accurate simulation [7], where realism is enhanced.

There are many challenges involved in providing a reasonable approximation of physical reality through simulation of rigid bodies [8, 9]. This is particularly a difficult issue when considering contact and friction. This consideration makes a system mathematically nonsmooth, and difficult to simulate accurately. A simulation must use constraints that closely represent the geometry of objects interacting with each other. First, accuracy is lost when representing real world objects as topologically connected vertices, faces, and edges in computer simulation [10–12]. In addition, constraints do not always exactly coincide with this ap-

proximated geometric representation, causing further error. These problems can be mitigated through increased accuracy by taking small time steps, or by more accurately mapping the constraints to the geometry. The formulations presented in this paper are of the latter solution.

Simulation of the dynamics of multibody systems has traditionally involved a rigid body model with Coulomb friction; with unilateral constraints described by complementarity conditions [8, 13, 14]. The field is rooted further on the study of classical mechanics [9]. Computer simulation of multibody systems is related to work in the computer graphics field, with polyhedral models of objects common [10–12], and contact generally modeled as occurring between topologically connected features such as vertices, edges, and faces. In common practice, the contact model between vertex and face is used to prevent penetration.

Time-stepping simulations calculate estimates of system state at discrete times. The simulation starts at an initial state, and then calculates the future system state one time step ahead. The solution for the next time step is dependent on the solution for the previous time step. And the simulation proceeds until criteria are met, e.g., reaching a predetermined final time.

Within a time step, two phases are common. First, the constraints are calculated, and the time-stepping subproblem is formulated. These constraints are determined by the geometry and kinematics of the system. Second, the dynamics of the system are calculated from the equations of motion, producing the derivative states. These derivative states are used to calculate all system states for the following time step.

Commonly used time-stepping methods are classified as either prevention, or correction methods. Correction methods consider only pairs of objects that are touching or in penetration during each timestep [15–17]. Current constraint violations are kept from becoming worse, but can take multiple timesteps to correct. This affects the overall accuracy of the simulation, as error from penetration is unavoidable without adjusting step size and affecting simulation performance. Many popular physics engines employ correction type methods, because common collision detection methods are designed only to consider current penetration.

Prevention methods form constraints for pairs of objects which could collide in the following timestep [18–20]. These methods have fewer penetrations, and can avoid undesirable effects such as tunnelling without having to adjust the step length to work correctly under specific system parameters. The Polyhedral Exact Geometry method is of the prevention method class.

The Stewart-Trinkle method is commonly used, and the errors of the constraints in non-convex freespaces can be mitigated by using a broad-phase collision detection algorithm considering only features within a small distance, ϵ of each other. This method bounds the error, but does not eliminate it completely. Further, it requires smaller step sizes as ϵ decreases, which requires more steps and increases total simulation time.

The common methods discussed in this work assume that all bodies are convex. In cases where non-convex bodies exist, they are decomposed into adjoining convex bodies [21, 22]. Non-convex bodies can be decomposed into components comprised of convex bodies, but a non-convex freespace cannot be decomposed trivially into convex regions.

Popular physics engines such as ODE [23] and Bullet [7] use variations of the Stewart-Trinkle [18, 24] and Anitescu-Potra [17, 25] formulations. Inaccuracies resulting from simplifications of Coulomb friction and geometric modeling of constraints result in undesirable behaviors in simulation that make the use of these popular engines for scientific work challenging [26]. The research presented herein continues the work in [27].

The mean error is used as a metric in evaluating the performance of the three formulations. The performance of the Polyhedral Exact Geometry formulation is shown to be more accurate than Stewart-Trinkle and Anitescu-Potra . The higher accuracy allows larger step sizes to be utilized while producing similarly accurate results as Stewart-Trinkle and Anitescu-Potra at smaller step sizes.

2 Methods

2.1 Equations of Motion

Rigid bodies in simulation move in accordance with the equations of motion. In the systems considered in this paper, the Newton-Euler equations are used in the dynamics formulation,

$$\mathbf{M}(\mathbf{q},t)\dot{\nu} = \lambda_{\nu p}(\mathbf{q},\dot{\mathbf{q}},t) + \lambda_{app}, \tag{1}$$

where $\lambda_{\nu p}$ is the sum of velocity dependent forces, and λ_{app} are the applied forces. The system is subject to the complementarity constraint,

$$0 \leq \mathbf{p}_n^{l+1} \perp \mathbf{G}_n^T \nu^{l+1} + \frac{\Psi_n^l}{h} + \frac{\partial \Psi_n^l}{\partial t} \geq 0, \tag{2}$$

where $\mathbf{p}_n$ are impulses normal to contact surfaces, Ψ_n are gap distances, or distances between active bodies and contact surfaces, h is the length of the time step, ν are the velocities of the active bodies, and $\mathbf{G}_n$ is a normal contact wrench.

2.2 Broad phase collision detection

A broad phase collision detection scheme is utilized to avoid the need to include all possible contact constraints at every time step. In the specific implementation discussed herein, if any edge in a body is closer than a predetermined value ϵ to the particle, then all edge constraints in that body are added to the formulation for the current time step. ϵ can be calculated as a function of particle velocity, or tuned for specific systems. In the examples presented in Section 3, ϵ is set to a large value in order to illustrate the differences between solution methods.

The inclusion of the tunable parameter ϵ can cause undesirable behavior when used in conjunction with the

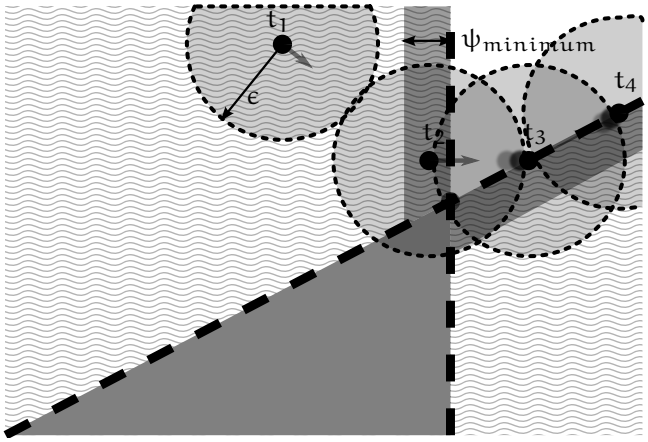

Fig. 1: *Vertex at times $t_1 \ldots t_4$ moving within the constraint half-spaces near an object, Stewart-Trinkle formulation. A large impulse acts on the vertex at t_2, and the diagonal constraint is active at t_3 and t_4.*

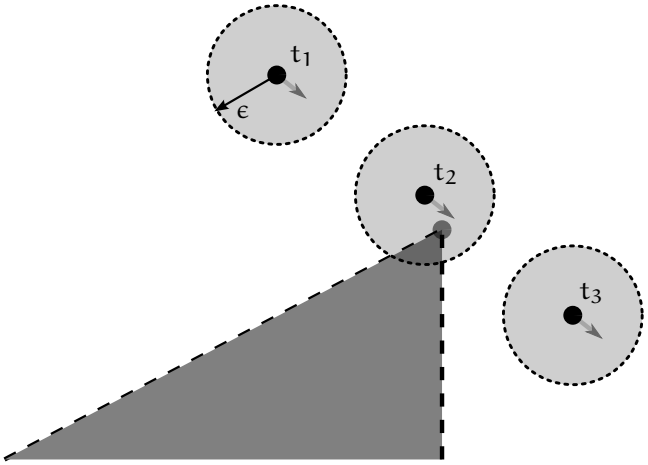

Fig. 2: *Vertex at times $t_1 \ldots t_3$ moving in proximity to an object, Polyhedral Exact Geometry formulation*

Stewart-Trinkle formulation. Figure 1 illustrates a case where an object picks up additional kinetic energy from interacting with constraints extended beyond the actual boundary of the object.

A particle moving in two dimensional space is represented as a black dot. The gray triangular area represents an obstacle, and the particle should be prevented from penetrating it. The two edges, represented by dashed lines create the shaded half-spaces. These depict the region of the unilateral constraints. The space of potential collision around the object within range of ϵ is represented as a dashed circular region.

The active object is shown at several time steps. Starting at t_1, the object has a velocity down and to the right, as indicated by the arrow. At this time, the obstacle object is outside the sphere of radius ϵ, and therefore no constraints are active. At t_2, the obstacle object is within the ϵ bubble, and now both constraints are active. Because this event occurred within a single time-step, the vertical constraint is now violated. This penetration causes a large impulse to be applied. The object hits the diagonal constraint at the next time step t_3, and proceeds to follow the boundary even though the true edge of the non-active object is not in this region. At t_4, the active object has traveled at high velocity, and the other object has crossed outside the ϵ bubble. Now unconstrained, it proceeds at high velocity up and to the right of the figure. This behavior is seen later in Figure 11.

The energy gain can be partially mitigated by defining the parameter $\psi_{minimum}$. This parameter is represented by darker regions on the half-spaces. This allows the simulation to ignore constraints in deep penetration that cause large energy gains.

Figure 2 demonstrates the behavior of the system under the Polyhedral Exact Geometry formulation. The two constraints become active at t_2 as the active object comes with distance ϵ of the non-active object. But the constraints do not alter the behavior of the active object, as they conform to the actual geometry of the non-active object.

2.3 Stewart-Trinkle Formulation of the Mixed Linear Complementarity Problem

Equations (1) and (2) are formulated as a mixed linear complementarity problem [28,29], using the Stewart-Trinkle formulation,

$$\begin{bmatrix} 0 \\ \rho_n^{l+1} \end{bmatrix} = \begin{bmatrix} M & -G_n \\ G_n^T & 0 \end{bmatrix} \begin{bmatrix} v^{l+1} \\ p_n^{l+1} \end{bmatrix} + \begin{bmatrix} -Mv^l - p_{ext}^l \\ \psi_n^l/h \end{bmatrix}, \quad (3)$$

where M is the block diagonal mass-inertia matrix of all active bodies in the system, v is the generalized velocities, p_n and p_{ext} are normal and external impulsive forces, and h is the time step. The normal contact wrench is defined as

$$G_{n_{ij}} = \begin{bmatrix} \hat{n}_{ij} \\ r_{ij} \times \hat{n}_{ij} \end{bmatrix}, \quad (4)$$

which is indexed over k collisions by the i^{th} body and j^{th} collision. Friction is omitted from the formulation for clarity. The term r is the position vector from the center of mass of the body to the point on the body in collision with another object. $\hat{n}$ is a unit vector in the direction normal from the surface in collision.

The problem size for the Stewart-Trinkle formulation is $2 \cdot b + c$, where b is the number of active bodies, and c is the number of contacts [18].

When considering friction, the formulation in (3) becomes,

$$\begin{bmatrix} 0 \\ \rho_n^{l+1} \\ \rho_f^{l+1} \\ \sigma^{l+1} \end{bmatrix} = \begin{bmatrix} M & -G_n & -G_f & 0 \\ G_n^T & 0 & 0 & 0 \\ G_f^T & 0 & 0 & E \\ 0 & U & -E^T & 0 \end{bmatrix} \begin{bmatrix} v^{l+1} \\ p_n^{l+1} \\ p_f^{l+1} \\ s^{l+1} \end{bmatrix} + \begin{bmatrix} -Mv^l - p_{ext}^l \\ \psi_n^l/h \\ 0 \\ 0 \end{bmatrix}, \quad (5)$$

where each of the submatrices are defined in terms of the bodies found to be in collision, indexed over k collisions by the i^{th} body and j^{th} collision, i.e.

$$M = \mathrm{blockdiag}(M_1, ..., M_{n_b})$$

where $M_i = \begin{bmatrix} m_i I_{(3\times3)} & \mathbf{0} \\ \mathbf{0} & J_{i(3\times3)} \end{bmatrix}$ and n_b is the number of bodies. The friction submatrix is,

$$G_{f_{ij}} = \begin{bmatrix} \hat{d}_{ij1} & ... & \hat{d}_{ijn_d} \\ (r_{ij} \times \hat{d}_{ij1}) & ... & (r_{ij} \times \hat{d}_{ijn_d}) \end{bmatrix}$$

where n_d is the number of friction directions in the discretized friction cone, and

$$U = \mathrm{diag}(\mu_1, \ldots, \mu_k),$$

$$E = \mathrm{blockdiag}\left(\begin{bmatrix} 1 \\ 1 \\ \vdots \end{bmatrix}_1, \ldots, \begin{bmatrix} 1 \\ 1 \\ \vdots \end{bmatrix}_{n_c} \right),$$

where each column vector of ones has length equal to n_d and n_c is the number of contacts.

For each body P_i involved in a collision, v_i^{l+1} is extracted from a solution of the MCP and used to update the states of the body,

$$v_i \leftarrow v_i^{l+1}, \tag{6}$$

and

$$u_i \leftarrow u_i + h v_i^{l+1}, \tag{7}$$

and for bodies not in contact,

$$v_i \leftarrow v_i + h \frac{F_{ext_i}}{m_i}, \tag{8}$$

and

$$u_i \leftarrow u_i + h v_i. \tag{9}$$

2.4 Anitescu-Potra Formulation of the Mixed Linear Complementarity Problem

The Anitescu-Potra formulation is identical to the Stewart-Trinkle formulation, but with the constraint violation set to zero. So (3) becomes,

$$\begin{bmatrix} 0 \\ \rho_n^{l+1} \end{bmatrix} = \begin{bmatrix} M & -G_n \\ G_n^T & 0 \end{bmatrix} \begin{bmatrix} v^{l+1} \\ p_n^{l+1} \end{bmatrix} + \begin{bmatrix} -Mv^l - p_{ext}^l \\ 0 \end{bmatrix}, \tag{10}$$

where all terms are as defined in Section 2.3.

2.5 A particle with a single contact constraint

Consider a particle, M, moving in $\Re^2$ space with a single edge, with configuration $\mathbf{q}$ at time t, as depicted in Figure 3. A single edge is present in the space, bounded by vertices p_1 and p_2. This edge presents a unilateral constraint to the particle. That is, the particle is free to move on one side of the edge, but not on the other. The gap distance of the particle is represented as ψ_n, and is a function of the configuration at the current time. It is positive in the direction normal to the edge into the free space.

This unilateral constraint from (2) is represented as the complementarity condition,

$$0 \leq \lambda_n \perp \psi_n(\mathbf{q}, t) \geq 0, \tag{11}$$

where λ_n is the force on the particle, normal to the edge, to keep the particle from crossing the boundary.

The constraint in (11) is valid if the edge is of infinite length. But in this example, the edge is a line segment between two vertices in $\Re^2$ space.

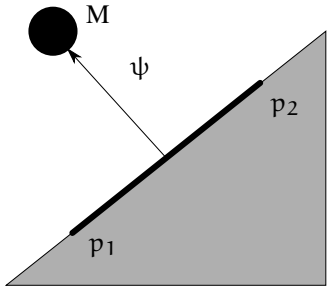

Fig. 3: *A particle interacting with a single edge*

2.6 A particle with two contact constraints

The Polyhedral Exact Geometry (PEG) method is introduced in this section, following its initial development in [27]. Consider a vertex near two edges as shown in Figure 4. The triangular gray area is a solid body that the particle is constrained from penetrating. It is represented as two edges, P_1 - P_2 and P_2 - P_3, shown as dashed lines. The half-spaces for each edge are represented by the shaded areas, with the striped area bounded by edge P_2 - P_3, and the checkered area bounded by edge P_1 - P_2. The area outside of the solid body is the free space of the particle.

If two constraints in the form in (11) are included in the formulation, the free-space for the particle in simulation will be restricted to the white region only. To prevent penetration of the body while allowing penetration into the half-spaces outside the body, the constraint

$$\max(\psi_1, \psi_2) \geq 0 \tag{12}$$

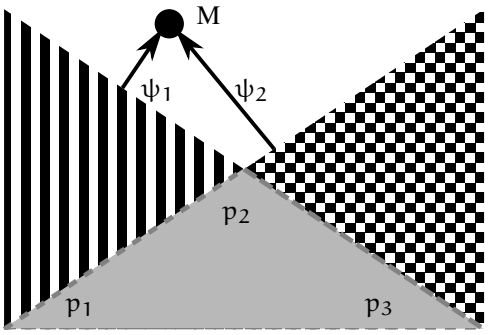

Fig. 4: *Particle M interacting with two edges,* p_1 - p_2 *and* p_2 - p_3*, with half-spaces represented by shaded areas*

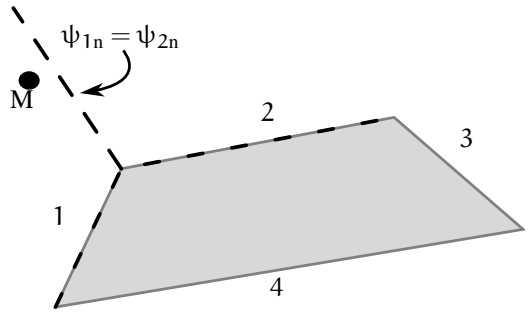

Fig. 5: *Particle M interacting with a planar polygon, the constraints are represented as dashed lines along edges* 1 *and* 2

is proposed [27]. That is, we intend to prevent penetration with at most one half-space. Consider the relation

$$max(\psi_1, \psi_2) = \psi_2 + max(0, \psi_1 - \psi_2). \tag{13}$$

Defining

$$c = max(0, \psi_1 - \psi_2) \tag{14}$$

results in the complementarity constraints

$$0 \leq c - (\psi_1 - \psi_2) \perp c \geq 0, \tag{15}$$

and

$$c + \psi_2 \geq 0. \tag{16}$$

Although not a complementarity condition, (16) is necessary to ensure (14).

Additionally, only one of the constraints should result in a force. We therefore include the constraints

$$0 \leq d_1 + \psi_1 \perp d_1 \geq 0 \tag{17}$$
$$0 \leq d_2 + \psi_2 \perp d_2 \geq 0 \tag{18}$$
$$0 \leq c + \psi_2 + d_1 \perp \lambda_1 \geq 0 \tag{19}$$
$$0 \leq c + \psi_2 + d_2 \perp \lambda_2 \geq 0. \tag{20}$$

Together, these equations allow only either λ_1 or λ_2 to be non-zero.

Consider three edges of a planar polygon shown in Figure 5. Moving left to right along the boundary, one makes a right turn when moving from one edge to the next. In this case, we call the object boundary *locally convex*. For the three edges shown, a point p, lies outside the interior if and only if the following two linear complementarity conditions are satisfied:

$$\begin{aligned} 0 \leq \quad & c_2 + \psi_{1n} - \psi_{2n} \perp c_2 \geq 0 \\ 0 \leq \; & c_3 + c_2 + \psi_{1n} - \psi_{3n} \perp c_3 \geq 0 \end{aligned} \tag{21}$$

where $c_i, ..., i = \{2, 3\}$ are unknowns defined by $c_i = \max(0, \psi_{1n} - \psi_{in})$.

The geometric interpretation of c_2 is shown in Figure 5. The bisector of the angle between edges 1 and 2 defines the line along which $\psi_{1n} = \psi_{2n}$. To the left of the bisector, ψ_{1n} is greater than ψ_{2n}, therefore the first complementarity condition requires $c_2 = 0$. However, on the right of the bisector, the first complementarity condition implies that $c_2 = \psi_{2n} - \psi_{1n}$. When the point in question is on the right of the edge 1 to edge 2 bisector, $c_2 = \psi_{2n} - \psi_{1n}$ can be substituted into the second complementarity condition to yield:

$$0 \leq c_3 + \psi_{2n} - \psi_{3n} \perp c_3 \geq 0, \tag{22}$$

which is the first complementarity condition with all indices incremented by one.

2.7 General Polyhedral Exact Geometry Formulation

Expanding the constraint in (15), a general system with m facets in contact is realized,

$$\begin{aligned}
0 \le c_2 - \psi_2 + \psi_1 \perp c_2 \ge 0 \\
0 \le c_3 - \psi_3 + c_2 + \psi_1 \perp c_3 \ge 0 \\
\vdots \\
0 \le c_m - \psi_m + c_m + c_{m-1} + ... + c_2 + \psi_1 \perp c_m \ge 0 \\
\\
0 \le d_1 + \psi_1 \perp d_1 \ge 0 \\
0 \le d_2 + \psi_2 \perp d_2 \ge 0 \\
\vdots \\
0 \le d_m + \psi_m \perp d_m \ge 0 \\
\\
0 \le d_1 + (c_2 + c_3 + ... + c_{m-1} + c_m) + \psi_1 \perp \lambda_1 \ge 0 \\
0 \le d_2 + (c_2 + c_3 + ... + c_{m-1} + c_m) + \psi_2 \perp \lambda_2 \ge 0 \\
\vdots \\
0 \le d_m + (c_2 + c_3 + ... + c_{m-1} + c_m) + \psi_m \perp \lambda_m \ge 0 \\
\\
0 \le (c_2 + c_3 + ... + c_{m-1} + c_m) + \psi_1,
\end{aligned} \tag{23}$$

where

$$c_i = max(0, \psi_1 - \psi_i), i = 2, \ldots, m, \tag{24}$$

and d are slack variables. Note that $\psi_i := [\psi_n]_i$. See [27] for further details of this derivation.

2.8 Polyhedral Exact Geometry Formulation of the Mixed Linear Complementarity System

The original Stewart-Trinkle formulation in (3) is altered to introduce a heuristic method based on the Polyhedral Exact Geometry formulation discussed in Section 2.3. The main formulation becomes

$$\begin{bmatrix} 0 \\ \rho_n^{l+1} \\ \rho_a^{l+1} \end{bmatrix} = \begin{bmatrix} M & -G_n & 0 \\ G_n^T & 0 & E_1 \\ G_a^T & 0 & E_2 \end{bmatrix} \begin{bmatrix} v^{l+1} \\ p_n^{l+1} \\ c_a^{l+1} \end{bmatrix} + \begin{bmatrix} -Mv^l - p_{ext}^l \\ \Psi_n^l / h \\ \Psi_a / h \end{bmatrix}, \tag{25}$$

where M is the mass matrix, and G_n and G_a are the normal and auxiliary contact wrenches, respectively. Multiple adjacent contacts are grouped, and the components of G_n from (4) are split into G_n and G_a in (25). G_n contains the contact wrenches of the contacts with the minimum value of ψ_n for each group of contacts. The remaining contact wrenches in each manifold are put into G_a. The auxiliary gap functions are defined as

$$\Psi_a = \begin{bmatrix} \Psi_{a_1} \\ \vdots \\ \Psi_{a_{n_s}} \end{bmatrix} \text{ where } \Psi_{a_j} = \begin{bmatrix} \Psi_1 - \Psi_2 \\ \vdots \\ \Psi_1 - \Psi_{n_s} \end{bmatrix}. \tag{26}$$

This employs a heuristic method, where only the closest constraint with a positive distance ψ_n is chosen as the active constraint. Take for example Figure 4, where $0 < \psi_1 < \psi_2$. In this case, ψ_1 would be the active constraint and $\Psi_a = \psi_1 - \psi_2$.

Further constraints are required to guarantee that only one constraint in each group of contacts is active at each instant. One normal contact wrench is put in $\mathbf{G}_n$ for each manifold, while all the other contact normals are put in the auxiliary wrench. The selection matrices apply the additional constraints to enforce that the correct constraint is active.

All elements of the component of the selection matrix E_{1_j} are equal to one, for each contact. The selection matrix E_2 is such that all E_{2_j} are equal to a lower triangular matrix where all nonzero values are equal to one.

The resulting size of the MCP system for the Polyhedral Exact Geometry formulation is $2 \cdot b + c$, where b is the number of active bodies, c is the total number of contacts.

3 Results

The Polyhedral Exact Geometry formulation is compared to the Stewart-Trinkle and Anitescu-Potra formulations through the simulation of multibody dynamic systems. The purpose of this new formulation is to increase simulation performance and accuracy. The median position error is used as a metric, and the simulated trajectories using each of the methods are compared to the solution from the Polyhedral Exact Geometry method at a small step size. Simulations are initially performed at a small step size, and then repeated sequentially after doubling the step size at each overall iteration.

The formulation of the Polyhedral Exact Geometry method is similar to the Stewart-Trinkle [18, 24] formulation, but extended to produce unilateral constraints that are geometrically correct in cases where polyhedral bodies have a locally non-convex free space. Constraints for faces in a polygon model are defined as half-spaces in the Stewart-Trinkle and Anitescu-Potra methods. These half-spaces extend beyond the limits of the faces, producing constraints which are not geometrically accurate.

The Polyhedral Exact Geometry method uses a heuristic to choose which constraints are active based on local geometry. This increases the accuracy of the simulation, and eliminates the need for the specialized methods of broad-phase collision detection required to avoid undesired behaviors in simulation caused by active constraints that should be ignored.

The RPI MATLAB Simulation Testbed [30], using PATH as a solver, is used to evaluate the performance of

all formulation methods. The testbed is designed to evaluate multiple formulations and solvers within a single standard framework. The results in the following sections show that Polyhedral Exact Geometry outperforms Stewart-Trinkle and Anitescu-Potra in accuracy and robustness under a variety of situations. The Anitescu-Potra method is chosen in some experiments as an additional comparison because it does not exhibit the large errors of Stewart-Trinkle when used in simulations with large ϵ values. In these cases, the trajectory error of the Anitescu-Potra method has a lower bound of ϵ.

Two planar systems without friction are presented to compare the Stewart-Trinkle and Polyhedral Exact Geometry time-stepping methods in Section 3.1 and Section 3.2. Both simulations involve a single particle moving in planar space and interacting with bodies represented by vertices connected by edges. Contact in this case is unilateral only, and no other constraints are present. Standard Earth gravity acts downward in the $-Y$ direction in all simulations. All planar simulations were performed using the *minisim* component of the RPI MATLAB Simulator [30].

A three dimensional system comprised of polyhedral rigid bodies is presented in Section 3.4. The same error analysis as in Section 3.3 is performed in Section 3.5. A single rigid body is given an initial position and velocity, then released to move with the force of gravity as the only external force. The RPI MATLAB Simulation [30] is used for this analysis.

3.1 Hills Simulation

A simulation is run with a series of hills as fixed obstacles. The hills are tessellated semicircles composed of 10 edges each. An external force of $(12, 0)$ N is applied to the particle, in addition to gravity. A step size of 10^{-2} seconds is used, and the particle starts at an initial position of 2 meters in the $+Y$ direction. The minimum value of ϵ increased to 0.5 meters.

Figure 6 shows the trajectories for both the Polyhedral Exact Geometry and Stewart-Trinkle methods. The dashed lines represent the boundaries of the half-spaces projected outward from each edge in the bodies.

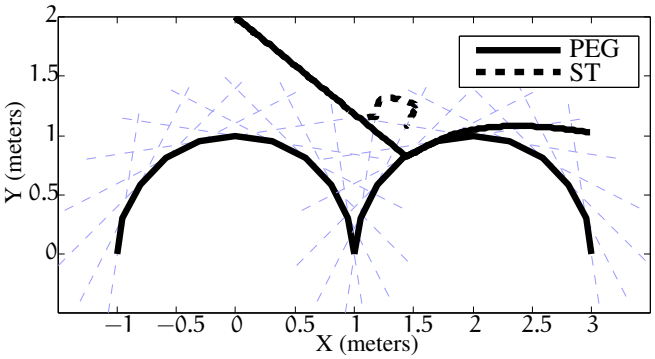

Fig. 6: *Trajectories of the Polyhedral Exact Geometry and Stewart-Trinkle methods*

A detail of the same trajectory from Figure 6, resulting from the Stewart-Trinkle method is shown in Figure 7. The boundaries of the half-spaces that impede the motion of the particle are highlighted.

The particle initially interacts with a horizontal constraint, and subsequently comes into contact with the half-space boundary on the left. Kinetic energy is picked up from this first interaction, as the particle enters the half-space from the side, resulting in penetration as the constraint becomes active when the particle is within range ϵ. This phenomenon is discussed in detail in Section 2.2. The high velocity, combined with the external force on the particle causes it to climb the boundary until the constraint becomes inactive as the distance from the right hill is greater than ϵ. After a brief phase of unconstrained motion, the constraint from one of the half-space edges depicted by thick dashed lines activates. The particle then travels a short distance down that constraint until meeting a second highlighted constraint. The particle is finally trapped at the intersection of these two half-spaces.

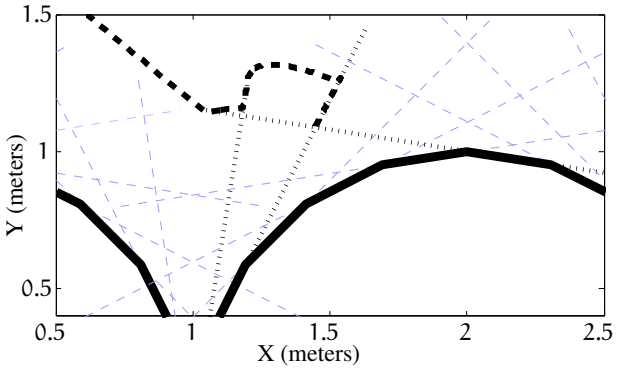

Fig. 7: *Detailed trajectory of Polyhedral Exact Geometry and Stewart-Trinkle methods*

The problem sizes at each time-step are given in Figure 8. The implementation of the methods used herein only considers edges that are within a distance dependent on the current position and velocity of the particle. This causes the problem size to fluctuate, but also keeps the number of constraints low. In the cases presented in this section, the minimum value of this distance is kept large, effectively making it velocity independent. The elapsed wall time for the PATH solver to solve the complementarity problem at each time-step is shown in Figure 9.

3.2 Sawtooth Simulation

A planar simulation featuring a particle interacting with a series of ramps is run using the RPI MATLAB Simulator to compare the performance of the Polyhedral Exact Geometry and the Stewart-Trinkle methods under varying conditions. The particle has mass 5 kg. All surfaces are frictionless in the initial simulation shown in Figure 10. System states are updated at each time step with the results obtained by solving the dynamics and complementarity constraints [8, 13, 14] in (3) or (25) with the PATH solver [28, 29].

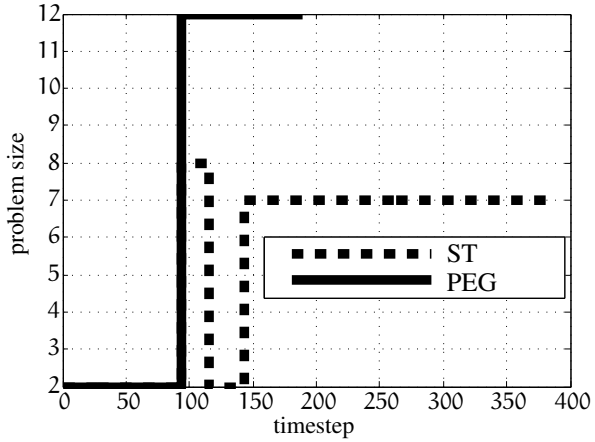

Fig. 8: *Problem size of the Polyhedral Exact Geometry and Stewart-Trinkle methods, hills simulation*

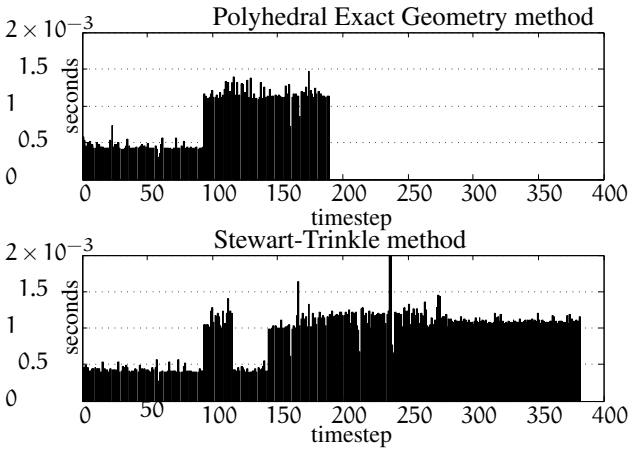

Fig. 9: *Solver time for the Polyhedral Exact Geometry and Stewart-Trinkle methods, hills simulation*

The simulated trajectories are shown in Figure 10. A single run using each method is depicted on the same plot. The particle starts at $[0.00, 0.75]$ meters, with an initial velocity of $[2.0, 1.8]$ meters per second.

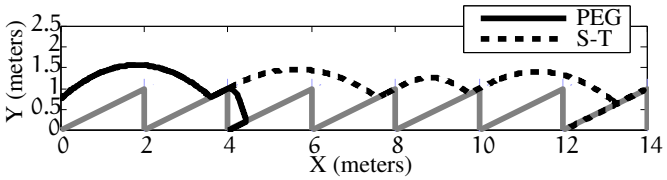

Fig. 10: *Trajectories of the Polyhedral Exact Geometry and Stewart-Trinkle methods without friction*

The trajectories are shown in more detail in Figure 11. The point near where the particle leaves the second ramp is examined. Using the Stewart-Trinkle method, the particle remains constrained past the boundary of the ramp. The value of ϵ is 0.25 meters; the effect of the constraint on the overall trajectory of the particle is significant. The particle leaves the sawtooth with a high velocity as it instantaneously enters a region in deep penetration of the half space constraint resulting from the vertical edge.

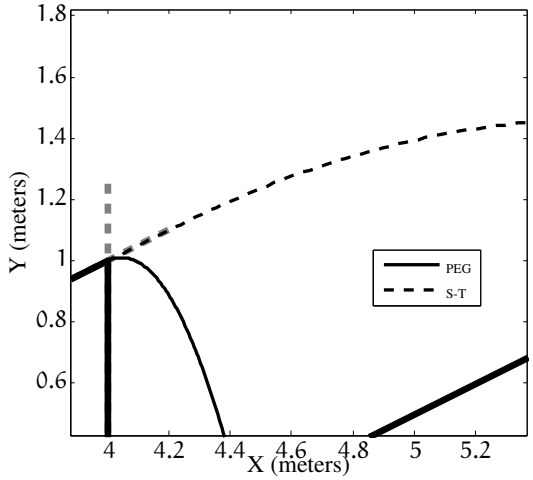

Fig. 11: *Detail of the trajectories of the Polyhedral Exact Geometry and Stewart-Trinkle methods without friction*

The problem size at each time-step during the simulations is plotted in Figure 12. The problem size varies depending on the broad phase collision detection, and is the size of the main matrices in (3) and (25). Both methods start with no active constraints, giving the minimum problem size of 2, which corresponds with the size of the mass matrix.

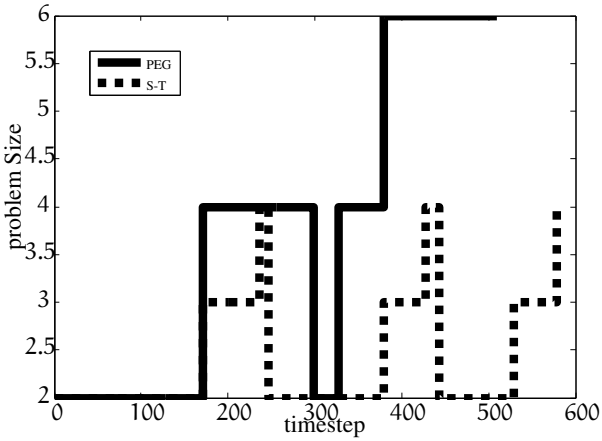

Fig. 12: *Problem size for the Polyhedral Exact Geometry and Stewart-Trinkle methods*

Figure 13 shows the elapsed time of the PATH solver at each time-step for both methods. The required computation time generally corresponds with the problem size, with the Polyhedral Exact Geometry method being at a slight disadvantage at the small problem sizes in these small benchmark problems.

Another simulation of the sawtooth system is performed using the RPI MATLAB Simulator, this time with friction.

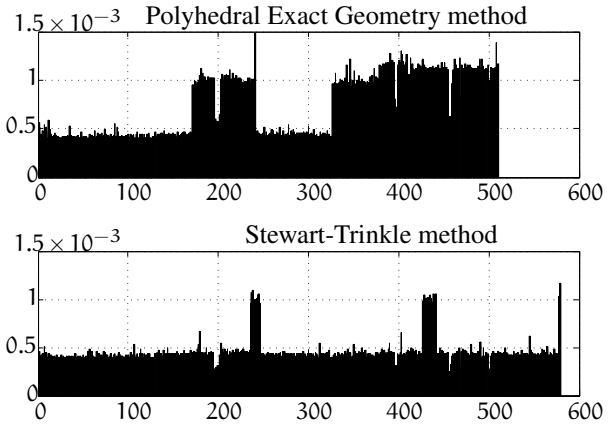

Fig. 13: *Solver time for the Polyhedral Exact Geometry and Stewart-Trinkle methods*

There is a single particle of mass 5 kg, starting at an initial position of $[0; 0.75]$ meters and velocity $[6.0; 1.8]$ meters per second. All surfaces have a coefficient of friction of 0.5. The resulting simulated trajectories are shown in Figure 14.

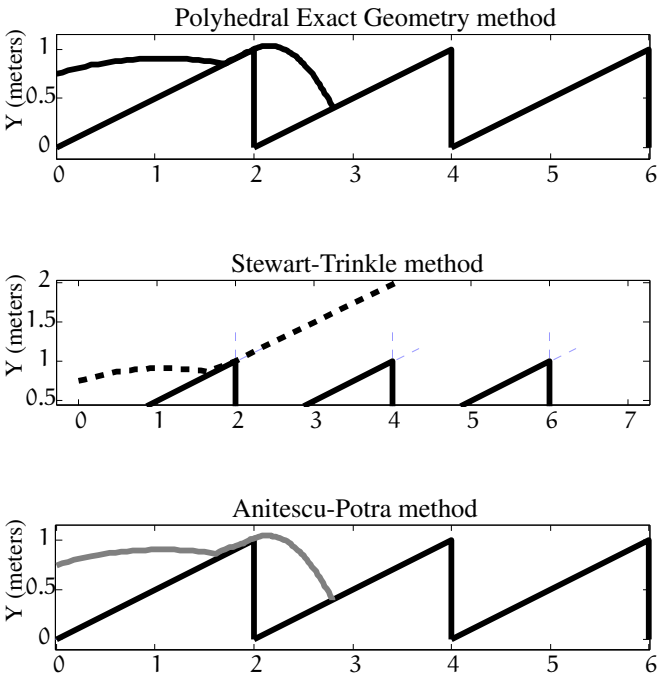

Fig. 14: *Trajectories of the sawtooth simulation with friction, using the Polyhedral Exact Geometry, Stewart-Trinkle, and Anitescu-Potra methods*

A detailed plot of the trajectories from Figure 14 is given in Figure 15. This highlights the differences present in the trajectories resulting from each method near where the particle interacts with the unilateral constraint from the sloped segment of the first sawtooth. The Polyhedral Exact Geometry method exhibits the correct behavior as verified by an analytical solution of the equations of motion. The particle in this case slides up the first edge, and then leaves for the second phase of ballistic motion.

The Stewart-Trinkle method shows the most error in the trajectory. The particle initially interacts with the constraint of the sloped edge. Constraints in this simulation are active if they are within a configured distance ϵ. As the particle slides up the slope, the constraint from the vertical edge becomes active suddenly. The constraint is violated, meaning that the particle is in penetration. A large impulse force results, which pushes the particle out at a high velocity.

The Anitescu-Potra method is added in simulation to illustrate the error present at high values of ϵ, without the large trajectory errors present while using the Stewart-Trinkle method. The Anitescu-Potra method is chosen because it behaves similar to the Stewart-Trinkle method, except that the impulsive force is not present when the constraints are active. The lack of an impulsive force prevents the particle from accumulating a large kinetic energy when constraints are erroneously violated. This behavior causes the trajectory to divert at distance ϵ near the sloped constraint.

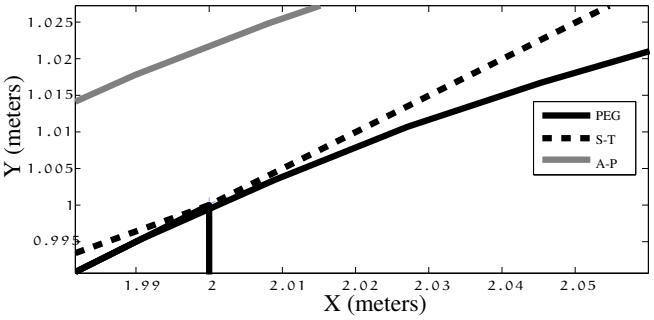

Fig. 15: *Sawtooth simulation trajectories in proximity to the beginning of the second ballistic phase*

3.3 Performance of formulations under different step sizes in planar simulations

The sawtooth simulation is run at multiple step sizes to compare the behavior of the Polyhedral Exact Geometry method and Stewart-Trinkle method under expected poor accuracy. Each simulation is run with step sizes of 10^{-4}, 10^{-3}, 10^{-2}, and 10^{-1} seconds. The same parameters as in Section 3.2 are used. Figure 16 illustrates the trajectories calculated for the two simulations. The trajectories for all step-sizes in each method are shown.

The trajectories at all step sizes are similar, except at the coarse value of 10^{-1} seconds. In these cases, the trajectory differs from the trajectories in the other runs.

A detailed view of the trajectories in Figure 16 is presented in Figure 17. An area of interest near the peak of the first sawtooth encountered by the particle. The particle leaves the surface of the object as it travels in the $+X$ direction in this region. For the cases with step sizes of 10^{-1} seconds and 10^{-2} seconds, the individual positions are marked. The square markers correspond with the trajectory with a coarse step size of 10^{-1} seconds.

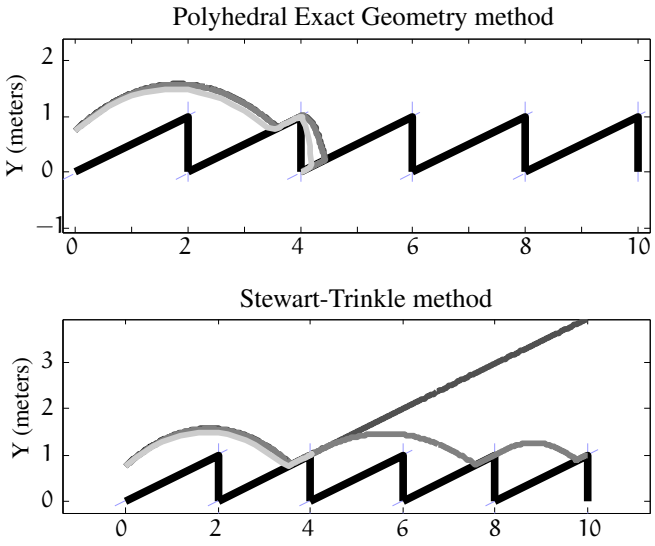

Fig. 16: *Trajectories of the Polyhedral Exact Geometry and Stewart-Trinkle methods*

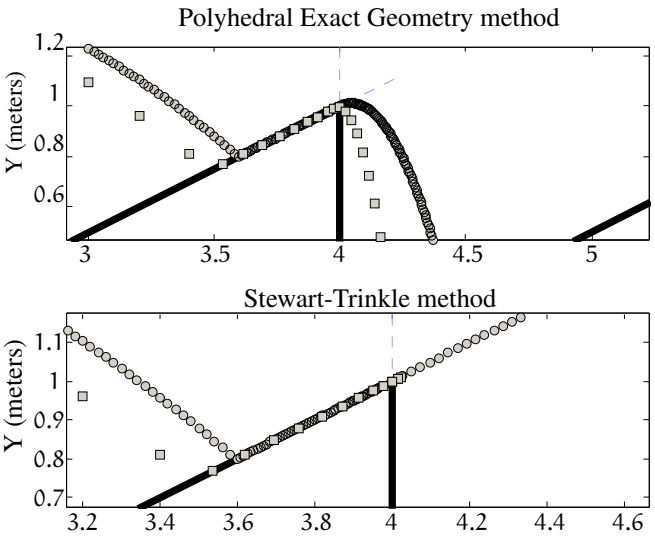

Fig. 17: *Detailed view of trajectories of the Polyhedral Exact Geometry and Stewart-Trinkle methods*

The lower subplot in Figure 17 shows a detail of the trajectories at different step sizes of the Stewart-Trinkle method. As in the Polyhedral Exact Geometry method, the coarse step size produces a trajectory that varies significantly from the trajectories of the finer step sizes. The consideration of edges only within a distance ϵ causes variation in the trajectories after the particle leaves contact with the edge of the object. The coarse trajectory gets stuck at the intersection of the two extended edges, while the fine trajectory picks up additional kinetic energy and enters the ballistic phase with a high velocity.

The sawtooth particle simulation is run multiple times with random initial states for each of the methods to eliminate any bias caused by initial conditions. The simulations start with a small step size of 10^{-5} seconds. This step size is doubled for subsequent simulations over 15 iterations until reaching a value of 0.3277 seconds.

The mean error of 62 total simulation runs with random initial states is plotted in Figure 18. The Polyhedral Exact Geometry method has two orders of magnitude less error compared to the Stewart-Trinkle method at each step size. This shows that the Stewart-Trinkle and Anitescu-Potra require a much smaller step size to achieve the same accuracy as the Polyhedral Exact Geometry method.

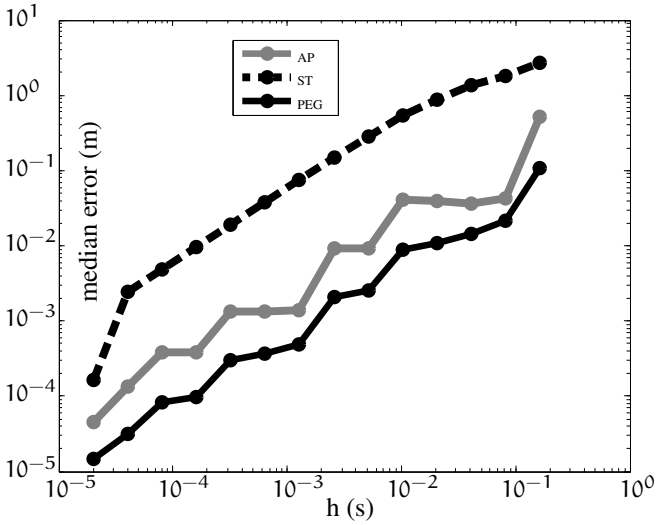

Fig. 18: *Median error in particle simulation of the Polyhedral Exact Geometry, Stewart-Trinkle, and Anitescu-Potra methods with respect to step size.*

For example, the Polyhedral Exact Geometry method can be used with a step size of 10^{-2} seconds, yet still achieve the same accuracy as the Stewart-Trinkle method with a step size of approximately 10^{-4} seconds. A significant speedup is realized here, as the problem sizes are similar for each formulation. This performance increase scales up to the general three dimensional problem consisting of many polyhedral bodies.

3.4 Slender Rod Simulation

A three-dimensional system is created to investigate the error performance of the Polyhedral Exact Geometry method against the Stewart-Trinkle and Anitescu-Potra methods. A common application envisioned for dynamic simulation is the locomotion of legged robots on rocky terrain. The leg of a robot is represented by a hexagonal prism, which approximates a cylindrical leg and is termed as a *slender rod* in this discussion.

The goal in this simulation in approximating a legged robot interacting with rocky terrain is to test the Polyhedral Exact Geometry formulation with a system that presents challenges with many areas where the freespace is non-convex. Further, additional challenges are presented by the rotation of the slender rod. This will stress the formulations

by moving features of the slender rod into violation of the constraints within larger time steps.

A profile view of the initial condition of the simulation is shown in Figure 19. The long axis of the rod is oriented along the long side of the rock field. Figure 20 gives an oblique view of the system.

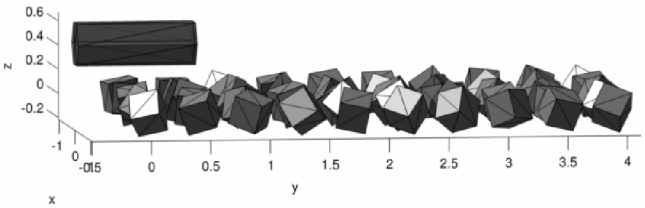

Fig. 19: *Profile view of initial configuration of slender rod simulation.*

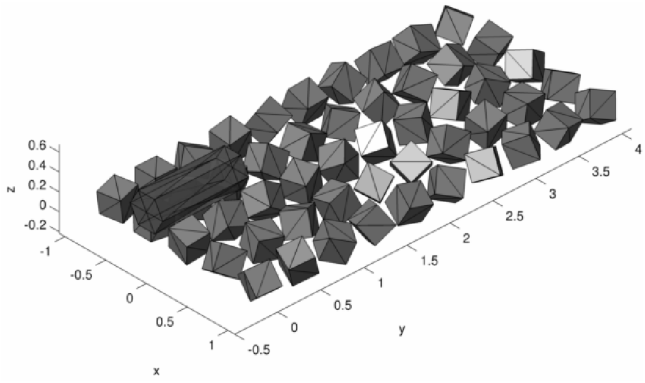

Fig. 20: *Oblique view of initial configuration of slender rod simulation.*

The plan view of the simulation is illustrated in Figure 21. The size of the field of cubes of 5 by 10 is chosen arbitrarily in order to interact with the moving slender rod for at least 1.75 seconds. The slender rod is initially centered over the edge of the field.

The rod is given an initial velocity that includes translational and rotational components. The velocity approximates the initial conditions of the rod being thrown in such a way that it tumbles during the ballistic phase before impacting the ground. The motion of the rod is simulated for 1.75 seconds for each run. The ground is comprised of identical cubes. The cubes are rigid bodies, and are massless and static. The surface is made to approximate immovable rocks, or other rough terrain. Cubes are chosen to minimize the number of active contacts during simulation, and also to maximize the areas of non-convex freespace in proximity to the ground.

The rod is idealized as a perfect cylinder when considering the mass and inertia of the body. Geometrically, it is modeled as a tessellated hexagonal cylinder for collision detection and visualization.

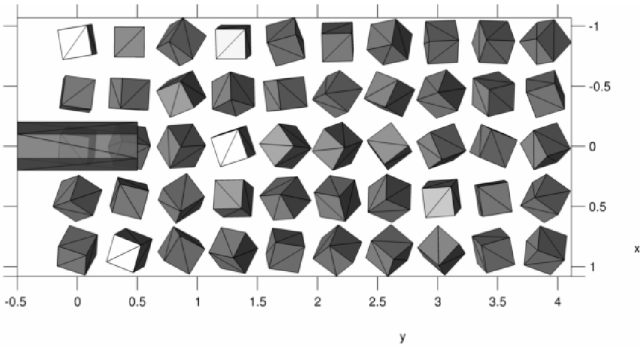

Fig. 21: *Plan view of initial configuration of slender rod simulation.*

Table 1: *Initial conditions and parameters of the slender rod*

length	1.0 m
radius	0.2 m
sides	6
density	900 kg/m^3
position	$[0, 0, 0.5]^T$ m
orientation	long axis along y
velocity	$[0, 5.0, 0.2]^T$ m/s
rotational velocity	$[-2\pi, 0, 0]^T$ s^{-1}

Each simulation run is started with the parameters and initial conditions from Table 1. The cubes are randomly oriented, and stored as initial conditions for all simulation runs. The ground truth simulation data is established as the simulation run using the Polyhedral Exact Geometry formulation at the smallest step size. This solution is considered to be the exact solution, as the exact solution cannot be readily calculated through analytical methods. The simulations for each formulation are run with a series of step sizes, starting at 5×10^{-4} seconds and doubling until reaching 0.1 seconds. Step sizes smaller than 10^{-3} seconds are impractical because of the long execution time required from the use of the RPI MATLAB Simulator.

3.5 Performance of formulations under different step sizes

Error data is compiled for simulations using each of the formulations under varying step sizes, shown in Figure 22. Multiple runs are made for each method at each step size. A group of simulation runs is defined as a run using the Polyhedral Exact Geometry formulation at the smallest step size, followed by runs of sequentially larger step sizes using all three formulations, all with the same orientations of the cubes and same parameters and initial conditions. Each

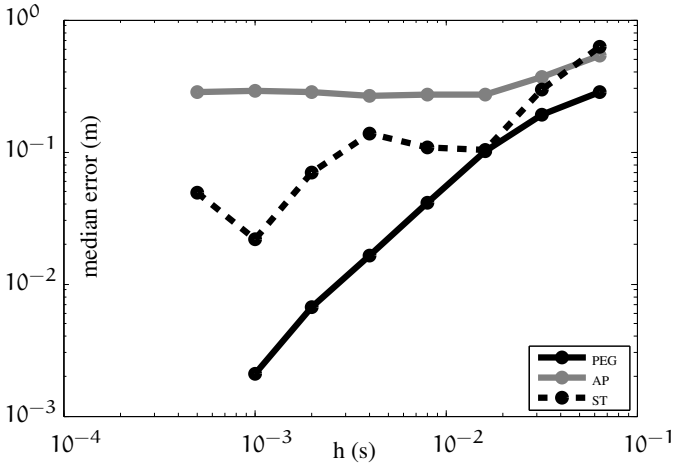

Fig. 22: *Median error in slender rod simulation of the Polyhedral Exact Geometry, Stewart-Trinkle, and Anitescu-Potra methods with respect to step size*

group of simulation runs has a different set of random orientations of the cubes, to minimize influence of the initial conditions on the error results.

The error metric is the absolute position error of the slender rod at each time step, as compared to the defined exact solution for the position at the corresponding simulation time. The error metric data is compiled for every time step of all simulations run for each method at specific step sizes. The median of the compiled error data for each run over all groups of runs is calculated, as plotted in Figure 22.

The simulation error using the Polyhedral Exact Geometry method monotonically increases as step size increases. Error in simulation using the Anitescu-Potra method is constant below 10^{-2} seconds, because of constant minimum distance ϵ. The lack of an impulse term in Equation 10 causes the rod to move within a specified distance of the static objects, without making actual contact. The error data from the Stewart-Trinkle method is not monotonically increasing with step size because of effects from large impulses in regions non-convex freespace. The simulated environment is complex such that the error metric cannot be predicted, though the general trend is that error increases with step size. The compiled error data in Figure 22 demonstrates that the Polyhedral Exact Geometry method produces a more accurate simulation at each step size compared to the Stewart-Trinkle and Anitescu-Potra methods.

Overall, the simulations utilizing the Stewart-Trinkle and Anitescu-Potra formulations here are contrived examples, with unrealistically large values for ϵ and ψ_{minimum}. This was done to illustrate the problems with contemporary formulation methods. They require fine tuning in order to work properly and produce reasonably accurate simulation results. The trajectory error in simulation for the Stewart-Trinkle and Anitescu-Potra formulations for more realistic parameters is small enough that it is manageable for most applications. The problem is that subtle errors can cause results to be invalid without notice. These errors still cause serious flaws, and are increased to a noticeable size in order to highlight the problems they can cause. Further, it is demonstrated that the Polyhedral Exact Geometry formulation minimizes the required tuning, and produces consistent results at higher accuracy regardless of parameter tuning. It obtains accurate results while allowing larger step sizes, which gives a significant performance increase as fewer time-steps are required to maintain a desired accuracy. The results presented in this section show that the Polyhedral Exact Geometry formulation outperforms the Stewart-Trinkle and Anitescu-Potra formulations, while avoiding inaccuracies in constraint representations in areas of non-convex freespace.

4 Conclusion and Future Work

This work presents an analysis of the performance of the new Polyhedral Exact Geometry constraint formulation for multibody systems represented by dynamic equations with complementarity constraints. Several shortcomings are present in the widely-used Stewart-Trinkle and Anitescu-Potra formulations, and are addressed by the Polyhedral Exact Geometry formulation. For example, physical inaccuracies are present in areas with non-convex freespace, and the Polyhedral Exact Geometry formulation produces constraints that accurately model the geometry in these regions. concepts Simulations of planar particle simulations are produced and discussed to demonstrate the core concepts of the formulations, and their varying behavior in regions of non-convex freespace. These simulations are expanded to three dimensional systems to investigate error metrics, performance, and simulation accuracy under varying step sizes and initial conditions. The overall performance of the Polyhedral Exact Geometry, Stewart-Trinkle, and Anitescu-Potra formulations are compared.

We demonstrate in our results and discussion, that the Stewart-Trinkle and Anitescu-Potra formulations have physical inaccuracies because of constraint representation not correctly modeling the actual geometry of a system. The reasons for various undesired behaviors are analyzed and explored in detail. Further, it is shown how Stewart-Trinkle and Anitescu-Potra rely on parameter tuning to produce desired results, and how deliberately poor parameter tuning can illicit major inaccuracies from small systems. We show that the Polyhedral Exact Geometry formulation performs consistently regardless of parameter values and initial conditions, and does not require tuning to produce accurate results. The Polyhedral Exact Geometry method also produces similarly accurate results at larger step sizes compared to the Stewart-Trinkle and Anitescu-Potra methods at smaller step sizes. This allows an increase in computational performance to be realized as similar results can be obtained without the requirement for small step sizes or parameter refinement and tuning.

This work is limited at this point by the tools available. The collision detection functions in publicly available simulation software do not produce the desired output needed for use with the Polyhedral Exact Geometry formulation. Collision detection in the RPI MATLAB Simulator is particularly

slow in 3D. Data produced for this paper required computational times on the order of weeks to obtain. The bottleneck in performance is in collision detection, not in constraint formulation and system solving. Collision detection in the slender rod simulation required approximately 5 seconds per time step on average.

Future work includes improvements to the performance of collision detection. This will allow more data to be gathered, and larger systems to be practically simulated. We plan to fully incorporate the collision detection functions from Bullet Physics into the RPI MATLAB Simulator. With the performance issues resolved, we plan to perform further research into the trade-offs in problem size and computational performance with the Polyhedral Exact Geometry formulation. Further work will be to evaluate the Polyhedral Exact Geometry formulation by utilizing it in other available simulation systems, and testing it on real world problems instead of benchmarks.

Acknowledgements

Thanks to Ying Lu and Sarah Niebe for their contributions to the RPI MATLAB Simulator. This work was partially supported by NSF CCF-1208468 and DARPA W15P7T-12-1-0002. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

References

[1] Rahnejat, H., 1998. *Multi-body dynamics: vehicles, machines, and mechanisms*. Professional Engineering.

[2] Sinha, R., Paredis, C., Liang, V., and Khosla, P., 2001. "Modeling and simulation methods for design of engineering systems.". *Journal of Computing and Information Science in Engineering(Transactions of the ASME),* **123**(1), pp. 84–91.

[3] Flores, P., Leine, R., and Glocker, C., 2012. "Application of the nonsmooth dynamics approach to model and analysis of the contact-impact events in cam-follower systems". *Nonlinear Dynamics,* **69**(4), pp. 2117 – 2133.

[4] Ma, Z., and Perkins, N., 2003. "An efficient multibody dynamics model for internal combustion engine systems". *Multibody system dynamics,* **10**(4), pp. 363–391.

[5] Chakraborty, N., Berard, S., Akella, S., and Trinkle, J., 2008. "An implicit time-stepping method for quasi-rigid multibody systems with intermittent contact". Vol. 5 PART A, pp. 455 – 464.

[6] Sauer, J., and Schömer, E., 1998. "A constraint-based approach to rigid body dynamics for virtual reality applications". In Proceedings of the ACM symposium on Virtual reality software and technology, ACM, pp. 153–162.

[7] Coumans, E., 2013. Bullet physics library.

[8] Lotstedt, P., 1981. "Coulomb friction in two-dimensional rigid body systems". *Zeitschrift fur Angewandte Mathematik und Mechanik,* **61**(12), pp. 605 – 15.

[9] Schiehlen, W., 1997. "Multibody system dynamics: Roots and perspectives". *Multibody system dynamics,* **1**(2), pp. 149–188.

[10] Mirtich, B., 1998. "V-clip: fast and robust polyhedral collision detection". *ACM Trans. Graph.,* **17**(3), Jul, pp. 177–208.

[11] Jiménez, P., Thomas, F., and Torras, C., 2001. "3d collision detection: a survey". *Computers & Graphics,* **25**(2), pp. 269–285.

[12] Redon, S., Kheddar, A., and Coquillart, S., 2003. "Fast continuous collision detection between rigid bodies". In Computer graphics forum, Vol. 21, pp. 279–287.

[13] Trinkle, J. C., Pang, J.-S., Sudarsky, J.-S., and Lo, G., 1997. "On dynamic multi-rigid-body contact problems with coulomb friction". *Zeitschrift fur Angewandte Mathematik und Mechanik,* **77**(4), pp. 267–79.

[14] Johansson, L., 1999. "Linear complementarity algorithm for rigid body impact with friction". *European Journal of Mechanics, A/Solids,* **18**(4), pp. 703 – 717.

[15] Baraff, D., 1994. "Fast contact force computation for nonpenetrating rigid bodies". In Proceedings of the 21st annual conference on Computer graphics and interactive techniques, ACM, pp. 23–34.

[16] Mirtich, B., and Canny, J., 1995. "Impulse-based simulation of rigid bodies". In Proceedings of the 1995 symposium on Interactive 3D graphics, ACM, pp. 181–ff.

[17] 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.

[18] 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), 8, pp. 2673 – 91.

[19] Trinkle, J., 2003. "Formulations of multibody dynamics as complementarity problems". In Proceedings, ASME International Design Engineering Technical Conferences, September.

[20] Berard, S., Trinkle, J., Nguyen, B., Roghani, B., Fink, J., and Kumar, V., 2007. "davinci code: A multi-model simulation and analysis tool for multi-body systems". In Robotics and Automation, 2007 IEEE International Conference on, IEEE, pp. 2588–2593.

[21] Bajaj, C., and Dey, T., 1992. "Convex decomposition of polyhedra and robustness". *SIAM Journal on Computing,* **21**(2), pp. 339–364.

[22] Lien, J., and Amato, N., 2007. "Approximate convex decomposition of polyhedra". In Proceedings of the 2007 ACM symposium on Solid and physical modeling, ACM, pp. 121–131.

[23] Smith, R., 2013. Open dynamics engine.

[24] Stewart, D., and Trinkle, J., 2000. "Implicit time-stepping scheme for rigid body dynamics with coulomb

friction". In Proceedings - IEEE International Conference on Robotics and Automation, Vol. 1, pp. 162 – 169.

[25] Anitescu, M., and Potra, F., 2002. "A time-stepping method for stiff multibody dynamics with contact and friction". *International Journal for Numerical Methods in Engineering,* **55**(7), 11, pp. 753 – 84.

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

[27] Nguyen, B., and Trinkle, J., 2010. "Modeling non-convex configuration space using linear complementarity problems". In Proceedings - IEEE International Conference on Robotics and Automation, pp. 2316–2321.

[28] 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.

[29] Dirkse, S., Ferris, M. C., and Munson, T., 2013. The path solver. http://pages.cs.wisc.edu/ ferris/path.html.

[30] Williams, J., Lu, Y., Flickinger, D. M., and Trinkle, J., 2013. RPI matlab simulator. http://code.google.com/p/rpi-matlab-simulator/.