Coping with Friction for Non-penetrating Rigid Body Simulation

David Baraff

Program of Computer Graphics
Cornell University
Ithaca, NY 14853

Abstract

Algorithms and computational complexity measures for simulating the motion of contacting bodies with friction are presented. The bodies are restricted to be perfectly rigid bodies that contact at finitely many points. Contact forces between bodies must satisfy the Coulomb model of friction. A traditional principle of mechanics is that contact forces are impulsive if and only if non-impulsive contact forces are insufficient to maintain the non-penetration constraints between bodies. When friction is allowed, it is known that impulsive contact forces can be necessary even in the absence of collisions between bodies. This paper shows that computing contact forces according to this traditional principle is likely to require exponential time. An analysis of this result reveals that the principle for when impulses can occur is too restrictive, and a natural reformulation of the principle is proposed. Using the reformulated principle, an algorithm with expected polynomial time behavior for computing contact forces is presented.

Categories and Subject Descriptors: I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling; I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism

Additional Key Words and Phrases: dynamics, friction, simulation, *NP*-complete

1. Introduction

The synthesis of realistic motion is one of the goals of computer graphics. Recently, much attention has been given to physically based simulation methods, and in particular, rigid body simulation. To achieve realism, simulations must incorporate the effects of friction between contacting bodies. If the total number of contact points is small, for instance one to four, the effects of friction are easily computed. However, as the number of contact points grows, the problem becomes considerably more challenging. Simulation algorithms with exponential (in the number of contact points) running times are known[5] but are impractical for problems involving as few as 10 to 15 contact points. In order to make rigid body simulations with friction practical for computer graphics, efficient, polynomial time algorithms are needed.

This paper considers the problems of computing friction forces for configurations of perfectly rigid bodies with a finite number of contact points. For polyhedral bodies, only the vertices of the line segment and polygonal contact regions are considered as contact points. Unless otherwise stated, it is assumed that bodies are not colliding at any contact point. No restriction is placed on the allowable sliding motion between bodies at contact points. Forces at contact points are classified as either *normal* or *friction* forces. Normal forces prevent inter-penetration by acting perpendicularly to the contact surfaces. Friction forces act tangentially to the contact surfaces and oppose slipping motion. The friction force at a contact point is called *dynamic* friction if the two bodies are slipping at the contact point; otherwise, the friction force is called *static* friction. The contact forces (the normal and friction forces) must satisfy the Coulomb model of friction. The Coulomb model of friction is a well accepted empirical relationship between the normal and friction force at a contact point.

An important first step to coping with the problems of friction is understanding the simulation behavior specified by the Coulomb model of friction. We need to know both what kind of result the model specifies and the degree of difficulty in computing that result. When computing contact forces, a principle of rational mechanics called the *principle of constraints* [9] is usually accepted. The principle of constraints states that constraints should be satisfied by non-impulsive forces if possible; otherwise, impulsive forces should be used to satisfy constraints. (Impulsive forces, or impulses, have the units of mass times velocity and discontinuously change velocities; impulses most commonly arise when bodies collide. Non-impulsive forces, or just forces, have the units of mass times acceleration and cannot produce velocity discontinuities.) The first result of this paper is a proof that computing friction forces according to the principle of constraints is likely to require exponential time (section 5). Under the Coulomb friction model, even in the absence of collisions it is sometimes necessary to introduce impulses between contacting bodies to prevent inter-penetration. Adopting the principle of constraints requires that a particular behavior, non-impulsive contact forces, be searched for among possibly exponentially many other choices, whenever possible. In formal terms, we will prove that deciding if non-impulsive contact forces are sufficient to prevent inter-penetration is *NP*-complete. Essentially, this means that an efficient (that is, polynomial time) algorithm for computing contact forces is widely believed not to exist. (See Garey and Johnson[7] for a discussion on *P*, *NP*, *NP*-complete and *NP*-hard problems).

However, the preference for non-impulsive behavior is neither necessary nor justified. Using insights from the *NP*-completeness results of section 5, section 6 presents a physical model for contact that argues against the principle of constraints. We will use this model to reformulate the problem of computing contact forces. Using the reformulated problem, we present an efficient algorithmic simulation method for dealing with dynamic friction. The algorithm has an expected running time that is polynomial in the number of contact points of the configuration. This is the first efficient algorithm we know of for computing dynamic

©1991 ACM-0-89791-436-8/91/007/0031 $00.75

friction forces.

As a first step towards dealing with both static and dynamic friction, we present two preliminary approaches for computing static and dynamic friction forces (section 8). The first approach approximates both static and dynamic friction by using the general algorithm for dynamic friction. The second approach uses an iterative technique to compute static and dynamic friction forces; however, convergence is not guaranteed.

2. Definitions

For configurations without friction, a *valid* set of contact forces is a set of normal forces satisfying three conditions. First, the normal force at each contact point must be oriented to "push" the bodies apart. Second, the normal forces must be sufficient to prevent inter-penetration between bodies. Third, if two bodies are separating at a contact point, the normal force at the contact point must be zero. For configurations without friction, a valid set of contact forces exists for any configuration of bodies. Although a valid set of contact forces is not necessarily unique for frictionless configurations, all valid contact forces yield the same accelerations of the bodies in the configuration[4]. Contact forces for frictionless configurations with n contact points can be found by formulating and solving a convex quadratic program (QP) of n variables. Methods for formulating this QP for bodies composed of polyhedra and curved surfaces have been presented in [1,2,6,11]. Convex QP's with n variables can be solved in time polynomial to n and in practice are solved by algorithms whose worst case behavior is exponential but whose expected running time is polynomial[14].

Configurations with friction are more complicated. Contact forces with friction are valid if they satisfy both the previous three conditions for normal forces and the Coulomb friction model (sections 4 and 8). Valid contact forces for configurations with just dynamic friction (and no static friction) can be found, as in the frictionless case, by computing the solution to a QP. Unlike the frictionless case though, the QP associated with a configuration involving dynamic friction is not necessarily convex. The existence of a practical solution method for non-convex QP's is considered unlikely, because solving non-convex QP's is *NP*-hard.

Additionally, it is possible that the QP for a configuration with dynamic friction may not even have a solution. Although the Coulomb friction model is well accepted, it has been known for at least a century that configurations of rigid bodies with dynamic friction exist that have no valid set of contact forces. We call such a configuration *inconsistent*. Conversely, there are also configurations with dynamic friction where neither the set of valid contact forces nor the accelerations resulting from those contact forces are unique. Such a configuration is called *indeterminate*. (See sections 4.1 and 4.2).

3. Previous Work

Wang and Mason[16] present a detailed discussion on single contact point collisions involving friction; in particular, methods for computing the contact impulse resulting from the collision are described. (We will not consider the general problem of collisions involving friction in this paper). Mason and Wang[13] discuss inconsistent configurations and explain how to resolve the inconsistency by applying impulsive contact forces to the configuration. However, it is first necessary to identify configurations as inconsistent. As we will show, this turns out to be a difficult problem.

A paper by Lötstedt[11] discusses a simulation method that avoids inconsistency by modification of the friction law. Lötstedt's method changes the Coulomb model into a relation between normal forces from the *previous* time step and friction forces from the current time step. Lötstedt's method approximates both dynamic and static friction by solving a convex QP. It is not clear that Lötstedt's method can always be initialized so that it is numerically stable. It is also unclear how to perform such an initialization efficiently.

4. Contact Force Model

We begin by considering configurations with only dynamic friction. Static friction is not considered until section 8. This section introduces a special-case of a single contact point configuration (figure 1). This configuration, and minor variations of it, will be used several times throughout this paper.

In figure 1, body A is a thin rod of length two with a symmetric mass distribution that contacts body B at a single contact point. Body B (the "base") is fixed.

Variables

p_a	contact point	$\dot{p}_a$	contact point velocity
$\hat{n}$	unit surface normal	$\hat{t}$	unit surface tangent
m	A's mass	I	A's moment of inertia
$\vec{\omega}$	A's angular velocity	μ	coefficient of friction
g	gravitational acceleration		

Relations

$I = \frac{m}{16}$ $\theta = 72°$ $16(\cos^2\theta - \mu\cos\theta\sin\theta) = -2$

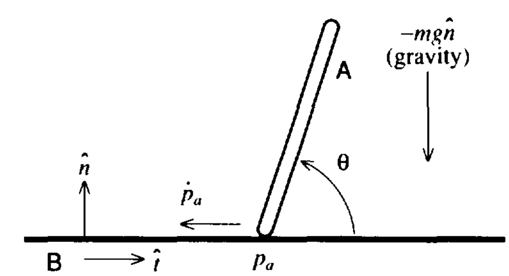

Figure 1. A one contact point configuration with dynamic friction between a thin rod *A* and a fixed base *B*.

By choosing A's angular velocity $\vec{\omega}$ and the magnitude g of the gravity force $-mg\hat{n}$ acting on A, an indeterminate and an inconsistent configuration can be produced. This particular example can be found in a number of papers; for example, Lötstedt[10], Erdmann[5], or Mason and Wang[13].

For a given value of $\vec{\omega}$, the linear velocity of A is chosen such that the point p_a on A has a non-zero velocity tangent to B, and zero velocity normal to B. The unit vector $\hat{n}$ is normal to the surface of B. The unit vector $\hat{t}$ is tangent to the surface of B, and is directed opposite to the motion of the point p_a; $\hat{n}$ and $\hat{t}$ are perpendicular. The particular values of I, θ and μ ($\mu \approx 3/4$) given in figure 1 are somewhat arbitrary; these values are chosen to simplify later computations.

The Coulomb model of friction states that since p_a is sliding across B, a friction force in the direction $\hat{t}$ acts on A. (An equal and opposite friction force acts on B, but B is fixed.) If the normal force acting on A has magnitude f, then the Coulomb friction model states that the friction force has a magnitude of μf (figure 2). The net contact force acting on A is

$$f\hat{n} + \mu f\hat{t} = f(\hat{n} + \mu\hat{t}). \qquad (1)$$

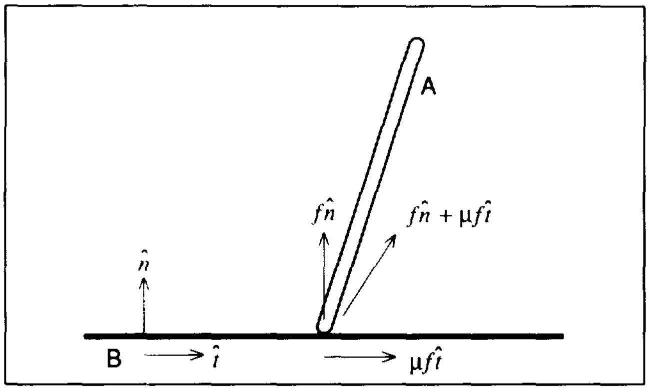

Figure 2. Normal and friction forces acting on *A*.

What effect do the contact and external force $-mg\hat{n}$ have on A? In appendix A, the component of acceleration of the point p_a, normal to B, is found to be

$$\hat{n} \cdot \ddot{p}_a = -\frac{f}{m} + (|\vec{\omega}|^2 \sin\theta - g). \tag{2}$$

This configuration has the odd property that as the normal force magnitude f is *increased*, the point p_a is accelerated more strongly *towards* B! Geometrically, the *direction* of the net contact force $f(\hat{n} + \mu\hat{t})$ does not change as f is increased. However, as f is increased, the torque due to friction causes p_a to angularly accelerate downward. The normal force $f\hat{n}$ also causes the center of mass of A, and thus p_a, to accelerate upwards, but not fast enough to overcome the downwards acceleration due to the torque. The net result is that increasing f decreases the value of $\hat{n} \cdot \ddot{p}_a$. (See appendix A for details).

4.1 An Indeterminate Configuration

To produce an indeterminate configuration, let $\vec{\omega}$ and g satisfy $|\vec{\omega}|^2 \sin\theta - g = 1$. Then equation (2) becomes

$$\hat{n} \cdot \ddot{p}_a = -\frac{f}{m} + 1. \tag{3}$$

Recall that valid contact forces satisfy three conditions. The first condition, that the normal force "push" bodies apart is simply $f \geq 0$. The second condition, that contact forces prevent inter-penetration, requires the acceleration of p_a in the $\hat{n}$ direction to be non-negative. This yields the constraint $\hat{n} \cdot \ddot{p}_a \geq 0$. The last condition is that if the bodies are separating, the normal force must be zero. Since the bodies are separating if and only if $\hat{n} \cdot \ddot{p}_a$ is strictly positive, this condition may be written as $f\hat{n} \cdot \ddot{p}_a = 0$. For $|\vec{\omega}|^2 \sin\theta - g = 1$ and using equation (2), the above three conditions are

$$f \geq 0, \quad -\frac{f}{m} + 1 \geq 0 \quad \text{and} \quad f(-\frac{f}{m} + 1) = 0. \tag{4}$$

The valid contact forces are given by the solution of equation (4); $f = 0$ and $f = m$.

For the $f = 0$ solution, $\hat{n} \cdot \ddot{p}_a = 1$. In this solution, the centripetal acceleration of $\ddot{p}_a$ is stronger than the force of gravity pulling A down; thus, A merely continues its rotation and the point p_a moves off of B (figure 3a).

In the second solution, $f = m$ and $\hat{n} \cdot \ddot{p}_a = 0$. A normal force of $m\hat{n}$ and a friction force of $\mu m\hat{t}$ act on A. The torque generated by friction balances the centripetal acceleration of p_a; as a result, A and B do not break contact (figure 3b). Note that the *only* valid values of f are $f = 0$ or $f = m$. Since the solutions produce different accelerations for A, the configuration is indeterminate.

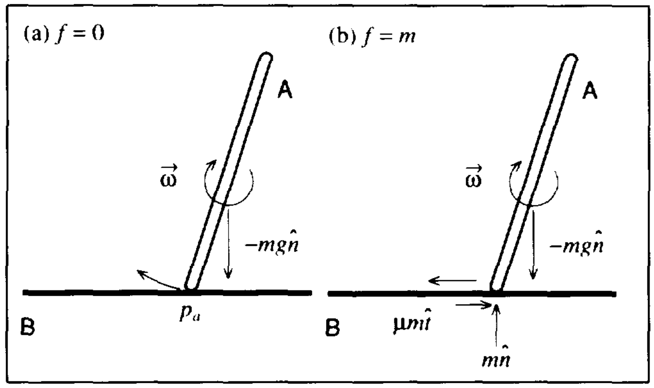

Figure 3. (a) The contact force between *A* and *B* is zero. p_a rotates to the left and up, breaking contact with *B*. (b) The normal and friction forces balance gravity and centripetal acceleration; p_a moves horizontally and maintains contact with *B*.

4.2 An Inconsistent Configuration

Now suppose that $|\vec{\omega}| = 0$ and A's linear velocity $\vec{v}$ is opposite $\hat{t}$ (figure 4). Then the condition $\hat{n} \cdot \ddot{p}_a \geq 0$ is

$$\hat{n} \cdot \ddot{p}_a = -\frac{f}{m} - g \geq 0. \tag{5}$$

However, if $g > 0$ (figure 4a), then no positive value of f can prevent p_a from accelerating downwards and thus inter-penetrating; that is, equation (5) cannot be satisfied by any $f > 0$. This means that the configuration is inconsistent. The existence of such a configuration may seem counter-intuitive; however, we will have more to say on this phenomenon in section 6.1.

Note that the value of g is crucial. If $g = 0$, so that no external force acts on A, then $f = 0$ becomes the (unique) valid contact force (figure 4b). Any positive value of f for this configuration causes inter-penetration. Figure 4b corresponds to p_a "skimming" horizontally over B, with neither a normal force nor a friction force exerted on A. If g becomes even slightly positive however, the configuration is inconsistent. Note that the requirement that B be fixed is not crucial. If B is massive compared to A, then inconsistency occurs if an external force acts on A to accelerate it towards B, or vice versa.

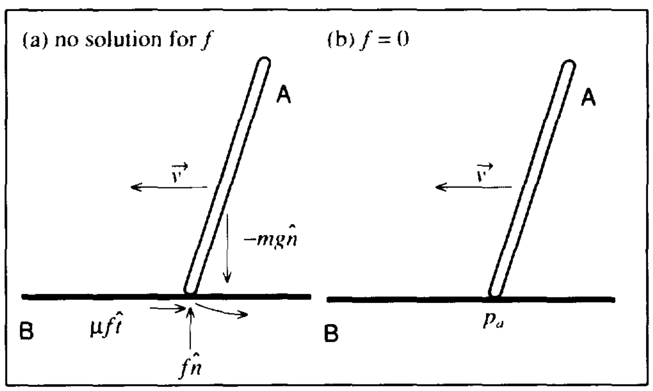

Figure 4. (a) An inconsistent configuration. For any $f \geq 0$, p_a is accelerated downwards into *B*. (b) The configuration has a unique solution of $f = 0$ when gravity is removed; *A* skims along the surface of *B*.

5. An *NP*-complete Class of Configurations

We define the *frictional consistency* problem as the problem of deciding if a given configuration is consistent. In this section, we prove that the frictional consistency problem is *NP*-complete. We begin by showing that the frictional consistency problem lies in *NP* and then show that the frictional consistency problem is *NP*-hard. Although the configurations constructed in this section seem contrived (and arguably are), the *NP*-hardness result has grave implications even when inconsistency is not a concern during simulation.

Definition. *An instance of the frictional consistency problem is a configuration* C *of bodies that contact at* n *distinct contact points. The physical properties of each body (mass, moment of inertia, linear and angular velocity, position and orientation, and external forces) are described by rational numbers. The specifics of a contact point (position, coefficient of friction, surface normal) are also described by rational numbers. The relative motion between bodies at contact points with friction is non-zero in the direction tangent to the contact surface and zero in the direction normal to the contact surface. The notation* $|C| = k$ *means that configuration* C *is describable in* k *bits. Clearly* $k > n$.

Theorem 1. *The frictional consistency problem lies in NP.*

Proof. Given an instance of C, a QP of size n with the following two properties exists. (1) If C is consistent, then an n-vector $\vec{x}$ that is a solution to the QP exists. The set of contact forces such that the magnitude of the normal force at the ith contact point is x_i is a valid set of contact forces for the configuration C. (2) Otherwise, if C is inconsistent, the QP has no solution. The specifics of constructing the QP can be found in [6]. The numerical quantities in the QP are computed from the rational entries of C in a total of $O(n^3)$ arithmetical operations. The QP can therefore be constructed in time polynomial to k. Vavasis[15] has recently shown that quadratic programming lies in *NP*. It follows from this that deciding frictional consistency is also in *NP*. ■

In order to show that deciding frictional consistency is *NP*-hard, we reduce the *NP*-complete problem "subset sum" to the frictional consistency problem.

Definition. *An instance of the subset sum problem is a pair* (A,S) *where* $A = \{a_1, \cdots, a_n\}$ *is a set of positive integers and* S *is a single positive integer. A subset sum instance* (A,S) *is* satisfiable *if there exists a subset* $A' \subset A$ *such that*

$$\sum_{a \in A'} a = S. \tag{6}$$

Deciding if an instance of the subset sum problem is satisfiable is an NP-complete problem[7].

To show that deciding frictional consistency is *NP*-hard we take an arbitrary instance (A,S) of the subset sum problem and construct (in polynomial time) a configuration of bodies C. The configuration C will have the property that C is consistent if and only if (A,S) is satisfiable.

Theorem 2. *Deciding frictional consistency is NP-hard.*

Proof. Consider the configuration of figure 5. Body B of figure 5 is initially at rest and is positioned by four fixed triangular wedges that contact B without friction. Body B is therefore free to move horizontally, but can neither rotate nor move vertically. On either side of body B are thin rods E_1 and E_2. E_1 and E_2 have no angular velocity and have a linear velocity as indicated. E_1 and E_2 contact B in the same manner as the configuration of figure 4 (although the frames of reference for E_1 and E_2 are rotated by 90° with respect to figure 4). In figure 4, inconsistency occurred if external forces accelerated A towards B or vice versa. The same holds true for figure 5. If B has an acceleration leftwards (towards E_1), then inconsistency occurs. Likewise, if B has an acceleration rightwards (towards E_2), then inconsistency also occurs. Thus, the configuration of figure 5 is consistent only if the net horizontal acceleration of B is zero. In this case, the rods E_1 and E_2 skim along the surface of B as in figure 4b.

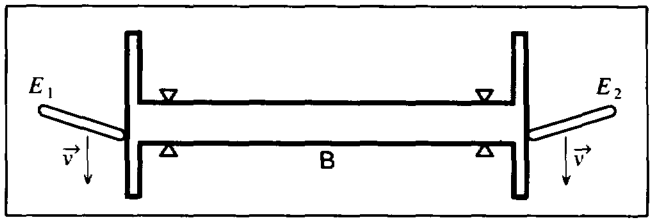

Figure 5. *B* is constrained by the fixed wedges and can only move horizontally. However, the configuration is consistent only if *B* is not subject to a net horizontal force.

Now consider figure 6, where a collection of thin rods $R_1, \cdots, R_n$ have been added. In addition, an external horizontal force with magnitude μS acts on B, trying to accelerate B to the right. Each rod R_i has mass m_i. The configuration between each rod R_i and B is the same as the configuration of figure 3; thus each rod R_i has angular velocity $\vec{\omega}$ and is subject to an external gravity force. Let f_i be the magnitude of the normal force between R_i and B. As in figure 3, the only valid solutions for f_i are $f_i = 0$ and $f_i = m_i$. If $f_i = 0$, then no friction force acts between R_i and B. Otherwise, $f_i = m_i$ and a friction force of magnitude μm_i acts between R_i and B. The friction force pushes R_i to the right and B to the left, with magnitude μm_i. The friction force on B therefore acts to oppose the external force of magnitude μS.

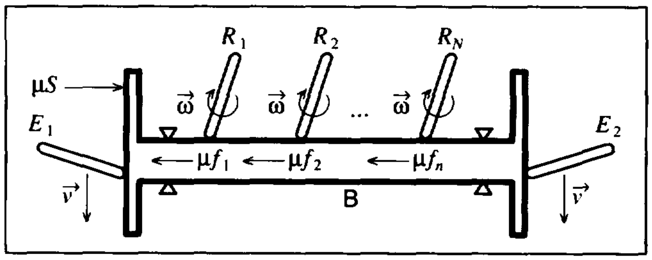

Figure 6. The configuration is consistent if and only if the friction forces on *B* sum to μS.

In order for the configuration of figure 6 to be consistent, B must have no net horizontal acceleration. This means that the friction forces exerted on B from the n rods must sum to μS, balancing the external force applied to B. Thus, the configuration is consistent if and only if

$$\sum_{i=1}^{n} \mu f_i = \mu S. \tag{7}$$

Since each f_i is either 0 or m_i, the configuration is consistent if and only if some subset of $\{m_1, \cdots, m_n\}$ sums to S.

We can now perform the reduction from subset sum to show *NP*-hardness. Given any set $A = \{a_1, \cdots, a_n\}$ and any target sum S, construct the configuration of figure 6. Assign $m_i = a_i$ for $1 \le i \le n$, and let an external horizontal force of μS act on B as shown in figure 6. By the above discussion, the configuration is consistent if and only if there exists a subset of $\{m_1, \cdots, m_n\}$ that sums to S. But since $A = \{m_1, \cdots, m_n\}$, the configuration is consistent if and only if (A,S) is satisfiable. We conclude that the problem of deciding frictional consistency is *NP*-hard. ■

Theorem 3. *Deciding frictional consistency is NP-complete.*

Proof. The result follows immediately from Theorem 1 and Theorem 2. ∎

Corollary 1. *Computing contact forces (if they exist) for a configuration is NP-hard.*

Proof. Since deciding if a set of contact forces exists is an NP-complete problem, computing the contact forces (if they exist) is an NP-hard problem. ∎

5.1 Implications

At this point, it may seem that the above results, while possibly of some (marginal) theoretical interest, have no bearing on any practical problem. Certainly, the above configurations were carefully constructed to produce configurations whose consistency was difficult to determine. But how likely is it that a configuration this carefully constructed could occur during simulation? For that matter, suppose the occurrence of *any* inconsistent configuration is so unlikely that the possibility can be completely disregarded. (This may be a reasonable assumption. We have not encountered an inconsistent configuration during simulation when $\mu < 1$.) Can a polynomial time algorithm that computes contact forces only for consistent configurations be constructed? The answer to this is no, unless it turns out that P and NP are equivalent, and it is widely believed that they are not.

Corollary 2. *A polynomial time algorithm for computing valid contact forces for consistent configurations exists if and only if $P = NP$.*

Proof. Suppose that $P = NP$. Since quadratic programming lies in NP, $P = NP$ implies a polynomial time algorithm for finding the solution to a QP. Since valid contact forces for a consistent configuration of bodies can be found by solving an associated QP, valid contact forces are computable in polynomial time if $P = NP$.

Conversely, suppose that contact forces for consistent configurations can be computed in polynomial time. Then there exists a machine M and a polynomial p with the following behavior. Whenever M is given a consistent configuration C as input, M outputs a valid set of contact forces within time $p(|C|)$. M's behavior when C is inconsistent is undefined. Given *any* configuration C, not necessarily consistent, M can be used to decide consistency in polynomial time as follows.

Let C be input to M and run for $p(|C|)$ time. If M fails to output within this time, then C is inconsistent. Otherwise, M has produced some output. Since deciding frictional consistency is in NP, the validity of M's output can be decided in an additional amount of time that is also a polynomial function of $|C|$. If M's output is a valid set of contact forces, then clearly C is consistent. If M's output is invalid, then C must be inconsistent (else M would have output a valid answer). In any event, the consistency of C has been decided in polynomial time.

Since deciding consistency is NP-complete, we conclude that the existence of a polynomial time algorithm for computing contact forces on consistent configurations would imply that $P = NP$. ∎

Given the above conclusions, it is unlikely that an efficient algorithm for computing contact forces can be found. This depressing result can be viewed in several ways. First, the simulation of rigid bodies with friction can be considered an intractable problem, unless the number of contact points with friction in a configuration is small. Second, the general simulation problem can be rejected as being too difficult a problem, although we might hope to find some natural class of configurations with friction for which contact forces can be computed efficiently. Such a class would have to be sufficiently general to cover situations likely to be encountered in practice. Third, heuristic methods for computing contact forces can be considered. However, this is essentially the same as hoping to find a natural class of configurations with easily computed contact forces. Rather than adopt any of these viewpoints, the next section presents a physical model of inconsistency that leads to a natural reformulation of the problem of computing contact forces.

6. Physical Models

In this section, a physical model for both inconsistency and indeterminacy is presented. Certainly, other models are possible, and a different choice of model might lead to different conclusions and results. The model in this section was developed in order to understand the behavior of inconsistent and indeterminate configurations. After the model was developed, we found that the model leads to a natural refutation of the principle of constraints. By abandoning this principle, the problem of computing contact forces is naturally reformulated and a correspondingly efficient way of computing contact forces is found. The model in this section is *not* an *ad hoc* attempt at dealing with friction. We feel that the model is not unreasonably based on the physical properties of rigid bodies, and sensible in the context of simulating rigid bodies with friction. The model and subsequent reformulation of the problem is presented in this section. In the next section, a computational algorithm is presented for solving the reformulated problem.

The motivation of a physical model stems from the need to answer the following basic question: what should be the result of a simulation when inconsistency is encountered? For inconsistent configurations, such as figure 4a, the only resolution is the introduction of an impulsive contact force at p_a[9, 13]. Impulses, however, arise from collisions between bodies. Given the fact that p_a has no velocity normal to B (so that A and B do not appear to be colliding), why should an impulse be applied between A and B? We answer this by presenting our physical model of inconsistency.

The physical model we present is based on questioning the rigid body assumption. In the physical world, there is of course no such thing as a perfectly rigid body. For near rigid bodies, contact forces arise as a result of small elastic deformations in the neighborhood of the contact area. Rather than geometrically model deformations, we shall (conceptually) allow bodies to inter-penetrate slightly, and consider a deformation in the contact surfaces proportional to the amount of inter-penetration. (We do not of course imagine that real bodies actually inter-penetrate). As the inter-penetration depth increases, a restoring normal force acts to oppose the inter-penetration. This is the so called "penalty method", a simulation method that models contact between bodies as spring and damper systems. The normal force between two bodies is zero when the amount of inter-penetration is zero, and increases monotonically as the inter-penetration increases. Typically, the normal force is modeled as a linear spring force $-Kd$, where K is the spring constant and d is the amount of inter-penetration. Although this is a very useful conceptual model, it is not well suited to simulation of very rigid bodies[1, 3]. We will use the penalty method to conceptually model inconsistency and indeterminacy, but we will *not* use the penalty method as a simulation technique.

6.1 A Model of Inconsistency

Figure 7 shows the behavior of the inconsistent configuration of 4a when the penalty method is applied. At time t_0, consider the tip of the rod, p_a, to be resting exactly on B, with zero inter-penetration. Since there is no inter-penetration, the normal force is zero. Even though p_a is sliding along B, the friction

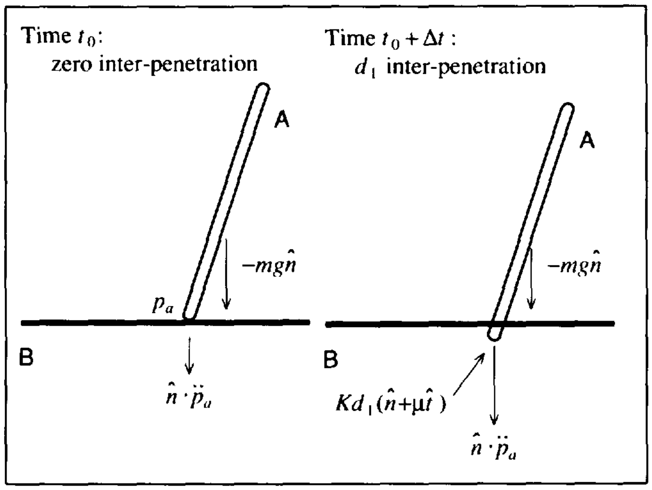

Figure 7. At time t_0, only gravity acts on A. At time $t_0 + \Delta t$, the inter-penetration distance is d_1 and both a penalty and a gravity force act on A, causing p_a's downwards acceleration to increase.

force is zero since the normal force is zero. Since the only force acting on A is the external gravity force $-mg\hat{n}$, p_a accelerates downwards. At time $t_0 + \Delta t$, p_a has inter-penetrated B by an amount d_1, so a normal force $Kd_1\hat{n}$ acts on A. Since p_a is still sliding, a friction force of $\mu K d_1 \hat{t}$ also acts on A. The net result, from equation (2), is that this causes p_a to accelerate downwards even faster than before. As the penalty force continues to increase, it causes more inter-penetration between A and B; a form of positive feedback. Accordingly, both the friction and the normal force increase, and the cycle continues. Since we are trying to model A and B as rigid bodies, the spring constant K must be allowed to be arbitrarily large. (It is this feature that makes the penalty method ill-suited to rigid body simulation). The larger K is, the faster inter-penetration increases and the faster the normal and friction forces build.

Recall that the friction force opposes the sliding motion of A across B. By making K arbitrarily large, the friction force brings p_a to rest (horizontally) in an arbitrarily short time. Now, suppose K is adjusted so that p_a comes to rest within time Δt. Then the amount of inter-penetration is $O(\Delta t^2)$, since the vertical distance traveled by p_a depends quadratically on the time for which it travels. In the limit as K goes to infinity, the contact force on A acts as an impulse and instantaneously brings p_a to rest horizontally, without inter-penetration occurring. This impulse also causes p_a to acquire a normal velocity towards B, bringing them into colliding contact. The (second) impulse resulting from this colliding contact can be computed according to [16].

Once p_a is at rest horizontally, dynamic friction is replaced by static friction. The Coulomb friction model states that the magnitude f_{static} of static friction satisfies $f_{static} \le \mu f$ whereas $f_{dynamic} = \mu f$ for dynamic friction. (Actually, μ is typically larger for static friction than dynamic friction, but this has no bearing on the model being developed.) Because static friction is less constrained than dynamic friction, once static friction occurs, a valid solution exists and the inconsistency is removed.

6.2 A Model of Indeterminacy

Consider the indeterminate configuration of figure 3, which has solutions $f = 0$ and $f = m$. Using the penalty method, the indeterminacy can be removed by assuming some amount of initial inter-penetration between A and B. If the initial inter-penetration between A and B is zero (figure 8) then no normal

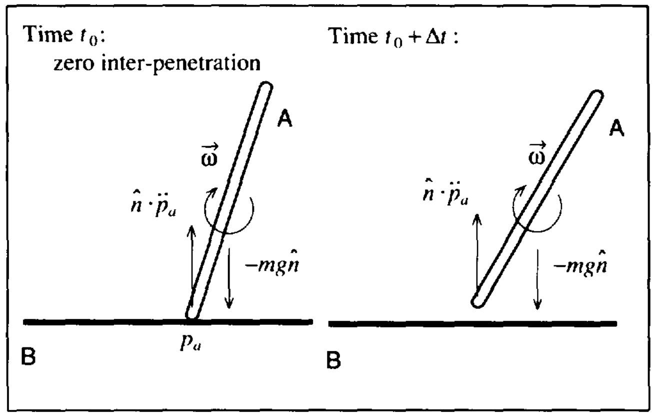

Figure 8. The initial inter-penetration is zero and only gravity acts on A. The centripetal acceleration of A pulls p_a away from B and contact is broken.

force exists, and contact is immediately broken (due to the centripetal acceleration of p_a away from B). The behavior is the same as in figure 3a. However, if the initial inter-penetration produces a normal force magnitude of m, then the normal and friction forces prevent A from breaking contact with B. In figure 9, let the initial inter-penetration d_1 be $\frac{m}{K}$.

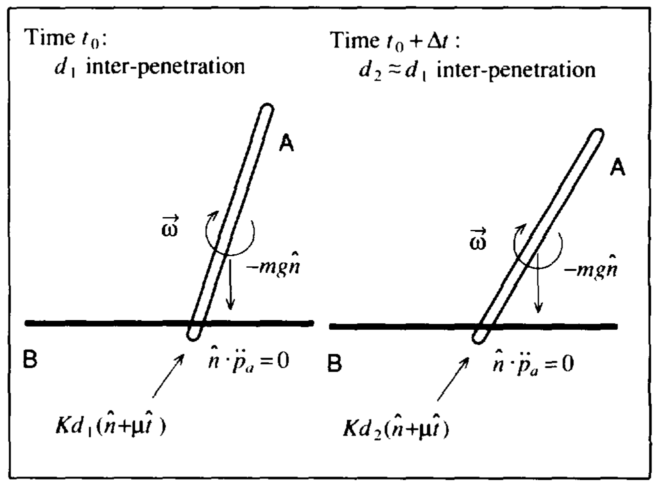

Figure 9. The initial inter-penetration is d_1. Both gravity and a penalty force act on A. A slides and falls without breaking contact with B.

Then the normal force magnitude at time t_0 is m. Since p_a is sliding on B, a friction force acts on A as shown. As A falls, maintaining contact with B, the inter-penetration varies smoothly, produce a varying normal force. At time $t_0 + \Delta t$, A still inter-penetrates B by an amount $d_1 \approx d_2$, and the behavior of the configuration is that of figure 3b. Thus, the initial amount of inter-penetration determines which behavior occurs.

The simulation method of computing and applying contact forces and impulses to bodies does not model inter-penetration. Instead of determining behavior by initial choice of inter-penetration, we can consider an initial normal force between bodies at contact points, and use that to determine subsequent behavior. For the applications we are interested in, we generally have no basis for preferring one set of initial normal forces over another. The numerical routines used for solving the contact force equations arbitrarily determine the behavior simulated. This may or may not be sensible for other applications.

6.3 The Principle of Constraints

The principle of constraints, applied to configurations with friction, states the following: when computing forces for a configuration of bodies, impulsive forces should be used only if non-impulsive forces do not exist for the configuration. In other words, if a configuration is consistent, non-impulsive forces should be computed and applied to the configuration; otherwise impulsive forces must be introduced into the system. Initially, this seems like a sensible principle, but we know of no real justification for it. If the physical model presented in this section is adopted, then this principle must be abandoned (at least in the context of rigid body simulation).

Consider figure 10. Once again, the combination of gravity and the angular velocity of R_1 is the same as in figures 5 and 6. Similarly, a horizontal acceleration of B results in inconsistency.

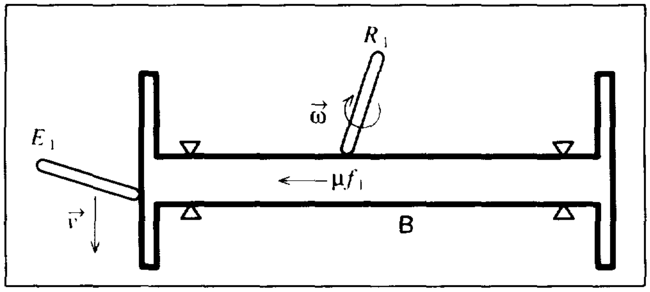

Figure 10. f_1 **must be either 0 or** m_1 **to be valid. However,** $f_1 = m_1$ **causes inconsistency. The only valid solution is** $f_1 = 0$.

If E_1 is ignored for the moment, then both $f_1 = 0$ and $f_1 = m_1$ are valid solutions for f_1. The only valid solution for the configuration as a whole though, is $f_1 = 0$; $f_1 = m_1$ pushes B to the left, causing inconsistency. However, using the physical model of indeterminacy, the value f_1 assumes depends on the initial inter-penetration between R_1 and B. If we adopt the physical model presented, we must conclude the following: even though the configuration is consistent, there is no *a priori* reason to prefer impulse-free behavior to non-impulse-free behavior for this configuration. The inconsistency resulting from $f_1 = m_1$, and subsequent application of impulsive contact forces is *as acceptable a behavior* as the application of non-impulsive contact forces resulting from $f_1 = 0$. Even though the configuration in figure 10 has only one valid solution of contact forces, ($f_1 = 0$), it has two possible behaviors and is thus indeterminate.

6.4 Reformulating the Contact Force Problem

Up to now, we have viewed the problem of computing valid contact forces as: given a configuration, efficiently compute a valid set of contact forces, *if they exist*. This viewpoint is based on the principle of constraints; that is, impulsive forces should be applied if and only if the configuration is inconsistent. It is this absolute *insistence* on a non-impulsive solution, if it exists, that makes the problem of computing contact forces so difficult. However, now that we have abandoned the principle of constraints, a different viewpoint of the problem is possible.

We reformulate the problem of computing contact forces as: given a configuration, efficiently compute *either* a valid set of contact forces *or* a valid set of contact impulses. (Validity for contact impulses is defined in section 7). Under the physical model we have assumed, there is no intrinsic reason to prefer valid contact forces over valid contact impulses.

By computing a particular set of valid contact forces or impulses, a particular behavior is chosen for the configuration, and other possible behaviors ignored. This means that we do not bother to decide if a configuration is consistent or not. If a valid set of contact impulses are computed, it will not be known if the configuration was consistent and could have been solved with contact forces; however, this is unimportant. In the next section, an efficient method is presented for computing valid contact forces or impulses.

7. Computing Valid Contact Forces and Impulses

Before an efficient method for computing either contact forces or impulses can be considered, the definition of validity must be extended to cover contact impulses. We first define validity for contact impulses and then present a computational algorithm.

7.1 Valid Contact Impulses

In the penalty method interpretation of figure 7, an impulse occurred because no matter how strong the normal force became, it was insufficient to prevent inter-penetration. As a result, after the contact impulse was applied, the relative velocity of the bodies at the contact point was directed inwards. Since contact impulses may need to be applied to configurations involving more than one contact point, validity must be defined for a set of contact impulses. For example, in figure 10, if the $f_1 = m_1$ behavior is chosen, a contact impulse should occur between E_1 and B. However, there should be no contact impulse between R_1 and B. In order for our definition of validity to be useful, all inconsistent configurations should have a valid set of contact impulses. We show in section 7.2 that our definition of validity for contact impulses satisfies this requirement.

We call a set of contact impulses valid under the following two conditions. First, the contact impulses must convert at least one of the contact points with dynamic friction to static friction. Second, every contact point at which a contact impulse occurs must end up with a non-positive relative normal velocity; that is, after the contact impulses are applied, bodies should *not* be separating wherever contact impulses occurred. The justification for this is that the contact impulses occur only when the normal force grows without bound to oppose inter-penetration. Intuitively, valid contact impulses are the limiting result of increasing normal forces without bound under the penalty method. If bodies are separating at a contact point after contact impulses are applied, then the normal force at the contact point should not have grown without bound into a contact impulse. As in section 6.1, bodies will be colliding at some contact points after valid contact impulses are applied, and a secondary set of impulses will have to be applied. These impulses may be calculated according to [16].

7.2 Computing Contact Forces and Impulses with Lemke's Algorithm

How can either contact forces or impulses be computed efficiently, given that computing contact forces alone is hard? In section 3, it was stated that every configuration of n contact points had an associated quadratic programming problem of n variables. Let a set of valid normal force magnitudes (if it exists) be denoted by the unknown n-vector $\vec{f}$; the magnitude of the ith normal force is given by $\vec{f}_i$. If $\vec{f}$ exists, it can be found by solving the QP

$$\underset{\vec{f}}{\text{minimize}}\ \vec{f}^{\,T}(A\vec{f} + \vec{b}) \text{ subject to } \begin{cases} \vec{f} \geq \vec{0} \\ A\vec{f} + \vec{b} \geq \vec{0} \end{cases} \tag{8}$$

where A and $\vec{b}$ are determined by the configuration. A is an $n \times n$ inverse mass matrix and $\vec{b}$ is an n-vector of known external and inertial accelerations. $A\vec{f} + \vec{b}$ represents the relative accelerations at contact points. (See [1, 2, 6, 11] for a discussion of the numerical properties of A and methods for computing A). Every

$\vec{f}$ such that equation (8) attains zero is valid. If equation (8) cannot attain zero subject to the above restrictions, then the configuration has no valid solution and is inconsistent. Thus, a valid $\vec{f}$ is a solution to the equation

$$\vec{f}^T(A\vec{f} + \vec{b}) = 0,\ \vec{f} \geq 0 \text{ and } A\vec{f} + \vec{b} \geq \vec{0}. \qquad (9)$$

Equation (9) is what is known as a *linear complementarity* (LCP) problem. Equation (9) is called a positive semidefinite (PSD) LCP if A is PSD[14].

One of the first algorithms for solving linear complementarity problems was introduced by Lemke[12] and is known as Lemke's algorithm. Lemke's algorithm is a pivoting method, similar to the simplex method of linear programming and has similar numerical properties. The algorithm is exponential in the worst case, but has an expected running time polynomial in n[14]. Lemke's algorithm progresses, like the simplex method, by trying various descent directions. If an LCP is PSD and has no solution then Lemke's algorithm will at some point encounter an *unbounded ray*; a descent direction along which one can travel infinitely far without making any progress. Otherwise, if a PSD LCP has a solution, then no unbounded ray exists for that LCP, and Lemke's algorithm terminates by finding a solution to the LCP. The algorithm is viewed as a practical solution method to the problem of solving PSD LCP's.

However, for non-PSD LCP's, Lemke's algorithm is not guaranteed to terminate correctly (although it still takes only expected polynomial time to do so). For a non-PSD LCP, if there is no solution, Lemke's algorithm terminates by encountering an unbounded ray. Unfortunately, if there is a solution, the algorithm is not guaranteed to find it. For non-PSD LCP's with solutions, Lemke's algorithm terminates either by finding a solution or by encountering an unbounded ray.[1] As a result, Lemke's algorithm is not suitable for solving non-PSD LCP's.

However, when Lemke's algorithm terminates by encountering an unbounded ray, it has found an n-vector $\vec{z}$ with the property[14]

$$\vec{z} \geq \vec{0} \text{ and } \forall i \text{ such that } \vec{z}_i > 0,\ (A\vec{z})_i \leq 0 \qquad (10)$$

where $(A\vec{z})_i$ is the ith component of the vector $A\vec{z}$. Why is this property of interest? Suppose that a set of contact impulses are applied to the configuration, with the magnitude of the normal impulse at the ith contact point denoted by $\vec{z}_i$. Then it can be shown[2, 6] that the relative velocity at the ith contact point after the impulse is $(A\vec{z})_i$. If the vector $\vec{z}$ satisfies equation (10) then every contact point subject to a non-zero contact impulse $\vec{z}_i > 0$ ends up with a non-positive relative normal velocity $(A\vec{z})_i \leq 0$. Thus, the vector $\vec{z}$ found by Lemke's algorithm gives rise to a valid set of contact impulses. To fully satisfy the definition of validity, $\vec{z}$ must be scaled upwards from zero until it causes a contact point with dynamic friction to be converted to static friction. After this, a real impact occurs, as described in section 6.1.

The behavior of Lemke's algorithm exactly matches our new view of the problem of computing contact forces. If the configuration has no valid contact impulse solutions, Lemke's algorithm cannot terminate with the special vector $\vec{z}$ and must therefore find a valid contact force solution. For inconsistent configurations, no valid contact force solution exists, so Lemke's algorithm must terminate with the vector $\vec{z}$, providing a contact impulse solution. For configurations with both a valid force and impulse solution, Lemke's algorithm will terminate by computing one or the other. Whenever Lemke's algorithm terminates by computing a contact impulse solution, it will still be unknown whether or not the configuration was consistent. For frictionless systems, the LCP is always PSD and has a solution, so frictionless configurations do not have valid impulse solutions. Thus, the reformulation of the problem does not add any new solutions to simulations of frictionless systems.

Although Lemke's algorithm runs, practically speaking, in polynomial time, this is not a proof that finding either valid contact forces or impulses is a polynomial time problem. From a practical standpoint, though, Lemke's algorithm provides an efficient algorithm for computing valid contact forces or impulses. The computational complexity of either solving an LCP or finding an unbounded ray is unknown.

8. Approaches for Static Friction

We conclude with two approaches to dealing with static friction. We stress that these approaches are only a first step towards dealing with the problems of static friction. Both approaches have their drawbacks, and currently have only limited applicability. The two approaches appear to produce (approximately the same) reasonably realistic results for the configurations we have simulated.

Consider the ith contact point of a configuration, and let the normal force magnitude there be f_i. The coefficient of friction, μ, is not indexed and may be different for each contact point. No distinction is made between the coefficient of static and dynamic friction, and both are assumed to be isotropic. In what follows, there is no difficulty in using a different value of μ depending on whether the friction force is static or dynamic. The next few computations take place in the tangent plane of the contact surface at each contact point; vectors are expressed in this plane as pairs (x,y) where $(1,0)$ and $(0,1)$ are orthonormal. Let (f_{x_i}, f_{y_i}) be the friction force, and (v_{x_i}, v_{y_i}) and (a_{x_i}, a_{y_i}) the relative tangential velocity and acceleration between bodies at the ith contact point.[2] If (v_{x_i}, v_{y_i}) is non-zero, then dynamic friction occurs and and the friction force has magnitude μf_i and is anti-parallel to the vector (v_{x_i}, v_{y_i}).

Static friction is more complex. For static friction,

$$|(f_{x_i}, f_{y_i})|^2 = f_{x_i}^2 + f_{y_i}^2 \leq (\mu f_i)^2. \qquad (11)$$

The main difficulty in static friction is determining when a contact points makes a transition from sticking to sliding. When the static friction force is sufficient to prevent sliding, any direction of the friction force constraining (a_{x_i}, a_{y_i}) to be zero is valid. If the body begins to slide, then (f_{x_i}, f_{y_i}) must at least partially oppose the acceleration; that is,

$$(f_{x_i}, f_{y_i}) \cdot (a_{x_i}, a_{y_i}) \leq 0. \qquad (12)$$

Also, if (a_{x_i}, a_{y_i}) is non-zero, then the friction force magnitude must attain its upper bound of μf_i. The law for static friction can be summarized as

$$f_{x_i}^2 + f_{y_i}^2 \leq (\mu f_i)^2,\ (f_{x_i}, f_{y_i}) \cdot (a_{x_i}, a_{y_i}) \leq 0 \text{ and}$$
$$((\mu f_i)^2 - (f_{x_i}^2 + f_{y_i}^2))(a_{x_i}^2 + a_{y_i}^2) = 0 \qquad (13)$$

where the last condition forces either $(\mu f_i)^2 = f_{x_i}^2 + f_{y_i}^2$ or $a_{x_i}^2 + a_{y_i}^2 = 0$. Unfortunately, equation (13) is too complex to be

[1]Encountering an unbounded ray when there is a solution is analogous to getting stuck at a non-global minimum in a non-convex minimization problem.

[2]Care must be taken here. The relative acceleration (a_{x_i}, a_{y_i}) is calculated by taking the first derivative of a velocity constraint, not the second derivative of a spatial constraint. See Goyal[8] for details.

formulated as part of a quadratic program. It also does not appear practical to solve with current non-linear programming techniques.

8.1 The Dynamic Friction Approximation

This approach for approximating static friction is extremely simple to implement. In order to determine whether static friction or dynamic friction should occur at a contact point, a simulator must have some threshold value ε. If $|(v_{x_i}, v_{y_i})| \geq \varepsilon$, then dynamic friction occurs. Otherwise $|(v_{x_i}, v_{y_i})| < \varepsilon$ and static friction occurs. Since dynamic friction is

$$(f_{x_i}, f_{y_i}) = \mu f_i \frac{-(v_{x_i}, v_{y_i})}{|(v_{x_i}, v_{y_i})|} \tag{14}$$

we approximate static friction as

$$(f_{x_i}, f_{y_i}) = \frac{|(v_{x_i}, v_{y_i})|}{\varepsilon} \mu f_i \frac{-(v_{x_i}, v_{y_i})}{|(v_{x_i}, v_{y_i})|} = \frac{-(v_{x_i}, v_{y_i})\mu f_i}{\varepsilon}. \tag{15}$$

Thus, we really use a dynamic friction force that varies in magnitude from zero to an upper limit of μf_i as the relative contact speed varies from 0 to ε. This allows us to use the method of section 7.2 to compute both static and dynamic friction.

Since static friction occurs only when the relative tangential velocity is non-zero, bodies must acquire some small amount of "crawl" in order to maintain a static friction force. This approach is reminiscent of the penalty method, where bodies must acquire some degree of inter-penetration for a sufficient normal force to exist. However, in the penalty method, it is necessary to increase the spring constant K without bound as the mass of bodies increases. Our approximation method does not suffer from this problem. If ε is made small enough, the "crawling" behavior of bodies is not visible, no matter what masses or forces exist. If ε is made excessively small, the differential equations of motion[1,3] may become stiff; otherwise, the approach has a reasonable performance. The major advantage to this approach is that it is guaranteed to produce a result, using Lemke's algorithm as described in section 7.2. Thus, either a set of contact forces or impulses is computed. The major disadvantage to this approach is that it is an *ad hoc* approximation to the law of static friction.

8.2 Modeling Static Friction by Quadratic Programming

This approach is much more ambitious. We attempt to model static friction as a quadratic programming problem, which can be solved to find the contact forces. We approximate the static friction law as follows. Equation (11) is rewritten as

$$-\mu f_i \leq f_{x_i} \leq \mu f_i \text{ and } -\mu f_i \leq f_{y_i} \leq \mu f_i . \tag{16}$$

Unfortunately, this allows the static friction force magnitude to exceed μf_i (by as much as a factor of $\frac{1}{2}\sqrt{2}$), unless the friction force happens to be aligned with a coordinate axis of the tangent plane. One possible solution is to iterate several times, trying to choose a coordinate system so either f_{x_i} or f_{y_i} is zero, for each contact point. For two-dimensional configurations however, the friction force is constrained to a line, not a plane, and is described by a single variable f_{x_i}. In this case, the constraint $-\mu f_i \leq f_{x_i} \leq \mu f_i$ is exact.

To satisfy equation (12), we add the conditions

$$\begin{aligned} &f_{x_i}\,\text{sgn}(f_{x_i}) \geq 0 \text{ and } a_{x_i}\,\text{sgn}(f_{x_i}) \leq 0 \\ &f_{y_i}\,\text{sgn}(f_{y_i}) \geq 0 \text{ and } a_{y_i}\,\text{sgn}(f_{y_i}) \leq 0 \end{aligned} \tag{17}$$

where $\text{sgn}(x) = 1$ if $x \geq 0$ and -1 otherwise. These conditions ensure that $(f_{x_i}, f_{y_i}) \cdot (a_{x_i}, a_{y_i}) \leq 0$. The condition that static friction attains its upper bound when slipping begins is written

$$\begin{aligned} &(\mu f_i - f_{x_i}\,\text{sgn}(f_{x_i}))(a_{x_i}\,\text{sgn}(f_{x_i})) = 0 \\ &(\mu f_i - f_{y_i}\,\text{sgn}(f_{y_i}))(a_{y_i}\,\text{sgn}(f_{y_i})) = 0. \end{aligned} \tag{18}$$

Finally, we add the standard constraint on the normal forces that

$$f_i \geq 0,\ a_i \geq 0 \text{ and } f_i a_i = 0 \tag{19}$$

where a_i is the relative normal acceleration of the ith contact point. If the signs of the f_{x_i} and f_{y_i} are known, then the above system of equations has unknown variables f_i, f_{x_i}, f_{y_i} (for each contact point) which are used to express the a_i, a_{x_i} and a_{y_i} terms. The entire system can be solved by a quadratic program because the *sgn* functions become known. How can the signs of the f_{x_i} and f_{y_i} variables be determined?

Iterative methods for quadratic programming and linear complementarity exist that can be adopted to this problem[14]. These iterative techniques are very similar to the Gauss-Seidel or Jacobi iterative methods used to solve linear systems. Iterative methods for quadratic programming are modified in a straightforward fashion to solve the system of equations (16) thru (19), without initially knowing the signs of the f_{x_i} and f_{y_i}. Unfortunately, convergence results are not available for the modified iterative methods. If the modified method fails to converge (or even before full convergence), the signs of f_{x_i} and f_{y_i} can be guessed by examining the unconverged solution. Quadratic programming is then used to solve equation (16) thru (19) as a quadratic program, given the estimate of the signs of the variables. If the estimate is correct, a solution is obtained for the friction forces.

However, the approach can break down at any number of places. If the method fails to converge, the estimates of the signs of the variables may not be correct. Even if the signs of the variables are correct, the form of the linear constraints in equation (16) do not allow us to use Lemke's algorithm for linear complementarity. Although we can apply standard quadratic programming methods, we know of no algorithm that will solve the quadratic program or indicate contact impulses, as Lemke's algorithm does. With regard to the entire issue of consistency and *NP*-hardness, this method for static friction is back to square one. It is possible that when the iterative step fails to converge, an analysis of the divergence of the iterates will indicate a valid set of contact impulses. At this time, however, we do not know how to perform such an analysis.

We have found however that the second approach, when it works, yields a very acceptable result. For large numbers of contact points ($n \approx 40$), the second approach sometimes breaks down, while the first approach does not. We have had reasonable success with the second approach for configurations with 40 contact points or less.

9. Conclusion

An efficient algorithm for dealing with configurations of bodies with only dynamic friction has been presented. Instead of attempting to force a behavior that avoids contact impulses, the algorithm allows either contact forces or contact impulses to occur. Two preliminary approaches for dealing with static friction are presented. The first approach is an approximation using the algorithm developed for simulating dynamic friction. The second approach is more exact but also more prone to failure than the first approach. Simulation of a complex configuration with static and dynamic friction is shown in figure 11.

Acknowledgements

This research was funded by an AT&T Bell Laboratories PhD Fellowship and two NSF grants (#DCR8203979 and #ASC87 15478). Simulations were performed on equipment generously donated by the Hewlett Packard Corporation and the Digital Equipment Corporation. I wish to thank both Bruce Donald and Andy Ruina for several stimulating conversations about *NP*-completeness results and frictional behavior.

Appendix A: Acceleration due to Contact Force

We compute the normal acceleration of p_a, $\hat{n}\cdot\ddot{p}_a$, for the configuration of figure 2. Let $\vec{a}$ and $\vec{\alpha}$ denote the linear and angular acceleration of A, $\vec{\omega}$, the angular velocity of A, and $\vec{r}$, the displacement of p_a from the center of mass of A. Vectors are treated as 3-space vectors: $\vec{a}$, p_a, $\ddot{p}_a$, $\hat{n}$ and $\vec{r}$ lie in the xy plane while $\vec{\omega}$ and $\vec{\alpha}$ are parallel to the z axis. From figure 1, $\vec{r} = (-\cos\theta, -\sin\theta, 0)$. $\ddot{p}_a$ may be expressed as the sum of three terms: the linear acceleration $\vec{a}$, the tangential acceleration $\vec{\alpha}\times\vec{r}$, and the centripetal acceleration $\vec{\omega}\times(\vec{\omega}\times\vec{r})$. The linear acceleration, $\vec{a}$, is

$$\vec{a} = \frac{f\hat{n} + \mu f\hat{t} + (-mg)\hat{n}}{m} = \frac{f\hat{n} + \mu f\hat{t}}{m} - g\hat{n}. \qquad (20)$$

The torque on A about its center of mass is $\vec{r}\times(f\hat{n} + \mu f\hat{t})$, which yields an angular acceleration of

$$\vec{\alpha} = \frac{\vec{r}\times(f\hat{n} + \mu f\hat{t})}{I}. \qquad (21)$$

Then

$$\begin{aligned}\ddot{p}_a &= \vec{a} + \vec{\alpha}\times\vec{r} + \vec{\omega}\times(\vec{\omega}\times\vec{r}) \\ &= \frac{f\hat{n} + \mu f\hat{t}}{m} - g\hat{n} + \frac{\vec{r}\times(f\hat{n} + \mu f\hat{t})}{I}\times\vec{r} + \vec{\omega}\times(\vec{\omega}\times\vec{r}). \qquad (22)\end{aligned}$$

Taking the dot product of equation (20) with $\hat{n}$,

$$\hat{n}\cdot\vec{a} = \frac{f\hat{n}\cdot\hat{n} + \mu f\hat{n}\cdot\hat{t}}{m} - g\hat{n}\cdot\hat{n} = \frac{f}{m} - g. \qquad (23)$$

Taking the dot product of the tangential acceleration $\vec{\alpha}\times\vec{r}$ with $\hat{n}$ and using the geometry of figure 1, $\hat{n}\cdot\vec{\omega}\times(\vec{\omega}\times\vec{r}) = |\vec{\omega}|^2\sin\theta$ and

$$\begin{aligned}\hat{n}\cdot(\vec{\alpha}\times\vec{r}) &= \hat{n}\cdot\left(\frac{\vec{r}\times(f\hat{n} + \mu f\hat{t})}{I}\times\vec{r}\right) \qquad (24) \\ &= \frac{f(\cos^2\theta - \mu\cos\theta\sin\theta)}{I}.\end{aligned}$$

Then, from the relations in figure 1,

$$\begin{aligned}\hat{n}\cdot\ddot{p}_a &= \frac{f}{m} - g + \frac{f(\cos^2\theta - \mu\cos\theta\sin\theta)}{I} + |\vec{\omega}|^2\sin\theta \\ &= \frac{f}{m}(1 + 16(\cos^2\theta - \mu\cos\theta\sin\theta)) + |\vec{\omega}|^2\sin\theta - g \qquad (25) \\ &= \frac{f}{m}(1 - 2) + |\vec{\omega}|^2\sin\theta - g = -\frac{f}{m} + (|\vec{\omega}|^2\sin\theta - g).\end{aligned}$$

References

1. Baraff, D., "Analytical methods for dynamic simulation of non-penetrating rigid bodies," *Computer Graphics (Proc. SIGGRAPH)*, vol. 23, pp. 223-232, 1989.
2. Baraff, D., "Curved surfaces and coherence for non-penetrating rigid body simulation," *Computer Graphics (Proc. SIGGRAPH)*, vol. 24, pp. 19-28, 1990.
3. Barzel, R. and Barr, A.H., "A modeling system based on dynamic constraints," *Computer Graphics (Proc. SIGGRAPH)*, vol. 22, pp. 179-188, 1988.
4. Cottle, R.W., "On a problem in linear inequalities," *Journal of the London Mathematical Society*, vol. 43, pp. 378-384, 1968.
5. Erdmann, M.A., *On Motion Planning with Uncertainty*, M.S. Thesis, Massachusetts Institute of Technology, 1984.
6. Featherstone, R., *Robot Dynamics Algorithms*, Kluwer, Boston, 1987.
7. Garey, M.R. and Johnson, D.S., *Computers and Intractability*, Freeman, New York, 1979.
8. Goyal, S., "Second order kinematic constraint between two bodies rolling, twisting and slipping against each other while maintaining point contact," *Technical Report TR 89-1043*, Department of Computer Science, Cornell University, 1989.
9. Kilmister, W. and Reeve, J.E., *Rational Mechanics*, Longman's, London, 1966.
10. Lötstedt, P., "Coulomb friction in two-dimensional rigid body systems," *Zeitschrift für Angewandte Mathematik un Mechanik*, vol. 61, pp. 605-615, 1981.
11. Lötstedt, P., "Numerical simulation of time-dependent contact friction problems in rigid body mechanics," *SIAM Journal of Scientific Statistical Computing*, vol. 5, no. 2, pp. 370-393, 1984.
12. Lemke, C.E., "Bimatrix equilibrium points and mathematical programming," *Management Science*, vol. 11, pp. 681-689, 1965.
13. Mason, M.T. and Wang, Y., "On the inconsistency of rigid-body frictional planar mechanics," *IEEE International Conference on Robotics and Automation*, 1988.
14. Murty, K.G., *Linear Complementarity, Linear and Nonlinear Programming*, Heldermann Verlag, Berlin, 1988.
15. Vavasis, S.A., "Quadratic Programming is in NP," *Technical Report TR 90-1099*, Department of Computer Science, Cornell University, 1990.
16. Wang, Y. and Mason, M.T., "Two dimensional rigid body collisions with friction," *Journal of Applied Mechanics*, (to appear).

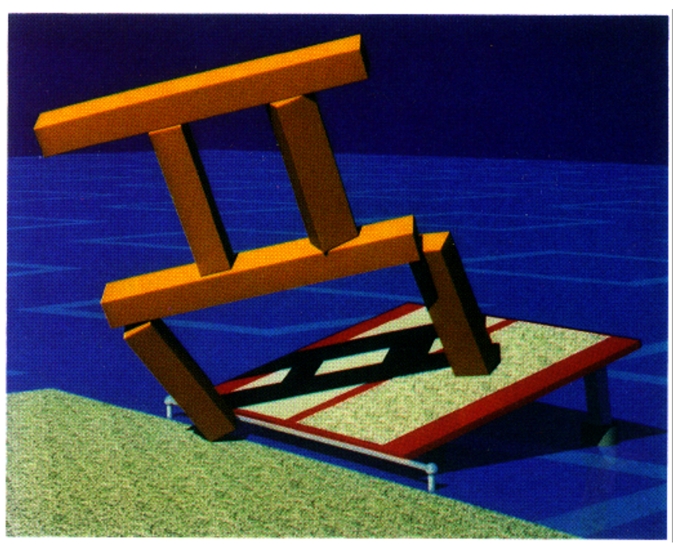

Figure 11. Simulation of a complex configuration.