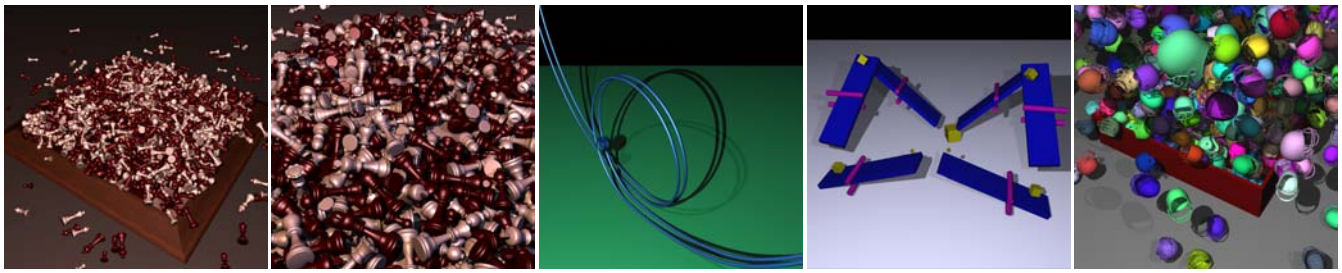
Fast Frictional Dynamics for Rigid Bodies

Danny M. Kaufman Timothy Edmunds Dinesh K. Pai

Department of Computer Science, Rutgers University



(a) (b) (c) (d) (e)

Figure 1: Examples of rigid body simulations with friction, using our approach. (a, b) 1000 chess pieces dropped in a pile, with a large number of non-convex contacts and stacking. (c) A ball with friction rolling through a loop. On the first pass through the loop the ball stays on the track. However, by the second pass, the ball's velocity has dissipated due to friction so that it falls off. (d) A complex mechanism showing frictional contact, sliding, and jamming. (e) 1000 thin walled helmets (24,308 triangles each), dropped in a pile with stacking.

Abstract

We describe an efficient algorithm for the simulation of large sets of non-convex rigid bodies. The algorithm finds a simultaneous solution for a multi-body system that is linear in the total number of contacts detected in each iteration. We employ a novel contact model that uses mass, location, and velocity information from all contacts, at the moment of maximum compression, to constrain rigid body velocities. We also develop a new friction model in the configuration space of rigid bodies. These models are used to compute the feasible velocity and the frictional response of each body. Implementation is simple and leads to a fast rigid body simulator that computes steps on the order of seconds for simulations involving over one thousand non-convex objects in high contact configurations.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling —Physically based modeling; I.6.8 [Simulation and Modeling]: Types of Simulation—Animation

Keywords: rigid bodies, non-smooth dynamics, contact, friction

1 Introduction

Rigid body models are widely used in computer games and feature films when deformation is imperceptible, or when computing and maintaining a deformation model is viewed to be an unnecessary expense. Even if the cost is not a problem, in many applications we simply do not know detailed constitutive properties or the internal states of the objects we need to simulate. In these situations, rigid body models can provide a good global description of many small scale phenomena such as localized deformation and shock propagation.

While rigid body models can provide useful simplifications, they also present unique challenges. One of the essential difficulties in the simulation of rigid bodies is the inherent "non-smoothness" of the dynamics. This is most evident when we consider that rigidity requires a body to instantaneously change its velocity whenever it collides with another surface. The strict rigid body constraints of non-interpenetration and negligible deformation make contact resolution difficult and highly sensitive to even minor variations in configuration and position. When multiple points of contact are permitted, this sensitivity can result in wildly varying behavior [Chatterjee and Ruina 1998b]. This effect is compounded when we note that any time-stepping scheme will necessarily limit the accuracy of our simulation. Time-stepping also forces rigid body methods to either allow interpenetration or to use intensive collision detection routines to find the earliest contact on every given interval. The former approach calls for the robust handling of constraint violations, while the latter approach can be quite expensive, especially if we consider the worst case scenario of an infinite sequence of bounces (consider for example a ping-pong ball coming to rest).

Imposing rigid body constraints in a reasonable manner, at low cost, while still producing plausible friction, elasticity, and momentum conservation has remained a challenge. In many cases, just determining which contacts actually contribute reaction forces to a set of bodies is difficult and subject to combinatorial explosion. Even for the case of a single non-convex rigid body interacting with a surface, a wide range of contact states are possible, making the selection of an appropriate response difficult. Conversely, because rigid bodies may be viewed as high level approximations of very stiff objects, there is no single physically accurate behavior to model. This allows for a large amount of latitude in choosing solution methods. Our goal is to develop a set of self-consistent rules governing rigid body behavior that always produce both *realistic* and *feasible* results at the fastest rates possible.

The rest of the paper is organized as follows. We briefly discuss the related work in Section 1.1, and our contributions in Section 1.2. In Section 2 we provide an overview of our approach. Section 3 shows how unilateral constraints and contact forces can be lifted into the six-dimensional configuration space of a rigid body. In Section 4 we discuss how the dynamics of a single frictionless rigid body can cope with multiple contacts. Section 5 extends this approach to deal with friction, while Section 6 presents a method for

handling a system of multiple moving bodies. In Sections 7 and 8 we discuss some details of the algorithm, and we present results in Section 9.

1.1 Related Work

Current popular approaches to the multiple contact problem with friction tend to depend upon collision ordering (propagation methods) or apply Linear Complementarity Programming (LCP). All methods however, share the common tasks of handling both the non-smoothness induced by unilateral constraints and the uncertainty of contact state [Delassus 1923].

Historically, much of the difficulty in resolving contacts has resulted from the uncertainty involved in deciding which contacting points are actually active (i.e., contributing a reaction force) at any given moment. This indeterminacy of configuration has traditionally been handled by a combinatorial approach in which all permutations of contact points are attempted until a feasible result is discovered. More recently, LCP methods have been used for this [Lötstedt 1984] (see also Baraff [1989]). These methods intelligently search the space of possible reactions for a feasible solution. They are known to provide reasonable and feasible behaviors for contact. LCP formulations have also been successfully used to simulate quasirigid objects that allow local compliance [Song and Kumar 2003; Pauly et al. 2004].

Although LCP methods are quite popular, they are not necessarily more "correct" than other approaches. In fact the velocity level LCP solution methods that have been adopted by the community to handle the solution of multiple frictional contacts have been shown to often be physically unrealistic [Chatterjee 1999]. Chatterjee and Ruina state "We know of no reason to prefer the *predictions* of an LCP-based law to those of any other law." [1998a]. We find that complementarity itself is a useful means by which to analyze contact states. However, LCP based methods are not unique in ensuring complementarity conditions [Brogliato 1999].

Moreau has proposed a contact rule that simultaneously resolves frictionless, multi-point rigid body contacts quickly and deterministically [1988]. This method decomposes the generalized velocity of a rigid body upon the tangent and normal cones of contact. It obtains a collision response that is both feasible and consistent with Gauss's principle of least constraint [Moreau 1966]. In the same work the author also proposes an extension to single point friction contacts. Redon et al. [2002] have proposed a similar velocity level treatment of Gauss's least constraint for frictionless rigid body simulation.

Consideration of friction is necessary for the realism of any physical simulation. Unfortunately, the nonlinear behavior of friction adds to the complexity of the rigid body contact problem. Initial attempts to apply LCP methods to frictional contact problems could not guarantee the existence of a solution for all cases [Baraff 1994; Trinkle et al. 1995]. This was later fixed by formulating an impulse-velocity, rather than a force-acceleration, approach to the LCP problem [Stewart and Trinkle 1996; Anitescu and Potra 1997]. For single point frictional contacts, reduced coordinate models have produced fast and realistic simulations of both sliding and rolling behaviors [Kry and Pai 2003].

Formulating rigid body dynamics at the impulse-velocity level provides a natural way of handling the impulses that are required to instantaneously change rigid body state. It was commonly held that impulses should only be applied in cases where there existed no other means to describe an interaction. In recent years there has been a general shift towards impulse-velocity approaches [Hahn 1988; Moore and Wilhelms 1988; Baraff 1991; Mirtich and Canny 1995; Guendelman et al. 2003]. Mirtich and Canny [1995] have formulated an impulse based method that treats collisions individually by backing up to the moment of first contact. Contacts and collisions are both treated as a series of many small collisions, while static contact is identified by the use of a threshold velocity.

Careful geometric analysis of the feasibility of frictional impulses is important in deriving any friction model [Mason 2001]. By considering separation rates, energy dissipation, normal impulses, and other reasonable restrictions, it is possible to define permissible regions of impulse space. Using this approach, a simple and effective collision rule has been developed for single point contacts of three-dimensional rigid bodies [Chatterjee and Ruina 1998a]. It is also tempting to formulate frictional dynamics in the configuration space of rigid bodies because of the compact representation this makes possible. Erdmann [1984] has constructed and analyzed some potential models of generalized friction cones for both the two- and three-dimensional rigid body configuration spaces.

Moreau [1994] has also formulated a Gauss-Seidel approach for multi-body simulations with friction. The approach resolves a succession of single point contacts and has been used to simulate large granular flow effects such as size-segregation, on sets of spheres as large as fourteen thousand in several weeks time.

Milenkovic and Schmidl [2001] apply a Quadratic Program (QP) with non-physically based position constraints to bring their rigid bodies as close as possible to a natural configuration determined by a non-contacting forward Euler step. With the aid of a freezing technique, the authors were able to simulate one thousand cubes falling down an hour glass in one and a half days [Schmidl and Milenkovic 2004].

Guendelman et al. [2003] use a novel propagation method that modifies the traditional ordering of collision and contact resolution. They apply collision resolution locally by treating each pairwise penetration independently. The order of processed bodies is chosen randomly, while the deepest penetrating point on each body is treated first. Contacts are treated similarly. This relaxation approach is similar to Moreau's Gauss-Seidel method. Their algorithm handles non-convex mesh based objects. Simulations of dropping one thousand rings averaged seven minutes a frame to compute. Likewise, a more complex simulation of five hundred bones also took approximately seven minutes a frame.

1.2 Our Contributions

We introduce a new velocity level algorithm for the simulation of large sets of non-convex rigid bodies. By avoiding repeated pairwise comparisons between bodies, we obtain a complexity linear in the total number of contacts detected in each iteration. We employ a novel contact model that uses mass, location, and velocity information from all contacts, at the moment of maximum compression, to constrain rigid body velocities. We also develop a new friction model in the configuration space of rigid bodies. Based on the principle of maximal dissipation, it unifies rolling and sliding friction in a natural manner, while producing expected behaviors. These models compute the feasible velocity and the frictional response for all bodies, by using two convex, separable QPs per body. Both of these QPs are linear in the number of contacts detected for each body. For further speedup we approximate these QPs using a simple and effective constraint satisfaction method. Taken together, these results lead to a fast, efficient rigid body simulator that can easily be integrated with a variety of broad and narrow phase collision detection systems. We obtain many complex behaviors such as rolling, sliding, stacking, tumbling, and shock propagation (as well as the appropriate transitions between them) without varying our algorithm.

2 Overview

We have started with Moreau's [1988] work on resolving multiple frictionless contacts for a single inelastic rigid body system with

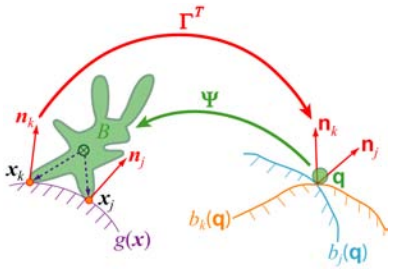


Figure 2: A unilateral constraint in $\mathbb{R}^3$ on the left and its corresponding $SE(3)$ constraints on the right.

fixed constraints. One feature of his approach is the treatment of all interpenetrations, detected in an initial half-step, as contacts to be resolved during a second half-step. A similar approach to penetration was adopted by Guendelman et al. [2003]. We extend these results by developing a new generalized friction model for multiple contacts that applies a maximally dissipative frictional impulse. This model incorporates both rolling and sliding friction. It gives us realistic behavior for a single non-convex rigid body in contact with multiple fixed constraints.

For the multi-body setting, we treat each body as an independent system in which all other moving bodies impose velocity level affine constraints. To the best of our knowledge this is a new approach for handling the multi-body problem. It is effective because we incorporate first-order contact information for each body, and so obtain velocities that produce non-penetrating behavior (at least locally) in the next step. We also impose approximate momentum conservation for many simultaneous contacts. For this, we compute an affine constraint for each contact using the velocity of the contact at the moment of maximal compression. Taken together this gives us a multi-body simulation algorithm that is linear in the total number of contacts processed for all bodies at each step.

We also introduce several practical optimizations to further improve performance. We reduce the number of contacts treated to both increase the speed and also to simplify implementation. We filter out many initially detected contacts by ignoring satisfied constraints (i.e., contacts from which a body is moving away at a sufficient velocity). Additionally, we introduce further speed-up in our algorithm by approximating the two QPs that arise in each step by the Surrogate Constraint Method [Yang and Murty 1992] (see Section 7).

Our development requires some basic concepts from kinematics and differential geometry. We have included an appendix that provides some background concepts and notation as well as a reference for further information. We suggest that the reader skim this appendix before reading the rest of the paper, and return to it if some concepts are not clear. In the following we reserve the bold sans serif font for spatial variables (i.e., six-vectors representing orientation and position of an object). Scalar values and $\mathbb{R}^3$ vectors are indicated by *italicized roman font*, while $\mathbb{R}^3$ vectors will be additionally distinguished as ***bold***. For clarity of notation we will not index inertia matrices (denoted M) unless the usage is unclear. We use leading superscripts, ${}^{i}\mathsf{v}$, to indicate spatial velocities (called *twists*), while leading subscripts, ${}_{i}\mathsf{v}$, denote spatial forces (called *wrenches*).

3 Unilateral Constraints and Forces

In practical terms, it is difficult for rigid body simulators to strictly enforce non-interpenetration. This effort may be unwarranted, as it is unreasonable to expect a "real" body to be entirely rigid all the time. Strict rigid body constraints may, in some senses, be over-constraining, since there is no reason to expect realistic behavior if we do try to enforce them fully.

We adopt Moreau's strategy of living with interpenetration. In other words we don't mind interpenetration in the middle of each step; we simply wish to minimize it at either end. Unlike Moreau, we ground this approach by viewing rigid bodies as approximations of real objects rather than a mathematical ideal. From this perspective, interpenetrations detected in the middle of a time-step are simply approximations of the deformed object-object boundary at some moment during the time-step. It is then reasonable to treat all of these detected interpenetrations as contacts. This allows us to use *all* contact information to get a better approximation of the state of our very stiff "real world" body.

We use this contact information to generate first-order constraints on each body's configuration (i.e., position and orientation). Each iteration begins by taking a configuration half-step using the last known feasible velocity of each body. We then perform collision detection. The set of detected contacts (this includes interpenetrations) are then used to form a set of constraints on the next velocity that each rigid body may take (over the next two half-steps).

In order to simplify our presentation, this section, and the following two, will assume that all constraints we encounter are fixed. Then, starting with Section 6, and throughout the rest of the paper, we discuss how to handle potential constraint motion.

3.1 Constraints in $\mathbb{R}^3$ and $SE(3)$

Given a rigid body B, we describe it by the set of points it occupies in $\mathbb{R}^3$. A unilateral (inequality) constraint acting on B in $\mathbb{R}^3$ is defined by

$$g(\boldsymbol{x}) \geq 0,\ \forall \boldsymbol{x} \in B \subset \mathbb{R}^3. \tag{1}$$

Here $g(\boldsymbol{x})$ is a real valued function and could, for instance, implicitly represent a surface constraint. We equivalently describe B in its configuration space, $SE(3)$. In this six-dimensional space of positions *and* orientations, we may treat B as a single mass point without extent. This is a useful perspective that we will employ throughout the paper. However, we must be careful since $SE(3)$ is a curved space. Here concepts like convexity and distance are difficult. Constraints on B in $SE(3)$ are described as

$$b(\mathsf{q}) \geq 0,\ \mathsf{q} \in SE(3), \tag{2}$$

where q indicates B's current configuration. We note that each individual contact of B with a constraint in $\mathbb{R}^3$ is equivalent to a different $SE(3)$ constraint (see Figure 2). This, in part, helps to explain why resolving contact conditions can be so challenging for seemingly simple rigid body objects. Large numbers of contact constraints in $\mathbb{R}^3$ introduce highly complex constraint hypersurfaces in $SE(3)$. Defining their curvature, especially at locations of intersecting constraints, would be very difficult.

Instead we prefer to work with the gradients of our constraints in $SE(3)$. These gradients exist in a linear vector space $se^*(3)$, which is dual to the linear vector space of rigid body velocities, $se(3)$. In both $se(3)$ and $se^*(3)$ concepts like convexity and projection are easier to formulate. For our purposes, we assume that each constraint in $SE(3)$ is the result of a single contact at $\boldsymbol{x}_k \in \mathbb{R}^3$ and can be described by an inequality, $b_k(\mathsf{q}) \geq 0$. Given a contact at $\boldsymbol{x}_k$, we obtain the corresponding $se^*(3)$ constraint gradient, $\nabla b_k(\mathsf{q})$, by multiplying the $\mathbb{R}^3$ constraint gradient, $\nabla g(\boldsymbol{x}_k)$, by the differential transpose matrix, ${\Gamma_k}^T$ (see Figure 2):

$$\nabla b_k(\mathsf{q}) = {\Gamma_k}^T \nabla g(\boldsymbol{x}_k) = \begin{pmatrix} [\boldsymbol{x}_k] \\ I \end{pmatrix} \nabla g(\boldsymbol{x}_k). \tag{3}$$

Generally, $\nabla g(\boldsymbol{x}_k)$ is the $\mathbb{R}^3$ contact normal of the contacting surface at $\boldsymbol{x}_k$, while "$[\ ]$" denotes the matrix of the cross product (see Appendix). In this form, contact constraints can be used to place bounds on locally acceptable velocities.

3.2 Constraints and Contact Forces

In this section we discuss the set of potential forces that can be imposed on a rigid body by a given contact. These forces are translated into velocity vectors that indicate the change in velocity that each force could apply to the body. This allows us to directly compare contact forces and velocity constraints, and will later be useful for choosing appropriate contact impulses.

A contact in $\mathbb{R}^3$ can impart a normal and tangential force. We find the span of these forces and convert them into wrenches (i.e., spatial forces in $se^*(3)$). Wrenches let us describe forces independently of the contact points that generate them. We assume all values are defined in a body frame i. Normal forces, induced by a contact with the constraint at $\boldsymbol{x}_k$, lie in the direction of the constraint gradient:

$$\boldsymbol{n}_k \stackrel{\text{def}}{=} \frac{\nabla g(\boldsymbol{x}_k)}{||\nabla g(\boldsymbol{x}_k)||}, \tag{4}$$

while tangential forces along the constraint surface span the constraint's tangent plane at $\boldsymbol{x}_k$. Their directions can be implicitly defined by the set of vectors orthogonal to $\boldsymbol{n}_k$:

$$\boldsymbol{S}_k = \{\boldsymbol{s}_k \in \mathbb{R}^3 : {\boldsymbol{s}_k}^T \boldsymbol{n}_k = 0\}. \tag{5}$$

The wrenches generated by applying these forces at $\boldsymbol{x}_k$, give us the normal force direction in $se^*(3)$,

$${}_i\mathsf{n}_k = {}_i\mathsf{\Gamma}_k{}^T \boldsymbol{n}_k = \begin{pmatrix} [\,{}^i\boldsymbol{x}_k] \\ I \end{pmatrix} \boldsymbol{n}_k, \tag{6}$$

and the set of tangent force directions in $se^*(3)$,

$${}_i\mathsf{S}_k = {}_i\mathsf{\Gamma}_k{}^T \boldsymbol{S}_k = \begin{pmatrix} [\,{}^i\boldsymbol{x}_k] \\ I \end{pmatrix} \boldsymbol{S}_k. \tag{7}$$

One of the chief difficulties of non-smooth dynamics is in correlating the forces imparted to rigid bodies with the velocity level information needed to determine these forces. We see this when resolving unilateral contact constraints, as well as in the velocity dependence inherent in friction models. For these cases, it helps us to embed our wrenches into the velocity space $se(3)$. We do this by multiplying our wrench by the inverse inertia tensor at q to get a twist (i.e., a spatial velocity),

$${}^i\mathsf{f} = \mathsf{M}(\mathsf{q})^{-1}\,{}_i\mathsf{f}. \tag{8}$$

Using Equations 6, 7 and 8, we obtain the twists ${}^i\mathsf{n}_k$ and ${}^i\mathsf{s}_k \in {}^i\mathsf{S}_k$. Unlike their counterparts in $\mathbb{R}^3$ ($\boldsymbol{n}_k$ and $\boldsymbol{s}_k$), ${}^i\mathsf{n}_k$ and ${}^i\mathsf{s}_k$ are not generally orthogonal (see Figure 3). This can be seen if we note that the inner product on $se(3)$ (see Appendix) for these two vectors is

$$\begin{aligned} {}^i\mathsf{n}_k^T \mathsf{M}\, {}^i\mathsf{s}_k &= (\mathsf{M}^{-1} \begin{pmatrix} [\,{}^i\boldsymbol{x}_k]\boldsymbol{n}_k \\ \boldsymbol{n}_k \end{pmatrix})^T \;\mathsf{M}\; (\mathsf{M}^{-1} \begin{pmatrix} [\,{}^i\boldsymbol{x}_k]\boldsymbol{s}_k \\ \boldsymbol{s}_k \end{pmatrix}) \\ &= \boldsymbol{n}_k^T\,([\,{}^i\boldsymbol{x}_k]^T \mathscr{I}^{-1} [\,{}^i\boldsymbol{x}_k] + \mathscr{M}^{-1})\,\boldsymbol{s}_k. \end{aligned} \tag{9}$$

Clearly there is no reason to expect this inner product to preserve orthogonality from $\mathbb{R}^3$. This means that if we want to construct a friction cone in $se(3)$, we must be careful not to assume orthogonality, as is commonly done for friction cones in $\mathbb{R}^3$.

3.3 Multi-point Contact

Following Moreau, we extend our definition of contact to include points that actually penetrate our constraints. This requires that our constraint function, $g(\boldsymbol{x})$, be constructed in such a way that $\nabla g(\boldsymbol{x})$ is nonzero in the neighborhood of $g(\boldsymbol{x}) = 0$. For every configuration, q, of a rigid body, our collision detection system finds the set of contacts:

$$C(\mathsf{q}) = \{k \in \mathbb{Z} : b_k(\mathsf{q}) \leq 0\}. \tag{10}$$

Then for all configurations q, we can define a normal cone of embedded wrenches that spans the entire range of possible contributions of normal forces from all points of contact,

$$\mathsf{N}(\mathsf{q}) = \Big\{ \sum_{k \in C(\mathsf{q})} \lambda_k\ {}^i\mathsf{n}_k,\ \lambda_k \geq 0 \Big\}. \tag{11}$$

Likewise, we define a sliding cone of embedded wrenches that spans the entire range of possible contributions of tangential forces from all points of contact:

$$\mathsf{S}(\mathsf{q}) = \bigoplus_{k \in C(\mathsf{q})} {}^i\mathsf{S}_k, \tag{12}$$

where $\bigoplus$ indicates a direct sum that gives us the span of all ${}^i\mathsf{S}_k$.

Because ${}^i\mathsf{n}_k$ is the gradient of a contact constraint embedded into the velocity space, it defines a constraint on acceptable velocities. For a twist, ${}^i\boldsymbol{\phi}$, to be a feasible velocity, given a contact at $\boldsymbol{x}_k$, its projection along ${}^i\mathsf{n}_k$ must not be less than 0. For the inner product in $se(3)$ we use ${}^i\boldsymbol{\phi}^T\,\mathsf{M}(\mathsf{q})\,{}^i\mathsf{n}_k$. This implicitly defines the subspace of feasible twists, given a contact at $\boldsymbol{x}_k$:

$$\mathsf{T}_k = \{\mathsf{t} \in se(3) : \mathsf{t}^T\,\mathsf{M}(\mathsf{q})\ {}^i\mathsf{n}_k \geq 0\}. \tag{13}$$

This subspace is called a tangent cone and defines a half-space constraint through the origin of $se(3)$. The intersection of these half-spaces defines a composite tangent cone (dual to $\mathsf{N}(\mathsf{q})$) that defines the subspace of locally feasible velocities in $se(3)$ at q, given the set of all current contacts, $C(\mathsf{q})$:

$$\mathsf{T}(\mathsf{q}) = \bigcap_{k \in C(\mathsf{q})} \mathsf{T}_k. \tag{14}$$

To keep these definitions meaningful for non-contacting cases, if $C(\mathsf{q}) = \emptyset$, we let $\mathsf{N}(\mathsf{q}) = 0$ and $\mathsf{T}(\mathsf{q}) = se(3)$.

While T_k spans a six-dimensional half-space, ${}^i\mathsf{S}_k$(because it is parameterized by a tangent plane in $\mathbb{R}^3$) is simply a two-dimensional plane in $se(3)$ (see Figure 3). In general, the composite tangent cone, $\mathsf{T}(\mathsf{q})$, is not identical to the sliding cone, $\mathsf{S}(\mathsf{q})$. This means that the space of feasible velocities is not spanned by the space of embedded tangential wrenches, S, and that in some cases, portions of S may actually lie outside of this feasible space [Erdmann 1984]. This is a direct result of the non-orthogonality described at the end of the previous section and will be important later, in developing our friction model.

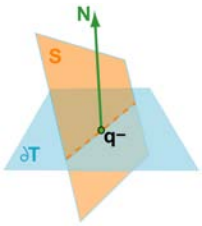


Figure 3: A single contact normal with its induced tangent cone and sliding plane in $se(3)$.

4 Non-smooth Dynamics

In general, in the absence of contacts, the dynamics of a rigid body i in its own frame are determined by the ODE:

$$\mathsf{M}\,{}^i\dot{\boldsymbol{\phi}} = {}_i\mathsf{f} + {}_i\mathsf{c}. \tag{15}$$

Here ${}^i\phi \in se(3)$ is the local body frame twist of rigid body i, with ${}^i\dot{\phi}$ as its time derivative, ${}_i\mathsf{f}$ is the net force due to non-contact phenomena such as gravity, M is the fixed, body-frame inertial tensor, and ${}_i\mathsf{c}$ is the coriolis term (${}_i\mathsf{c} = ({}^i\phi_\times)^T \mathsf{M}\, {}^i\phi$).

If there are contacts (that is, if $C(\mathsf{q}) \neq \emptyset$), Equation 15 must be modified to include a possible reaction force. Leaving consideration of friction aside for the moment, we can temporarily ignore the possibility of tangential force contributions. Then a reaction force, induced by the active contact constraints, must come from a cone defined by the convex combination of wrenches ${}_i\mathsf{n}_k$. This allows us to describe the system's dynamics by the differential inclusion:

$$\mathsf{M}\,{}^i\dot{\phi} - {}_i\mathsf{f} - {}_i\mathsf{c} \in \Big\{ \sum_{k \in C(\mathsf{q})} \lambda_k\, {}_i\mathsf{n}_k,\ \lambda_k \geq 0 \Big\}, \tag{16}$$

or, embedding the inclusion in se(3), we equivalently find

$${}^i\dot{\phi} - {}^i\mathsf{f} - {}^i\mathsf{c} \in \mathsf{N}(\mathsf{q}). \tag{17}$$

The embedded form of the inclusion is more useful, because it allows us to directly correlate potential contact impulses with the locally feasible set of spatial velocities given by the tangent cone $\mathsf{T}(\mathsf{q})$.

The non-contacting motion of a free body is continuous. However, in the event of a contact, with either a fixed or moving constraint, there will be a non-smooth transition in state. This motivates us to formulate our problem in a discrete domain. Using a first order discretization with a step size h we obtain the equation:

$$({}^i\phi^{t+1} - {}^i\phi^t) = h({}^i\mathsf{f} + {}^i\mathsf{c}) + {}^i\mathsf{r},\ {}^i\mathsf{r} \in \mathsf{N}(\mathsf{q}) \tag{18}$$

Note that scalar multiplication does not change a cone. Equivalently we may describe the above implicit relationship using a pre-resolution velocity defined as

$${}^i\phi^- \stackrel{\text{def}}{=} {}^i\phi^t + h({}^i\mathsf{f} + {}^i\mathsf{c}), \tag{19}$$

and a post-resolution velocity,

$${}^i\phi^+ \stackrel{\text{def}}{=} {}^i\phi^{t+1} = {}^i\phi^- + {}^i\mathsf{r}. \tag{20}$$

These two velocities are essential for our half stepping algorithm. The pre-resolution twist, ${}^i\phi^-$, is the velocity of our rigid body before collision detection is applied and ${}^i\phi^+$ is the velocity after all contacts are processed. Similarly q^t and q^{t+1} are the rigid body's configuration at the beginning and end of step t, while q^- is its configuration midway through this step.

4.1 Contact Resolution

The above inclusion (Equation 18), is still somewhat indefinite. It indicates the subspace of $se(3)$ that the normal reaction, ${}^i\mathsf{r}$, must belong to, but it does not tell us how to actually select ${}^i\mathsf{r}$. We require some sort of selection rule that maps spatial velocities onto the normal cone. Moreau [1988] defines a discrete, frictionless, inelastic contact rule, where ${}^i\mathsf{r}$ is defined as the twist formed by the minimum spanning vector in $se(3)$ between the pre-resolution velocity, ${}^i\phi^-$, and the subspace of feasible velocities, $\mathsf{T}(\mathsf{q})$. This twist can be found by projecting $(-{}^i\phi^-)$ onto $\mathsf{N}(\mathsf{q})$:

$${}^i\mathsf{r} = \text{proj}_{\mathsf{N}(\mathsf{q})}(-{}^i\phi^-). \tag{21}$$

Intuitively, ${}^i\mathsf{r}$ corresponds to the feasible impulse that opposes the constraint violating component of the pre-resolution velocity. Using some classical results from convex analysis (see [Moreau 1962; Rockafellar 1970]), it can be shown that applying this selection rule

Figure 4: A complex mechanism demonstrating frictional contact, sliding, momentum transfer, and jamming.

to obtain ${}^i\mathsf{r}$ is equivalent to directly projecting ${}^i\phi^-$ onto the subspace of feasible velocities, $\mathsf{T}(\mathsf{q}^-)$, to obtain ${}^i\phi^+$:

$${}^i\phi^+ = \text{proj}_{\mathsf{T}(\mathsf{q})}({}^i\phi^-) \in \partial\mathsf{T}(\mathsf{q}). \tag{22}$$

Here $\partial\mathsf{T}(\mathsf{q})$ indicates the boundary of the feasible subspace, $\mathsf{T}(\mathsf{q})$. This implies that under Moreau's rule, post-resolution velocities will be made feasible by ensuring that ${}^i\phi^+$ always lies in T, even when ${}^i\phi^-$ does not. More specifically, this selection rule imposes the condition that all post-resolution velocities lie on the boundary. Because of this, ${}^i\mathsf{r}$ may be considered a completely inelastic embedded impulse – a useful result that will come in handy when we compose our friction model.

We now can refine our contact set, $C(\mathsf{q})$. Satisfied contact constraints from which our body is moving away (i.e., k such that ${}^i\mathsf{n}_k^T\ \mathsf{M}\ {}^i\phi^- > 0$) are discarded, and a new set of potentially active contacts can be defined as $A(\mathsf{q}) = \{k \in C(\mathsf{q}) : \mathsf{n}_k^T\ \mathsf{M}\ {}^i\phi^- \leq 0\}$. This reduces the cost of our computations and simplifies later results.

When we obtain ${}^i\phi^+$ directly from the projection in Equation 22, we may assume that it was obtained by an implicit inelastic reaction, ${}^i\mathsf{r} \in \mathsf{N}(\mathsf{q})$, due to the equivalence between Equation 22 and Equations 20 and 21.

5 Friction

We extend Moreau's selection rule to include a tangential frictional response. To do this, we have formulated a new generalized friction model that does not require explicit point information once our contact wrenches have been generated. This approach encodes $\mathbb{R}^3$ contact information into $se(3)$, allowing us to determine the total contribution of all active contacts, without regard to their origin, via a single projection operation. Unlike the popular LCP friction models, the solution of our multi-point friction model reduces to a small, *convex*, separable QP.[1] This gives us a relatively small solution cost by nonlinear optimization standards – linear in the number of contacts per rigid body (see Section 7 for more details).

Our generalized friction model is derived from the principle of maximal dissipation and is motivated by Goyal et al.'s [1991] approach to resolving frictional planar sliding. It combines both linear and angular friction into a single unified response. The model produces the familiar sliding, tumbling, and rolling behaviors produced by friction (see Figures 1(c,d) and 4) and reduces to the traditional Coulomb model for a point mass in $\mathbb{R}^3$. The single point Coulomb model of friction states that a frictional impulse should be both in opposition to the tangential velocity direction, and proportional to forces applied in the normal direction. Our model maintains this relationship. This requires us to be able to correlate force and velocity level information. Embedding our wrenches in the linear velocity space, $se(3)$, allows us to form a direct relation between these two disparate quantities.

[1] See [Anitescu and Hart 2004] for a discussion of the potential non-convexity of LCP equivalent QPs.

We first modify Equation 18 so that contacts can generate both normal *and* tangential impulses:

$$ {}^{i}\phi^{t+1} - {}^{i}\phi^{t} = h\,({}^{i}\mathsf{f} + {}^{i}\mathsf{c}) + {}^{i}\mathsf{r} + {}^{i}\delta,\ {}^{i}\mathsf{r} \in \mathsf{N}(\mathsf{q}),\ {}^{i}\delta \in \mathsf{S}(\mathsf{q}). \quad (23) $$

Here ${}^{i}\mathsf{r}$ is still the change in spatial velocity induced by the active contact constraints at q, while the new quantity, ${}^{i}\delta$, is an embedded frictional impulse that opposes sliding along the constraint tangent planes in $\mathbb{R}^3$.

For all of the following results, we need to further restrict our sliding twists, ${}^{i}\mathsf{s}_k$, to be generated from unit tangent vectors in $\mathbb{R}^3$. This simply means that we normalize our tangent vectors at each contact, so that ${}^{i}\mathsf{s}_k = \mathsf{M}^{-1}\,{}_{i}\Gamma_k{}^{T}\,\boldsymbol{s}_k$, where $\boldsymbol{s}_k^T\boldsymbol{n}_k = 0$, *and* $||\boldsymbol{s}_k|| = 1$.

5.1 Frictional Impulse Constraints

For each generated sliding twist, ${}^{i}\mathsf{s}_k$, we define a generalized frictional coefficient μ_k. This allows us to produce frictional behavior that can both be anisotropic and vary with the location of the contact. We also let ${}^{i}\phi^{\tau}$ be the twist computed by our first projection in Equation 22.

We have already determined (via Moreau's rule) how to choose our normal impulse, ${}^{i}\mathsf{r}$. This normal impulse, and the set of frictional coefficients, μ_k, are used to determine our frictional impulse, ${}^{i}\delta$. As it is currently constructed, our sliding cone, S, contains all potential frictional reaction directions (embedded into velocity space). It does not, however, place any constraints on the magnitude of a selected impulse in S. We use a limit surface, a tool from plasticity theory that has been introduced for friction by Goyal et al. [1991], to bound the magnitude of our embedded frictional impulse. We will first discuss how to proceed if we know the normal distribution of our inelastic impulse, and then show how we obtain frictional behavior without it.

If we know the distribution of ${}^{i}\mathsf{r}$ across all active normals, then at each active contact k, we have available a scalar, $\alpha_k \geq 0$, that gives us the magnitude of each normal's contribution to the total normal impulse (i.e., ${}^{i}\mathsf{r} = \sum_{k\in A(\mathsf{q})} \alpha_k\ {}^{i}\mathsf{n}_k$). For each sliding plane, S_k, this allows us to construct a limit curve, using a set of generalized Coulomb inequalities: ${}^{i}\mathsf{s}_k{}^{T}\,\mathsf{M}\ {}^{i}\delta \leq \mu_k\,\alpha_k$. The area bounded by this curve in S_k contains all permissible contributions from contact k. If we take the direct sum of all such areas, we obtain a subset of $\mathsf{S}(\mathsf{q})$ that contains all potential frictional impulses. The boundary of this region describes a limit surface that indicates, for each direction in $\mathsf{S}(\mathsf{q})$, the maximum possible magnitude of a frictional impulse in this direction, given the known normal distribution.

Because we obtain ${}^{i}\mathsf{r}$ implicitly from our projection onto T, and because there are often redundancies in our constraint normals, obtaining a normal distribution can be difficult and also costly. However, each contact point is an approximation of a deforming contact region and so strict enforcement of Coulomb's inequality (an approximation itself) can be overly restrictive. We place a reasonable bound on the limit surface using the inequalities: ${}^{i}\mathsf{s}_k{}^{T}\,\mathsf{M}\ {}^{i}\delta \leq \mu_k\ {}^{i}\mathsf{n}_k{}^{T}\mathsf{M}\ {}^{i}\mathsf{r}$, for each active contact k. We denote each such constraint by $\mathsf{S}(\mathsf{s}_k, \mu_k, \mathsf{n}_k, \mathsf{r})$.

We additionally need to constrain our frictional impulse to lie in a feasible direction. Because of the non-orthogonality described in Section 3, portions of the sliding cone $\mathsf{S}(\mathsf{q})$ can actually lie outside the feasible cone T. We also know (from Section 4.1) that ${}^{i}\phi^{\tau}$ lies on the boundary of the feasible region of rigid body velocities. Thus, any embedded impulse applied to ${}^{i}\phi^{\tau}$ that lies outside T, will generate a new velocity that is also outside of T and therefore not feasible.

5.2 Maximal Dissipation and Projection

The intersection of our limit surface and the tangent cone $\mathsf{T}(\mathsf{q})$, defines a convex subset of $se(3)$, that contains our desired embedded frictional impulse, ${}^{i}\delta$. As a selection rule for ${}^{i}\delta$ in S, we choose a maximally dissipative impulse. This is the impulse (formed by forces applied along the contact constraints' tangent planes in $\mathbb{R}^3$) in S that most closely opposes tangential motion (i.e., sliding and rolling along $\partial\mathsf{T}$) up to the maximum magnitude permissible in each direction. Choosing such an impulse, from a limit surface composed of the direct sum of all local limit curves, gives us the sum of all maximally dissipative impulses generated at each active contact. Because ${}^{i}\mathsf{r}$ is inelastic, we know that ${}^{i}\phi^{\tau}$ is moving along the local constraint boundary ($\partial\mathsf{T}$), and is the tangential motion we wish to oppose. Since $(-\,{}^{i}\phi^{\tau})$ would clearly maximally dissipate this tangential motion, we want to find the twist in S that is closest to it. It can be found by projecting $(-\,{}^{i}\phi^{\tau})$ onto the subset formed by the intersection of our limit surface and $\mathsf{T}(\mathsf{q})$. This projection can be solved by a separable, convex QP,

$$ \begin{aligned} &{}^{i}\delta = \underset{\mathsf{y}}{\operatorname{argmin}} ||\,\mathsf{y} + {}^{i}\phi^{\tau}||^2 \\ &\text{subject to: } y \in \mathsf{S}. \\ &{}^{i}\mathsf{s}_k{}^{T}\ \mathsf{M}\ \mathsf{y} \leq \mu_k\ {}^{i}n_k{}^{T}\mathsf{M}\ {}^{i}\mathsf{r},\ \forall\ {}^{i}\mathsf{s}_k \in\ {}^{i}\mathsf{S}_k,\ \forall k \in A(\mathsf{q}). \\ &{}^{i}\mathsf{n}_k{}^{T}\ \mathsf{M}\ \mathsf{y} \geq 0,\ \forall\ k \in A(\mathsf{q}). \end{aligned} \quad (24) $$

Finally, we obtain our post-resolution velocity by adding our frictional reaction to the tangential velocity: ${}^{i}\phi^{+} = {}^{i}\delta + {}^{i}\phi^{\tau}$. In total, this amounts to two projections per rigid body: firstly, a projection onto the feasible space of velocities defined by the composite tangent cone, and secondly, the application of an oppositional frictional impulse, obtained by a second projection onto S proportional in magnitude to the first projection's impulse.

This gives us the essential properties expected from a frictional law; a purely positive dissipation, sticking, sliding, and a bound at each contact point that, at least approximately, enforces a relationship between a maximum frictional contribution and the local normal impulse.

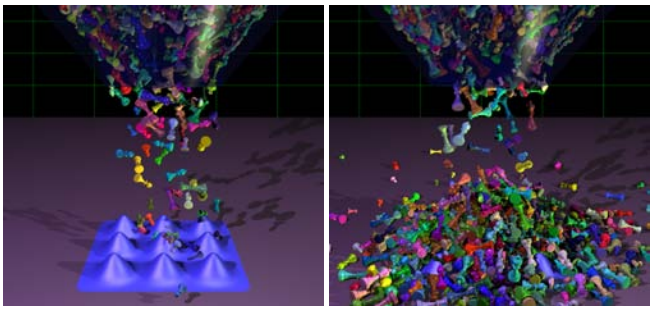


Figure 5: A many-body simulation of 2016 chess pieces falling through a hopper and stacking.

6 Multiple Rigid Bodies

This section describes our method for handling simulations of large sets of rigid bodies (see Figure 5). To resolve contacts, each body is processed a single time per step, while each step is linear in the total number of contacts. These bounds are maintained by treating each body as an independent system, carefully using the motion of all other contacting bodies to generate velocity level constraints. Because other bodies may be moving, and therefore can impose moving constraints, Moreau's solution for fixed constraints (Section 4.1) is no longer sufficient for resolving contacts. We address this by adding an offset to each velocity level contact constraint, which takes into account the velocity of the contacting body. This leads to a more general, but still convex, QP for computing the tan-

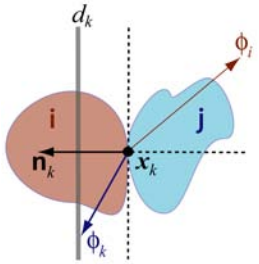


Figure 6: Body j imposing a moving constraint upon body i. Their common contact point, $\boldsymbol{x}_k$, with the normal n_k, is given a velocity of ϕ_k . Body i's post-resolution velocity must then be greater than d_k, when projected onto n_k.

gential velocity. In addition, we show how momentum conservation and elastic restitution can be obtained within this model.

6.1 Moving Rigid Body Constraints

In Section 4.1 we implicitly assumed that our constraints did not change over the span of each time step. This allowed us to center our velocity constraints at the origin of $se(3)$, since these constraints would only affect velocity direction, not magnitude. Now, to compensate for constraint change over each interval, we offset each contact constraint by the relative velocity of its generating contact. Each velocity-level constraint will now define an affine half-space.

For body i, we denote the spatial velocity of the body with which it shares contact k by ${}^i\phi_k$. For convenience, we assume that this velocity has been transformed into body i's frame. Then the magnitude of the contact's velocity along the contact constraint normal is given by

$$d_k = {}^i\mathsf{n}_k{}^T \mathsf{M}\, {}^i\phi_k. \tag{25}$$

We then generate an affine velocity constraint on body i, by requiring that its velocity not only be in the feasible direction, but also have a sufficiently large component along the contact normal (see Figure 6):

$${}^i\mathsf{n}_k{}^T \mathsf{M}\, {}^i\phi - d_k \geq 0. \tag{26}$$

Each such constraint, computed from a contact k, defines a feasible affine half-space, $\mathsf{T}(\mathsf{n}_k, d_k)$. As before, the intersection of all such half-spaces generates $\mathsf{T}(\mathsf{q})$, the subspace of feasible velocities for contact set $C(\mathsf{q})$. While this subspace does not always form a cone, it will remain convex (this is important from an optimization perspective). We then compose a direct projection of ${}^i\phi^-$ onto $\mathsf{T}(\mathsf{q})$, that obtains a tangential velocity, ${}^i\phi^\tau$, along the boundary, $\partial\mathsf{T}(\mathsf{q})$. We can compute this projection as a separable, convex QP:

$$\begin{aligned} {}^i\phi^\tau &= \underset{\mathsf{y}}{\operatorname{argmin}} \,||\,\mathsf{y} - \phi^-||^2 \\ \text{subject to: } {}^i\mathsf{n}_k{}^T \mathsf{M}\ \mathsf{y} &\geq d_k,\ \forall k \in C(\mathsf{q}). \end{aligned} \tag{27}$$

Using a simple convexity argument (similar to the one mentioned in Section 4.1), we find that this projection implicitly defines an inelastic impulse, ${}^i\mathsf{r}$, that belongs to the normal cone, $\mathsf{N}(\mathsf{q})$ and so satisfies Equation 18.

For added efficiency we note that we can compute d_k locally in $\mathbb{R}^3$, at the point of contact $\boldsymbol{x}_k$. This is because

$$\begin{aligned} d_k &= {}^i\mathsf{n}_k{}^T \mathsf{M}\, {}^i\phi_k = (\mathsf{M}^{-1}\, {}_i\Gamma_k^T \boldsymbol{n}_k)^T \mathsf{M}\, {}^i\phi_k \\ &= \boldsymbol{n}_k^T\, {}^i\Gamma_k\, {}^i\phi_k = \boldsymbol{n}_k^T \boldsymbol{v}_k. \end{aligned} \tag{28}$$

Then, since $\boldsymbol{n}_k$ is still the $\mathbb{R}^3$ contact normal, and $\boldsymbol{v}_k$ is the relative velocity of the contact point, $\boldsymbol{n}_k^T \boldsymbol{v}_k$ can be computed in the contact frame, saving the expense of transformations into frame i.

6.2 Approximate Momentum Conservation

Consider the inelastic contact between bodies i and j from i's perspective. Because we treat j as imposing a moving constraint, if we pick ϕ_j^- as the velocity for a contact at $\boldsymbol{x}_k$, then i will behave as if j is a juggernaut with infinite inertia. Body i will be swept along with j's normal velocity without any regard for their respective masses. For objects of finite mass, in order to conserve momentum, it is better to consider the contact velocity at the instant when the two bodies are at rest relative to each other. This corresponds to the time of maximum compression in the Poisson model of restitution. We do not need to actually determine this instant in time. Simply knowing that this instant exists in our step interval is enough.

Momentum conservation allows us to obtain the velocity of a contact point along the contact normal, at the time of maximum compression, independent of the time of occurrence. First, we compute the effective mass of a body i with inertia M_i, as seen by a force acting along a contact normal $\boldsymbol{n}_k$, at a contact point $\boldsymbol{x}_k$. This is:

$$m_{k_i} = (\boldsymbol{n}_k^T\, {}^i\Gamma_k\ \mathsf{M}_i^{-1}\, {}_i\Gamma_k{}^T \boldsymbol{n}_k)^{-1}. \tag{29}$$

Now, if we let v_k be the velocity along the contact normal $\boldsymbol{n}_k$, we have $m_k v_k$ as the corresponding momentum. Then, at the time of maximum compression between bodies i and j, momentum conservation implies that there exists a velocity along the normal, $v_{k_{mc}}$, such that $(m_{k_i} + m_{k_j}) v_{k_{mc}} = m_{k_i} v_{k_i} + m_{k_j} v_{k_j}$. We compute this velocity by:

$$v_{k_{mc}} = \frac{(m_{k_i} v_{k_i} + m_{k_j} v_{k_j})}{(m_{k_i} + m_{k_j})}. \tag{30}$$

Here, v_{k_i} and v_{k_j} are body i and j's velocities at the point of contact projected along $\boldsymbol{n}_k$. This new velocity, $v_{k_{mc}}$, is the average of these two velocities weighted by their respective masses in the contact frame. Because $v_{k_{mc}}$ is our contact point's velocity at the instant of maximum compression along $\boldsymbol{n}_k$, we find that $d_k = {}^i\mathsf{n}_k^T\, \mathsf{M}\, {}^i\phi_{k_{mc}} = v_{k_{mc}}$, and so use $v_{k_{mc}}$ to generate our affine contact constraint.

6.3 Restitution and Friction

Using the above affine QP (Equation 27), we directly obtain the tangential velocity ${}^i\phi^\tau$. Although this step is implicit, we may still retrieve the effective imparted impulse, ${}^i\mathsf{r}$, from the difference between ${}^i\phi^\tau$ and ${}^i\phi^-$. Because ${}^i\mathsf{r}$ is inelastic and computed at the moment of maximum compression, we may treat it as the normal component of the impulse computed over a compression phase. Then, by again applying Poisson's model, we may add a restitution phase contribution of $\varepsilon\, {}^i\mathsf{r}$ to our rigid body. We let $\varepsilon \in [0,1]$, with $\varepsilon = 0$ corresponding to fully inelastic behavior.

Similarly, because ${}^i\mathsf{r}$ combines the appropriately weighted contributions of all contacts, our friction model can also be integrated into the final contact resolution. Friction is handled using a second projection (Equation 24) just as in the single body case.

7 Surrogate Constraint Method

Convex QPs with linear constraints (like our own) have a worst-case complexity polynomial in both dimension and the number of constraints applied [Kozlov et al. 1979]. If we additionally note that the dimension of our problem is fixed and that our QP is separable, we obtain a bound linear in the number of constraints (in our case contacts) [Megiddo and Tamir 1993]. However, the coefficient in the linear bound is still too high. We approximately solve our QP using the Surrogate Constraint Method (SCM) [Yang and Murty 1992] for constraint satisfaction. SCM moves towards our feasible tangent space by a projection onto a surrogate hyperplane that approximates our set of affine constraints. It produces a conservative step that is guaranteed to converge to a feasible solution.

8 Algorithm

We adopt a staggered Verlet [Verlet 1967] scheme for integration. During non-contacting periods this gives us second order, symplectic behavior for the same cost as a traditional explicit Euler step. During periods of contact, the same half-steps dovetail neatly with our pre- and post-resolution velocity steps. Taken together, this leads to a simple integration scheme in which we take second order explicit steps during free motion and semi-implicit non-smooth steps during contact.

8.1 Pseudocode

Let $\mathscr{B}$ denote the indexed set of rigid bodies. Performing a collision check on $\mathscr{B}$ produces, for all $i \in \mathscr{B}$, a set of detected contacts C_i. For each $k \in C_i$ we have an index $j \in \mathscr{B}$ of the other body involved in contact k, and the coordinates, $\boldsymbol{x}_k$, of this contact. At time t we start a step, with our configurations, q_i^t, and the post-resolution twists obtained in the middle of the previous step, $\boldsymbol{\phi}_i^+$. If $t = 0$, we use our initial velocities, $\boldsymbol{\phi}_i^0$, instead. We then proceed, following the steps in Figure 7.

8.2 Algorithm Details

We step through this algorithm using a fixed step size h. For each body $i \in \mathscr{B}$ we pre-compute the principal body frame. This allows us to generate a fixed diagonal inertia matrix M (in the body frame), using Mirtich's algorithm [Mirtich 1996]. We also compute each body frame's configuration relative to the world frame, q_i, in matrix form as ${}_i^W\mathsf{E}$ (see Equation 32). Our velocity half-step is then

velocity half-step($\boldsymbol{\phi}_i, \mathsf{q}_i$)
$\quad {}^i\dot{\boldsymbol{\phi}}_i := \mathsf{M}^{-1}\ {}_i\mathsf{f} + \mathsf{M}^{-1}({}^i\boldsymbol{\phi}_\times)^T \mathsf{M}\ {}^i\boldsymbol{\phi}_i.$
$\quad {}^i\boldsymbol{\phi}_i^{new} := {}^i\boldsymbol{\phi}_i + \frac{1}{2}h\ {}^i\dot{\boldsymbol{\phi}}_i.$
$\quad$ **return** ${}^i\boldsymbol{\phi}_i^{new}$

Here the wrench ${}_i\mathsf{f}$ is the sum of all non-contacting forces, including gravity. For our position half-step we use the exponential map as an $se(3)$ explicit Euler,

position half-step($\boldsymbol{\phi}_i, \mathsf{q}_i$)
$\quad {}_i^W\mathsf{E}^{new} := {}_i^W\mathsf{E}\ \exp([\,{}^i\boldsymbol{\phi}_i]\,\frac{1}{2}h).$
$\quad$ **return** ${}_i^W\mathsf{E}^{new}$

As each contact constraint is computed, we can immediately discard those which are strictly satisfied (see line 18 in Figure 7). Because these constraints are generated early in our pipeline, this is an efficient way to reduce the cost of each step. We proceed for each contact $k \in C$ by first computing our generalized normals, ${}^i\mathsf{n}_k$ (line 12), and our affine distances, d_k (lines 13–17). All ${}^i\mathsf{n}_k$ for which ${}^i\mathsf{n}_k^T M\,{}^i\boldsymbol{\phi}^- - d_k > 0$ are thrown away (since we will not need them to compute S, ${}^i\mathsf{r}$, and ${}^i\delta$), while all other k are saved in a new active contact set, A (line 19), for further computations.

These details reflect the built-in velocity-level complementarity of our approach. Contact constraints that a body is moving away from are satisfied, and so will not contribute normal and tangential impulses, while all other constraints are potentially active, and so will contribute impulses proportional to the degree of their constraint violation.

If $A \neq \emptyset$ then for each $k \in A$, we use each ${}^i\boldsymbol{n}_k$ to uniformly sample a finite set of the tangent plane at contact k (lines 21–25). These samples are used to approximate the set of all possible tangential impulses at k. The projections in lines 28 (Equation 27) and 33 (Equation 24) both use the SCM algorithm [Yang and Murty 1992] to approximate their respective QPs.

```
1.  for i ∈ B
2.      q_i^- := position half-step(φ_i^+, q_i^t)
3.      φ_i^- := velocity half-step(φ_i^+, q_i^t)
4.  collision check(B)
5.  for i ∈ B
6.      C := get contacts for body i
7.      A := ∅
8.      T := se(3)
9.      S := 0
10.     for k ∈ C
11.         j := index of other body in contact k
12.         n_k := M_i^{-1} {}_iΓ_k^T {}^in_k
13.         m_i := ( {}^in_k^T {}^iΓ_k M_i^{-1} {}_iΓ_k^T {}^in_k )^{-1}.
14.         v_i := {}^in_k^T [ {}^iφ_i^- ] {}^ix_k
15.         m_j := ( {}^jn_k^T {}^jΓ_k M_j^{-1} {}_jΓ_k^T {}^jn_k )^{-1}.
16.         v_j := {}^jn_k^T [ {}^jφ_j^- ] {}^jx_k
17.         d_k := (m_i + m_j)^{-1}(m_i v_i + m_j v_j)
18.         if (n_k^T M φ_i^-) − d_k ≤ 0
19.             A := A ∪ {k}
20.             T := T ∩ T(n_k, d_k)
21.             for m := 1 to sample size
22.                 s_{k_m} := m^th sample ⊥ to {}^in_k
23.                 μ_{k_m} := get μ for s_{k_m}
24.                 s_{k_m} := M_i^{-1} {}_iΓ_k^T s_{k_m}
25.                 S := S ⊕ s_{k_m}
26.     if A ≠ ∅
27.         φ_i^τ := proj_T(φ_i^-)
28.         r_i := φ_i^τ − φ_i^-
29.         for k ∈ A
30.             S := S ∩ T(n_k, 0)
31.             for m := 1 to sample size
32.                 S := S ∩ S(s_{k_m}, μ_{k_m}, n_k, r_i)
33.         δ_i := proj_S(−φ_i^τ)
34.         φ_i^+ := δ_i + εr_i + φ_i^τ
35.     else
36.         φ_i^+ := velocity half-step(φ_i^-, q_i^-)
37.     q_i^{t+1} := position half-step(φ_i^+, q_i^-)
```

Figure 7: Pseudocode for our algorithm.

9 Results

In this section, our goal is to analyze the performance of our implementation in a manner that will facilitate the comparison of our algorithm with existing and future methods. We first present a breakdown of the computational complexity of a single step in our iteration, with respect to the pertinent variables. We then present detailed performance figures for a sample simulation. Finally, we discuss the factors that affect the runtime performance of our simulation.

For the purposes of complexity analysis, each step of our implementation can be broken down into four phases: the preliminary half-step, contact detection, contact resolution, and the final half-step. In both the preliminary and the final half-step phases, each body is visited once, and a fixed amount of work is performed, so the complexity for both these phases is $O(n)$, where n is the number of bodies being treated by the simulator.

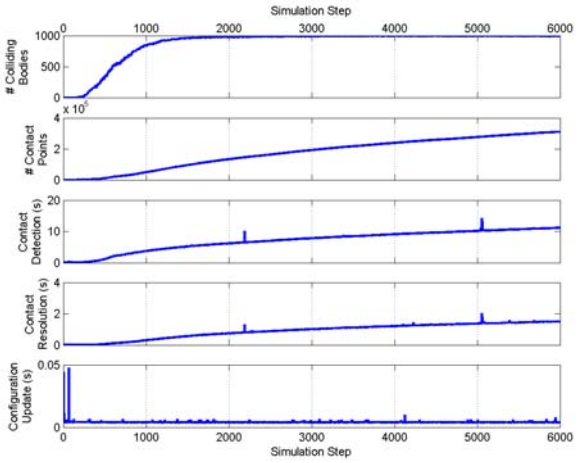


Figure 8: Performance results for a simulated drop of 1000 chess pieces onto a board (see Figure 1(a,b) and our accompanying video) showing the number of colliding bodies, number of detected contacts, and wall-time taken for each phase of each step of the simulation. When rendered into an animation the simulation is played at 210 simulation steps per second.

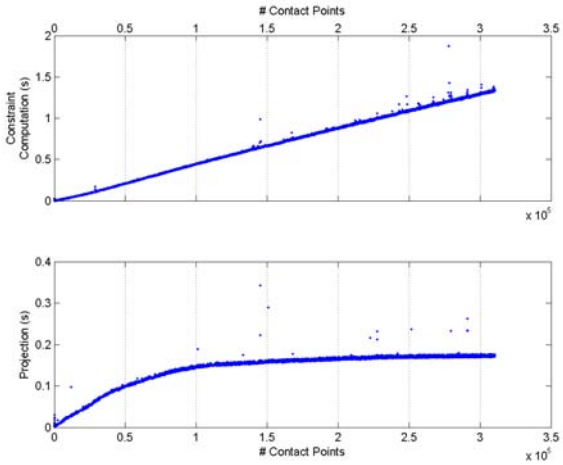


Figure 9: Performance results for the same simulation as in Figure 8 showing the wall-time spent in constraint computation and projection, as a function of the total number of contacts in a step.

The contact detection phase is responsible for finding all the points of contact occurring between any pairs of bodies in the simulation. The problem of efficient collision detection for rigid bodies is widely studied (see [Lin and Gottschalk 1998]), but is not a focus of this work. For our contact detection, we implemented a basic broad-phase collision detector and made use of an off-the-shelf mesh-mesh contact detector, PQP [Gottschalk et al. 1996], for pairwise contact detection. The complexity of our broad-phase collision detector depends on the number of bodies that are colliding or nearly colliding; it is $O(n)$ best case, and $O(n^2)$ worst case (in the physically implausible event that all n bodies are sufficiently close to each other). The complexities given above include calling the narrow-phase detector for each pair of possibly colliding bodies. For the purposes of this analysis, we will treat PQP as a constant-time operation in quantities such as number of triangles, but note that it is output-bound to be $\Theta(m_c)$, where m_c is the number of contact points detected between a given pair of bodies. Thus, the complexity of our contact detection as a whole is $O(n^2+m)$, where m is the total number of contacts found on all the bodies. Here it should be noted both that this is a very conservative bound, and that a more efficient collision detector could be seamlessly swapped into the implementation.

Each of the contacts reported by the contact detection phase must then be processed to resolve the new body velocities. As described in Section 8, each contact on a body is either determined to be inactive, and discarded, or marked as active and used to generate contact constraints. For each body that receives at least one active contact, two projections are performed; the cost of each of these projection computations is linear in the number of active contacts. The complexity of the contact resolution phase is thus $O(n_a+m)$, where n_a is the number of bodies with active contacts (in the worst case, $n_a = n$).

The key result of the above analysis is that excepting the collision detection phase (where optimization and careful analysis is likely to reduce the average-case complexity), our implementation is linear in the number of bodies being simulated, and linear in the total number of contact points detected at each step. Below, we will provide an indication of the relative coefficients of the complexities described here, by presenting the performance results of an example simulation.

We inserted wall-time recording code into our Java™ implementation, and ran a simulation of a $10 \times 10 \times 10$ grid of chess pieces free-falling onto a chess board (see Figure 1(a,b) and our accompanying video). Four different chess pieces were modeled with meshes sizes ranging from 1916 to 5814 triangles. The grid contained approximately 250 of each type. We ran the simulation using Sun's JDK 1.5 on a 3.6 GHz Pentium 4 machine with 1 GB RAM running Windows XP. In Figure 8, we show the progression over the course of the simulation of the number of colliding bodies and detected contacts, as well as the time taken by each of the algorithm phases. With the exception of outliers (caused by the use of wall-time as the performance measure), both half-step phases display the expected constant-time performance (since the number of simulated bodies is constant). Of note here is the continued increase in the number of contacts after the number of colliding bodies has saturated (during the time when settling objects are packing together). This underlines the importance of analyzing the cost per contact.

In Figure 9, we show the contact resolution time as a function of the total number of detected contacts. Here, we consider separately the two parts of the contact resolution phase: the constraint computation section, wherein each contact is either discarded as inactive or deemed active and then used to generate constraints, and the projection part, wherein two convex projections are performed for each active body. Note that while the former is linear in the total number of contacts, the latter evinces a saturation curve due to the correlation between the total number of contacts and the number of active bodies.

We have shown that the cost-per-step of our contact resolution is linear in both the number of simulated bodies and the total number of contact points. However, to meaningfully compare our performance with other methods, we must consider the computation of the contact points, and the simulation step size. Up to this point we have not mentioned number-of-triangles in our cost analysis, but our implementation's cost depends to a certain extent on the size of our meshes. Our choice of PQP as our final collision detection tool dictates that our contacts be at points of triangle-triangle intersection between colliding meshes. Therefore, the degree of mesh tessellation affects the number of contact points reported, and can also affect the quality (i.e. accuracy) of the contacts (and thus the quality of the simulation). This qualitative difference restricts our ability to analyze our implementation in the abstract, and makes us reliant on reporting the number of contacts processed in a given simulation. Similarly, the quality of the simulation improves as step size becomes sufficiently small, with "sufficient" depending on the speed of the bodies relative to their spatial frequency.

10 Summary

We have introduced a non-smooth, contact based approach to dynamics that results in a consistent theory and a robust linear time simulator. We have carefully formulated our approach in a geometric setting which has allowed us to handle multiple contacts with non-linear friction in a local manner. This, in turn, has allowed us to obtain large scale plausible simulations of non-convex geometries using an easily implemented rigid body simulator.

Acknowledgements: A special thanks to the anonymous reviewers and to Paul Kry, Qi Wei, and Shinjiro Sueda for their invaluable help. This work was supported in part by NSF grants IIS-0308157, ITR-0205671, and EIA-0215887.

References

ANITESCU, M., AND HART, G. D. 2004. A fixed-point iteration approach for multibody dynamics with contact and small friction. *Mathematical Programming 101*, 1, 3–32.

ANITESCU, M., AND POTRA, F. R. 1997. Formulating dynamic multirigid-body contact problems with friction as solvable linear complementarity problems. *ASME Nonlinear Dynamics 14*, 231–247.

BARAFF, D. 1989. Analytical methods for dynamic simulation of non-penetrating rigid bodies. In *Computer Graphics (SIGGRAPH 89)*, 223–232.

BARAFF, D. 1991. Coping with friction for non-penetrating rigid body simulation. In *Computer Graphics (SIGGRAPH 91)*, 31–41.

BARAFF, D. 1994. Fast contact force computation for nonpenetrating rigid bodies. In *Proceedings of ACM SIGGRAPH 94*, 23–34.

BROGLIATO, B. 1999. *Nonsmooth Mechanics*. Springer–Verlag.

CHATTERJEE, A., AND RUINA, A. L. 1998. A new algebraic rigid-body collision law based on impulse space considerations. *Journal of Applied Mechanics 65*, 4, 939–951.

CHATTERJEE, A., AND RUINA, A. L. 1998. Two interpretations of rigidity in rigid body collisions. *Journal of Applied Mechanics 65*, 4, 894–900.

CHATTERJEE, A. 1999. On the realism of complementarity conditions in rigid body collisions. *Nonlinear Dynamics 20*, 159–168.

DELASSUS, E. 1923. Sur les lois du froittement de glissement. *Bulletin de la Societe Mathematique de France 51*, 22–23.

ERDMANN, M. E. 1984. *On Motion Planning with Uncertainty*. Master's thesis, Massachusetts Institute of Technology.

GOTTSCHALK, S., LIN, M. C., AND MANOCHA, D. 1996. OBB-tree: A hierarchical structure for rapid interference detection. In *Proceedings of ACM SIGGRAPH 96*, 171–180.

GOYAL, S., RUINA, A., AND PAPADOPOULOS, J. 1991. Planar sliding with dry friction, Part 1. Limit surface and moment function. *Wear 143*, 307–330.

GUENDELMAN, E., BRIDSON, R., AND FEDKIW, R. 2003. Nonconvex rigid bodies with stacking. *ACM Transactions on Graphics (SIGGRAPH 03) 22*, 3, 871–878.

HAHN, J. K. 1988. Realistic animation of rigid bodies. In *Computer Graphics (SIGGRAPH 88)*, 299–308.

KOZLOV, M. K., TARASOV, S. P., AND HAČIJAN, L. G. 1979. Polynomial solvability of convex quadratic programming. *Soviet Mathematics Doklady 20*, 5, 1108–1111.

KRY, P. G., AND PAI, D. K. 2003. Continuous contact simulation for smooth surfaces. *ACM Transactions on Graphics 22*, 1, 106–129.

LIN, M. C., AND GOTTSCHALK, S. 1998. Collision detection between geometric models: a survey. In *IMA Conference on Mathematics of Surfaces*, 37–56.

LÖTSTEDT, P. 1984. Numerical simulation of time-dependent contact friction problems in rigid body mechanics. *SIAM Journal of Scientific Statistical Computing 5*, 2, 370–393.

MASON, M. T. 2001. *Mechanics of Robotic Manipulation*. MIT Press.

MEGIDDO, N., AND TAMIR, A. 1993. Linear time algorithms for some separable quadratic programming problems. *Operations Research Letters 13*, 203–211.

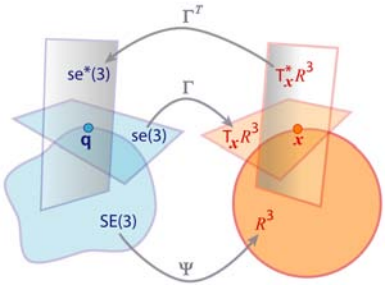


Figure 10: A Diagram of $SE(3)$ mappings.

MILENKOVIC, V. J., AND SCHMIDL, H. 2001. Optimization-based animation. In *Proceedings of ACM SIGGRAPH 01*, 37–46.

MIRTICH, B., AND CANNY, J. F. 1995. Impulse-based dynamic simulation of rigid bodies. In *Symposium on Interactive 3D Graphics*.

MIRTICH, B. 1996. Fast and accurate computation of polyhedral mass properties. *Journal of Graphics Tools 1*, 2, 31–50.

MOORE, M., AND WILHELMS, J. 1988. Collision detection and response for computer animation. In *Computer Graphics (SIGGRAPH 88)*, 289–298.

MOREAU, J. J. 1962. Décomposition orthogonale d'un espace hilbertien selon deux cônes muteullement polaires. *Comptes Rendus Academy Science Paris 255*, 238–240.

MOREAU, J. J. 1966. Quadratic programming in mechanics: One-sided constraints. *Journal SIAM Control 4*, 1, 153–158.

MOREAU, J. J. 1988. *Nonsmooth Mechanics and Applications*. No. 302 in CISM Courses and Lectures. Springer–Verlag, New York, ch. Unilateral Contact and Dry Friction in Finite Freedom Dynamics, 1–82.

MOREAU, J. J. 1994. Some numerical methods in multibody dynamics: application to granular materials. *European Journal of Mechanics A/Solids 13*, 4, 93–114.

MURRAY, R. M., LI, Z., AND SASTRY, S. S. 1994. *A Mathematical Introduction to Robotic Manipulation*. CRC Press.

PAULY, M., PAI, D. K., AND GUIBAS, L. 2004. Quasi-rigid objects in contact. In *ACM SIGGRAPH Symposium on Computer Animation*, ACM/Eurographics.

REDON, S., KHEDDAR, A., AND COQUILLART, S. 2002. Gauss' least constraints principle and rigid body simulations. In *IEEE International Conference on Robotics and Automation*.

ROCKAFELLAR, R. T. 1970. *Convex Analysis*. Princeton University Press.

SCHMIDL, H., AND MILENKOVIC, V. J. 2004. A fast impulsive contact suite for rigid body simulation. *IEEE Transactions on Visualization and Computer Graphics 10*, 2, 189–197.

SONG, P., AND KUMAR, V. 2003. Distributed compliant model for efficient dynamic simulation of systems with frictional contacts. In *The 2003 ASME Design Engineering Technical Conferences*.

STEWART, D., AND TRINKLE, J. C. 1996. An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction. *International Journal Numerical Methods Engineering 39*, 2673–2691.

TRINKLE, J. C., PANG, J. S., SUDARSKY, S., AND LO, G. 1995. On dynamic multi-rigid-body contact problems with coulomb friction. Tech. rep., Texas A+M University, Department of Computer Science.

VERLET, L. 1967. Computer experiments on classical fluids, thermodynamical properties of lennard-jones molecules. *Physical Review 159*, 98–103.

YANG, K., AND MURTY, K. G. 1992. New iterative methods for linear inequalities. *Journal of Optimization Theory and Applications 72*, 1, 163–185.

A Background

In this section we collect some notations used in this paper. We also describe some basic background concepts from kinematics and differential geometry. Murray et al. [1994] is a good resource for further information. We follow the font and index conventions described at the end of Section 2.

The configuration space of a rigid body, that is, the space of possible orientations and positions, is called $SE(3)$, the Special Euclidean group in three dimensions. Given a rigid body B with configuration $\mathsf{q} \in SE(3)$ and a point $\boldsymbol{x}_k \in \mathbb{R}^3$ belonging to B, there exists a mapping between B's configuration and point $\boldsymbol{x}_k$'s position in $\mathbb{R}^3$:

$$\Psi_k : \mathsf{q} \in SE(3) \rightarrow \boldsymbol{x}_k \in \mathbb{R}^3. \tag{31}$$

We represent the homogeneous coordinates of a three dimensional vector x in a frame i as ${}^i\boldsymbol{x}$. The homogeneous coordinates of this vector in frame j are obtained by left multiplying by the matrix:

$${}^j_i\mathsf{E} = \begin{pmatrix} {}^j_i\boldsymbol{\Theta} & {}^j_i\boldsymbol{p} \\ 0 & 1 \end{pmatrix} \tag{32}$$

such that

$${}^j\boldsymbol{x} = {}^j_i\mathsf{E}\,{}^i\boldsymbol{x}. \tag{33}$$

Here $\boldsymbol{\Theta}$ is a 3×3 rotation matrix and $\boldsymbol{p}$ is a 3×1 displacement.

Then, if we define q as the configuration of a body-fixed frame B with respect to a world frame W, and if the coordinates of the points $\boldsymbol{x}_k$ are designated in the body's frame as ${}^B\boldsymbol{x}_k$, we can represent ${}^W\mathsf{q}$ in a matrix form using ${}^W_B\mathsf{E}$. That is,

$${}^W\Psi_k(\mathsf{q}) = {}^W_B\mathsf{E}\,{}^B\boldsymbol{x}_k. \tag{34}$$

From the above coordinate transform we obtain the time derivative for the change of coordinates from frame i to frame j in the coordinates of frame j,

$${}^j\dot{\boldsymbol{x}} = {}^j_i\dot{\mathsf{E}}\,{}^i\boldsymbol{x}. \tag{35}$$

We define the bracket operator on $\mathbb{R}^3$, which constructs the skew-symmetric matrix:

$$[\boldsymbol{\omega}] \stackrel{\text{def}}{=} \begin{pmatrix} 0 & -\omega_2 & \omega_1 \\ \omega_2 & 0 & -\omega_0 \\ -\omega_1 & \omega_0 & 0 \end{pmatrix}. \tag{36}$$

Multiplying by this matrix gives us the cross product. Then,

$${}^i_j\mathsf{E}\,{}^j_i\dot{\mathsf{E}} = \begin{pmatrix} \Theta^T\dot{\Theta} & \Theta^T\dot{\boldsymbol{p}} \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} [\boldsymbol{\omega}] & \boldsymbol{v} \\ 0 & 0 \end{pmatrix} \tag{37}$$

gives us the change of frame i with respect to frame j in coordinates of frame i. This is an element of the tangent space to $SE(3)$, which is denoted $se(3)$. This quantity determines the spatial velocity vector, called a *twist*, defined in coordinates of frame i. We extract this twist using a linear operator $\mathfrak{U}$, so that

$${}^i\boldsymbol{\phi}(j,i) \stackrel{\text{def}}{=} (\boldsymbol{\omega}, \boldsymbol{v})^T = \mathfrak{U}({}^i_j\mathsf{E}\,{}^j_i\dot{\mathsf{E}}). \tag{38}$$

We denote $\mathfrak{U}$'s inverse by the bracket operator on twists so that, for the above spatial velocity we have

$$[\,{}^i\boldsymbol{\phi}(j,i)] = \begin{pmatrix} [\boldsymbol{\omega}] & \boldsymbol{v} \\ 0 & 0 \end{pmatrix}. \tag{39}$$

${}^i\boldsymbol{\phi}(j,i)$ describes the relative motion of frame i with respect to frame j, in coordinates of frame i. Here $\boldsymbol{\omega}$ is the angular velocity and $\boldsymbol{v}$ is the linear velocity of a point at the origin of frame i. When j is the world frame we will sometimes abbreviate ${}^i\boldsymbol{\phi}(j,i)$ using ${}^i\boldsymbol{\phi}_i$, or, when the context is clear ${}^i\boldsymbol{\phi}$ or $\boldsymbol{\phi}_i$. The 4×4 matrix in Equation 37 gives us the velocity of any point, ${}^i\boldsymbol{x}_k$, relative to frame j in coordinates of frame i:

$${}^i\dot{\boldsymbol{x}}_k = \begin{pmatrix} [\boldsymbol{\omega}] & \boldsymbol{v} \\ 0 & 0 \end{pmatrix} {}^i\boldsymbol{x}_k. \tag{40}$$

We dehomogenize these velocities by simply discarding the fourth component. Using Equation 40, we derive a frame specific differential mapping of spatial velocities (elements of $se(3)$) to the velocities of points in $\mathbb{R}^3$. These velocities, $\dot{\boldsymbol{x}}_k$, live in the tangent space at $\boldsymbol{x}_k$, denoted by $T_{\boldsymbol{x}_k}\mathbb{R}^3$. The mapping,

$${}^i\Gamma_k : {}^i\boldsymbol{\phi} \rightarrow {}^i\dot{\boldsymbol{x}}_k. \tag{41}$$

is performed by multiplying a twist by the 3×6 matrix

$${}^i\Gamma_k = \begin{pmatrix} -[\,{}^i\boldsymbol{x}_k] & I \end{pmatrix}. \tag{42}$$

A change in coordinate frames induces a change in spatial velocity coordinates. Here the adjoint matrix,

$${}^j_i\mathsf{Ad} = \begin{pmatrix} \Theta & 0 \\ [\boldsymbol{p}]\Theta & \Theta \end{pmatrix}, \tag{43}$$

is used. Since twists are contravariant quantities, they are transformed by left multiplying,

$${}^j\boldsymbol{\phi} = {}^j_i\mathsf{Ad}\,{}^i\boldsymbol{\phi}. \tag{44}$$

We equip $SE(3)$ with the kinetic metric so that

$$\mathsf{a} \cdot \mathsf{b} = \mathsf{a}^T\mathsf{M}(\mathsf{q})\,\mathsf{b} = \mathsf{a}^T \begin{pmatrix} \mathscr{I} & 0 \\ 0 & \mathscr{M} \end{pmatrix} \mathsf{b},\ \forall\, \mathsf{a}, \mathsf{b} \in se(3), \tag{45}$$

where $\mathsf{M}(\mathsf{q})$ is the frame appropriate inertial matrix at q. An explicit Euler step of length h can be defined for $se(3)$ using the exponential map. Given a configuration, ${}^j_i\mathsf{E}^0$, and a twist, ${}^i\boldsymbol{\phi}$, the explicit Euler step is:

$${}^j_i\mathsf{E}^h = {}^j_i\mathsf{E}^0 \exp([\,{}^i\boldsymbol{\phi}]h). \tag{46}$$

Spatial forces, called *wrenches*, are linear operators on tangent vectors in $se(3)$. They belong to the dual (*cotangent*) space $se^*(3)$. They are represented as $\mathsf{f} = (f_r, f_t)^T$ where f_r is the torque component and f_t is the translational force. Since they are covariant quantities, wrenches transform by left multiplying with the inverse transpose adjoint,

$${}_j\mathsf{f} = {}^i_j\mathsf{Ad}^T\,{}_i\mathsf{f}. \tag{47}$$

This is analogous to the way in which surface normals are transformed by the inverse transpose of a transformation matrix in the rendering pipeline. A force derived from a contact at $\boldsymbol{x}_k$ will lie within the cotangent force space denoted, $T^*_{\boldsymbol{x}_k}\mathbb{R}^3$. Because $T^*_{\boldsymbol{x}_k}\mathbb{R}^3$ is isomorphic to $\mathbb{R}^3$ (as is $T_{\boldsymbol{x}_k}\mathbb{R}^3$), we do not distinguish between them in this paper, in order to simplify notation. The transpose of ${}^i\Gamma_k$,

$${}_i\Gamma_k{}^T = \begin{pmatrix} [\,{}^i\boldsymbol{x}_k] \\ I \end{pmatrix}, \tag{48}$$

maps a force $f \in T^*_{\boldsymbol{x}_k}\mathbb{R}^3$ into $se^*(3)$ by left multiplying:

$${}_i\mathsf{f} = {}_i\Gamma_k{}^T f. \tag{49}$$

The Newton-Euler equation for a rigid body B in its own frame is

$${}_B\mathsf{f} = \mathsf{M}\,{}^B\dot{\boldsymbol{\phi}} - ({}^B\boldsymbol{\phi}_\times)^T\mathsf{M}\,{}^B\boldsymbol{\phi}, \tag{50}$$

with

$${}^B\boldsymbol{\phi}_\times \stackrel{\text{def}}{=} \begin{pmatrix} [\,{}^B\boldsymbol{\omega}] & 0 \\ [\,{}^B\boldsymbol{v}] & [\,{}^B\boldsymbol{\omega}] \end{pmatrix}. \tag{51}$$

Computing all dynamics in body coordinates allows us to use a fixed inertial tensor M, which can be computed in a principal axis aligned, diagonalized form.