Chapter 15

Future Directions

In this thesis, we have used a model of rigid body dynamics that is greatly simplified to examine the difficulties of simulation of complex physical systems. Even using a simplified model of rigid body dynamics, this thesis has shown that the problem of dynamic simulation with non-interpenetration constraints is a difficult problem. Beginning with the dynamics of frictionless configurations with a single point of contact, we have seen that producing a computationally tractable solution method for the contact force at the single contact point is non-trivial. Next, we considered properties and solution methods for configurations with multiple contact points and friction. The most difficult problems we encountered have centered on the behavior of configurations with friction. We have shown that the computational complexity of computing a non-impulsive solution for configurations without colliding contacts is *NP*-hard; more importantly, however, we have shown that it is not necessarily important to specifically compute such a solution, and that an absolute preference for such solutions over the course of a simulation violates causality and is thus unnatural. A model that allows expected polynomial time solution methods has been proposed, and iterative computational methods for computing contact forces

for configurations with and without collisions have been proposed.

A large number of graphical simulations have been produced using the methods described in this thesis, and images from several simulations are shown in Appendix D. Although these simulations yield results that seem visually correct, they are still simulations of a system subject to a number of simplifying assumptions, or restrictions. How severe have these restrictions been, and what is required to relax them?

The major restriction in the work described in this thesis has been the assumption that contact between bodies occurs at a finite number of contact points. For simulations with frictionless polyhedra, this restriction introduces no approximations into the simulation, since frictionless systems are determinate. However, this restriction prohibits simulations involving contact areas with curved boundaries, such as a cylinder standing upright on a plane. The problem with such a simulation is how to formulate the non-penetration constraint: since the cylinder could potentially be tipped over from any direction, every point on the circular boundary of the contact region (between the plane and the cylinder) must be included in a non-penetration constraint. Currently, it is unclear how one should write or attempt to solve the constraint equations for this case.

Interestingly, the same problem arises when considering the formulation of non-penetration constraints between flexible bodies that contact with a one- or two-dimensional contact region. Even if the contact region is polyhedral, inter-penetration is not necessarily prevented by constraints on the allowed motion of the vertices. A deformation that causes a flat surface to become curved could result in inter-penetration at an interior point of a contact region even though no inter-penetration had occurred on the boundary of that contact region[2]. Additionally, dealing with one- or two-dimensional contact regions also requires research into contact determination algorithms that can both detect and describe, in some fashion,

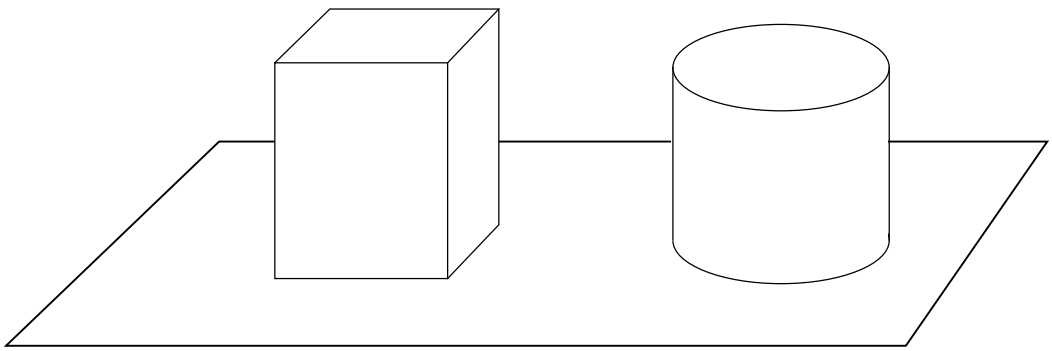


Figure 15.1: Non-polygonal contact region. The contact area between the cube and the plane is a polygon, while the contact area between the cylinder and the plane is a circle. The non-penetration constraint for the cube is posed in terms of the finitely many vertices describing the contact area boundary. For the cylinder, infinitely many points are necessary to describe the contact area boundary.

one- and two-dimensional contact regions. Finding collision/contact determination algorithms that can also deal with non-convex curved surfaces is also an interesting research problem.

For simulations with friction, the restriction to finitely many contact points is more severe. In essence, the tangential friction forces that should occur over a line or area of contact are restricted to act only at the vertices of the contact region. Again, the problem is how to formulate the Coulomb friction laws over an entire region of contact; this is a very open research problem.

Another problem is computing contact forces and impulses in the presence of friction. The assumption of constant slip velocity direction for three-dimensional collisions simplifies computations, but is unrealistic and must be dealt with. Also, we do not have a guaranteed solution method for configurations involving both static and dynamic friction. There is also the philosophical issue of what a simulation should do when faced with indeterminacy: this problem has been largely (if not completely) ignored by this thesis. There has been some speculation that both indeterminacy and inconsistency due to friction could be eliminated by discarding the restriction that bodies are perfectly rigid. Depending on the model of non-rigidity adopted, inconsistency and indeterminacy can either be partly or completely eliminated. For example, if a standard mass point and spring system is used to model deformable bodies, both inconsistency and indeterminacy are completely eliminated. In such a model, a given contact force affects the instantaneous acceleration of only a single mass point; the change in position of the mass point indirectly affects neighboring mass points via the spring forces. Since a mass point's instantaneous normal acceleration is unaffected by any tangential friction forces, neither indeterminacy nor inconsistency is possible. Note that this model of flexibility eliminates inconsistency and indeterminacy by giving the body an essentially unlimited number of degrees of freedom ($3n$ degrees of freedom for a body with n mass points). There also exists

a middle ground, in which bodies are allowed to deform, but with only a limited number of degrees of freedom. Witkin[56] and Baraff and Witkin[2] discuss flexible bodies that are limited to linear or quadratic deformations of a rest shape. Under this model, inconsistency and indeterminacy are only partly eliminated; the limited flexibility of the body eliminates inconsistency and indeterminacy in configurations with one contact point, but allows inconsistency and indeterminacy in configurations with two or more contact points. In general, one could speculate that the more degrees of freedom a model permits, the less likely indeterminacy and inconsistency can occur.

Appendix A

Extremal Point Conditions for Parametric Surfaces

The derivations in this thesis have all assumed implicit descriptions of curved surfaces. In this appendix, derivations for surfaces described parametrically are given. We will begin by defining the parametric analogue of Equation (5–2).

Let the surfaces A and B be modeled by the time-varying parametric functions $S(u,v,t)$ and $T(u,v,t)$. Componentwise,

$$S(u,v,t) = \begin{pmatrix} S_x(u,v,t) \\ S_y(u,v,t) \\ S_z(u,v,t) \end{pmatrix} \quad \text{and} \quad T(u,v,t) = \begin{pmatrix} T_x(u,v,t) \\ T_y(u,v,t) \\ T_z(u,v,t) \end{pmatrix}. \tag{A–1}$$

A normal vector $S_N(u,v,t)$ is given by

$$S_N(u,v,t) = \frac{\partial S}{\partial u}(u,v,t) \times \frac{\partial S}{\partial v}(u,v,t) \tag{A–2}$$

and similarly for T. If the extremal points p_a and p_b at time t are defined in terms of parametric coordinates (u_a, v_a) and (u_b, v_b) by

$$p_a = S(u_a, v_a, t) \quad \text{and} \quad p_b = T(u_b, v_b, t) \tag{A–3}$$

then the analogue of Equation (5–2) is

$$\begin{cases} E_1: & S_N(u_a, v_a, t) + \lambda_2 T_N(u_b, v_b, t) = \mathbf{0} \\ E_2: & (T(u_b, v_b, t) - S(u_a, v_a, t)) + \lambda_1 T_N(u_b, v_b, t) = \mathbf{0}. \end{cases} \tag{A–4}$$

The analogue of Equation (5–4) is

$$\begin{cases} S_N(u_a, v_a, t)^T r = 0 \\ (S(u_a, v_a, t) - (P_0 + c\, r)) + \lambda_1 S_N(u_a, v_a, t) = \mathbf{0}, \end{cases} \tag{A–5}$$

while the analogue of Equation (5–5) is

$$\begin{cases} (S(u_a, v_a, t) - P_0) + \lambda_1 S_N(u_a, v_a, t) = \mathbf{0}. \end{cases} \tag{A–6}$$

These equations can be solved to find the extremal points the during collision/ contact determination step.

In deriving an expression for $\ddot{C}$ for force determination, the functions S and T are treated as configuration dependent functions $S(u, v, \overline{p})$ and $T(u, v, \overline{p})$. Equation (A–4) then becomes

$$\begin{cases} S_N(u_a, v_a, \overline{x}(t)) + \lambda_2 T_N(u_b, v_b, \overline{x}(t)) = \mathbf{0} \\ (T(u_b, v_b, \overline{x}(t)) - S(u_a, v_a, \overline{x}(t))) + \lambda_1 T_N(u_b, v_b, \overline{x}(t)) = \mathbf{0}. \end{cases} \tag{A–7}$$

For force determination, it was also necessary to define functions $f(x, y, \overline{p})$ and $g(x, y, \overline{p})$ that explicitly describe the surfaces of A and B in a local neighborhood of the extremal points. For parametric functions, the analogue of Equation (6–14) is

$$f(S_x(u, v, \overline{x}(t)), S_y(u, v, \overline{x}(t)), \overline{x}(t)) = S_z(u, v, \overline{x}(t)) \tag{A–8}$$

and similarly for g and T. As in Section 6.4, the existence of f and g follows from the implicit function theorem. Partial derivatives of f and g can be expressed in terms of partial derivatives of S and T. If $\overline{p}$ has k coordinates, then differentiating

Equation (A–8) with respect to u, v, and $\overline{p}_1$ through $\overline{p}_k$ yields

$$\begin{array}{ccccccc}
\dfrac{\partial f}{\partial x}\dfrac{\partial S_x}{\partial u} & + & \dfrac{\partial f}{\partial y}\dfrac{\partial S_y}{\partial u} & & & = & \dfrac{\partial S_z}{\partial u} \\
\dfrac{\partial f}{\partial x}\dfrac{\partial S_x}{\partial v} & + & \dfrac{\partial f}{\partial y}\dfrac{\partial S_y}{\partial v} & & & = & \dfrac{\partial S_z}{\partial v} \\
\dfrac{\partial f}{\partial x}\dfrac{\partial S_x}{\partial \overline{p}_1} & + & \dfrac{\partial f}{\partial y}\dfrac{\partial S_y}{\partial \overline{p}_1} & + & \dfrac{\partial f}{\partial \overline{p}_1} & = & \dfrac{\partial S_z}{\partial \overline{p}_1} \\
\vdots & & \vdots & & \ddots & = & \vdots \\
\dfrac{\partial f}{\partial x}\dfrac{\partial S_x}{\partial \overline{p}_k} & + & \dfrac{\partial f}{\partial y}\dfrac{\partial S_y}{\partial \overline{p}_k} & & + \dfrac{\partial f}{\partial \overline{p}_k} & = & \dfrac{\partial S_z}{\partial \overline{p}_k}.
\end{array} \qquad \text{(A–9)}$$

Thus, the first partials of f may be expressed in terms of the first partials of S by solving a simpler linear system (note that Equation (A–9) is mostly diagonal). Second partials of f are obtained by differentiating Equation (A–9) and solving another linear system. These derivatives are needed for Equation (6–29), along with the derivatives with respect to time of the extremal points: applying the implicit function theorem to Equation (A–4) yields

$$\dot{u}_a = -\frac{\dfrac{\partial(E_1, E_2)}{\partial(t, v_a, u_b, v_b)}}{J} \qquad \text{and} \qquad \dot{v}_a = -\frac{\dfrac{\partial(E_1, E_2)}{\partial(u_a, t, u_b, v_b)}}{J} \qquad \text{(A–10)}$$

(and similarly for $\dot{u}_b$ and $\dot{v}_b$) where

$$J = \frac{\partial(E_1, E_2)}{\partial(u_a, v_a, u_b, v_b)}. \qquad \text{(A–11)}$$

Given the derivatives of the parametric coordinates, we can calculate $\dot{p}_a$ and $\dot{p}_b$. From Equation (A–1), and since $p_a = S(u_a, v_a, \overline{x}(t))$

$$\dot{p}_{ax} = \frac{\partial S_x}{\partial u}(u_a, v_a, \overline{x}(t))\dot{u}_a + \frac{\partial S_x}{\partial v}(u_a, v_a, \overline{x}(t))\dot{v}_a + \frac{\partial S_x}{\partial \overline{p}}(u_a, v_a, \overline{x}(t))\dot{\overline{x}}(t) \qquad \text{(A–12)}$$

and similarly for $\dot{p}_{ay}$, $\dot{p}_{bx}$ and $\dot{p}_{by}$.

Appendix B

Termination Conditions of Lemke's Algorithm

In this appendix, we will prove that when Lemke's algorithm fails to find a solution to Equation (8–7), it finds a vector $\mathbf{z}$ satisfying Equation (13–1); furthermore, for some $1 \leq i \leq n$ (where n is the size of the LCP of Equation (8–7)), the value z_i is positive, and the ith contact point of the system has dynamic friction. The proof is based on a proof by Murty[40, section 2.3, pages 86–88] that Lemke's algorithm, when applied to a copositive plus LCP,[1] either finds a solution, or terminates with an unbounded ray.

THEOREM 5. *When Lemke's algorithm fails to find a solution to the LCP of Equation* (8–7)*, it generates a vector* $\mathbf{z} \geq \mathbf{0}$ *with the following two properties.*

(1) *For all* $1 \leq i \leq n$*, if* $\mathrm{z_i} > 0$ *then* $(\mathbf{Az})_\mathrm{i} \leq 0$*.*

(2) *For at least one contact point of the system with dynamic friction, say the* i*th contact point,* $\mathrm{z_i} > 0$*.*

[1] An LCP is copositive plus if for all $\mathbf{x} \geq \mathbf{0}$, the matrix $\mathbf{A}$ in the LCP satisfies $\mathbf{x}^T\mathbf{Ax} \geq 0$, and $(\mathbf{A}^T + \mathbf{A})\mathbf{x} = \mathbf{0}$ whenever $\mathbf{x}^T\mathbf{Ax} = 0$.

PROOF. Since the ordering of contact points is arbitrary, let the contact points of the system be ordered such that contact points 1 to k are all frictionless, and contact points $k+1$ to n have dynamic friction. Let the matrix $\mathbf{A}$ and constant vector $\mathbf{b}$ corresponding to this reordered system be partitioned by

$$\mathbf{A} = \begin{pmatrix} \mathbf{M} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix} \qquad \text{and} \qquad \mathbf{b} = \begin{pmatrix} \mathbf{b_1} \\ \mathbf{b_2} \end{pmatrix} \tag{B–1}$$

where $\mathbf{M}$ is a $k \times k$ matrix, $\mathbf{B}$ is a $k \times (n-k)$ matrix, $\mathbf{C}$ is an $(n-k) \times k$ matrix, $\mathbf{D}$ is an $(n-k) \times (n-k)$ matrix, $\mathbf{b_1}$ is a k-vector, and $\mathbf{b_2}$ is an $(n-k)$-vector. Since the first k points are frictionless, the matrix $\mathbf{M}$ is PSD, and the vector $\mathbf{b_1}$ is in the range of $\mathbf{M}$. Thus, as established in Chapter 8, the LCP

$$\mathbf{a_N} = \mathbf{M}\boldsymbol{f_N} - \mathbf{b_1} \geq \mathbf{0}, \quad \boldsymbol{f_N} \geq \mathbf{0}, \qquad \text{and} \qquad \boldsymbol{f_N}^T \mathbf{a_N} = 0 \tag{B–2}$$

has a solution.

The proof that a vector satisfying property (1) exists is essentially a restatement of the termination conditions of Lemke's algorithm as detailed by Murty. For convenience, in this appendix the vector $\mathbf{e}$ is taken to be an appropriately sized vector of ones; that is, $e_i = 1$ for all i. When Lemke's algorithm fails to find a solution to the LCP of Equation (B–1), it finds vectors $\mathbf{z}^h$, $\mathbf{z}^k$, $\mathbf{w}^h$, $\mathbf{w}^k$ and scalars z_0^h and z_0^k satisfying

$$\begin{gathered} \mathbf{w}^k = \mathbf{A}\mathbf{z}^k - \mathbf{b} + z_0^k \mathbf{e} \\ \mathbf{w}^h = \mathbf{A}\mathbf{z}^h + z_0^h \mathbf{e} \\ \mathbf{w}^k \geq \mathbf{0}, \quad \mathbf{w}^h \geq \mathbf{0}, \quad \mathbf{z}^k \geq \mathbf{0}, \quad \mathbf{z}^h \geq \mathbf{0} \text{ (but } \mathbf{z}^h \neq \mathbf{0}), \\ z_0^h \geq 0, \quad z_0^k > 0 \end{gathered} \tag{B–3}$$

and

$$(\mathbf{w}^k + \lambda \mathbf{w}^h)^T (\mathbf{z}^k + \lambda \mathbf{z}^h) = 0 \qquad \text{for all } \lambda \geq 0\,. \tag{B–4}$$

From Equations (B–3) and (B–4), it follows that

$$(\mathbf{w}^h)^T\mathbf{z}^h = 0, \quad (\mathbf{w}^k)^T\mathbf{z}^k = 0, \quad (\mathbf{w}^k)^T\mathbf{z}^h = 0 \qquad \text{and} \qquad (\mathbf{w}^h)^T\mathbf{z}^k = 0. \tag{B–5}$$

From this, it follows that if $\mathbf{z}_i^h > 0$ then $\mathbf{w}_i^h = 0$. But if $\mathbf{z}_i^h > 0$, then

$$0 = \mathbf{w}_i^h = (\mathbf{A}\mathbf{z}^h)_i + (z_0^h\mathbf{e})_i \tag{B–6}$$

so

$$(\mathbf{A}\mathbf{z}^h)_i = -(z_0^h\mathbf{e})_i = -z_0^h e_i = -z_0^h \leq 0. \tag{B–7}$$

Thus, the vector $\mathbf{z}^h$ satisfies property (1) of the proof.

In order for $\mathbf{z}^h$ to satisfy property (2), at least one non-zero element of $\mathbf{z}^h$ must correspond to a contact point with dynamic friction. Let $\mathbf{z}^h$, $\mathbf{w}^k$ and $\mathbf{w}^h$ be partioned as follows:

$$\mathbf{w}^h = \begin{pmatrix} \mathbf{v}^h \\ \mathbf{u}^h \end{pmatrix}, \quad \mathbf{w}^k = \begin{pmatrix} \mathbf{v}^k \\ \mathbf{u}^k \end{pmatrix}, \qquad \text{and} \qquad \mathbf{z}^h = \begin{pmatrix} \mathbf{x}^h \\ \mathbf{y}^h \end{pmatrix}. \tag{B–8}$$

The vectors $\mathbf{v}^h$, $\mathbf{v}^k$ and $\mathbf{x}^h$ have dimension k, while $\mathbf{u}^h$, $\mathbf{u}^k$ and $\mathbf{y}^h$ have dimension $n-k$. To show that property (2) is true for $\mathbf{z}^h$, we use proof by contradiction. Suppose that $\mathbf{y}^h = \mathbf{0}$; that is, suppose that all elements of $\mathbf{z}^h$ that correspond to a contact point with dynamic friction are zero. Then $\mathbf{z}^h$ would have the form

$$\mathbf{z}^h = \begin{pmatrix} \mathbf{x}^h \\ \mathbf{y}^h \end{pmatrix} = \begin{pmatrix} \mathbf{x}^h \\ \mathbf{0} \end{pmatrix}. \tag{B–9}$$

Now suppose that $z_0^h > 0$. Then since $\mathbf{z}^h$ is non-zero, for some $1 \leq j \leq k$, we have $z_j^h > 0$. As before, if $z_j^h > 0$ then $w_j^h = 0$. But then

$$0 = w_j^h = (\mathbf{A}\mathbf{z}^h)_j + (z_0^h\mathbf{e})_j = (\mathbf{A}\mathbf{z}^h)_j + z_0^h \tag{B–10}$$

so

$$(\mathbf{A}\mathbf{z}^h)_j = -z_0^h < 0. \tag{B–11}$$

Since $\mathbf{z}^h$ satisfies property (1),

$$(\mathbf{z}^h)^T \mathbf{A}\mathbf{z}^h = \sum_{i=1}^{n} z_i^h (\mathbf{A}\mathbf{z}^h)_i \leq 0; \tag{B–12}$$

but $(\mathbf{A}\mathbf{z}^h)_j = -z_0^h < 0$ and $z_j^h > 0$ for some $1 \leq j \leq k$ implies that $(\mathbf{z}^h)^T \mathbf{A}\mathbf{z}^h < 0$. Now, since $(\mathbf{z}^h)^T = (\mathbf{x}^h, \mathbf{0})^T$,

$$0 > (\mathbf{z}^h)^T \mathbf{A}\mathbf{z}^h = ((\mathbf{x}^h)^T, \mathbf{0}^T) \begin{pmatrix} \mathbf{M} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix} \begin{pmatrix} \mathbf{x}^h \\ \mathbf{0} \end{pmatrix} = (\mathbf{x}^h)^T \mathbf{M}\mathbf{x}^h. \tag{B–13}$$

But this contradicts the fact that $\mathbf{M}$ is PSD; thus $z_0^h \neq 0$ leads to a contradiction.

Conversely, suppose that $z_0^h = 0$. Since $(\mathbf{x}^h)^T \mathbf{M}\mathbf{x}^h = (\mathbf{z}^h)^T \mathbf{A}\mathbf{z}^h \leq 0$, and $\mathbf{M}$ is PSD, it must be that $(\mathbf{x}^h)^T \mathbf{M}\mathbf{x}^h = 0$. However, the fact that $\mathbf{M}$ is PSD means that $\mathbf{M}$ is also copositive plus; this in turn implies that $(\mathbf{M}^T + \mathbf{M})\mathbf{x}^h = \mathbf{0}$. Since $\mathbf{M}$ is symmetric as well, it follows that $\mathbf{M}\mathbf{x}^h = \mathbf{0}$.

Since

$$\mathbf{w}^k = \mathbf{A}\mathbf{z}^k - \mathbf{b} + z_0^k \mathbf{e},$$

from the partitioning of $\mathbf{A}$, $\mathbf{w}^k$, and $\mathbf{z}^k$,

$$\mathbf{u}^k = \mathbf{M}\mathbf{x}^k - \mathbf{b_1} + z_0^k \mathbf{e}. \tag{B–14}$$

Then since $(\mathbf{x}^h)^T \mathbf{M} = (\mathbf{M}\mathbf{x}^h)^T = \mathbf{0}^T$,

$$\begin{aligned} (\mathbf{x}^h)^T \mathbf{u}^k &= (\mathbf{x}^h)^T (\mathbf{M}\mathbf{x}^k - \mathbf{b_1} + z_0^k \mathbf{e}) \\ &= (\mathbf{x}^h)^T \mathbf{M}\mathbf{x}^k - (\mathbf{x}^h)^T \mathbf{b_1} + (\mathbf{x}^h)^T z_0^k \mathbf{e} \\ &= -(\mathbf{x}^h)^T \mathbf{b_1} + (\mathbf{x}^h)^T z_0^k \mathbf{e}. \end{aligned}$$

Since $\mathbf{w}^k$ and $\mathbf{z}^h$ satisfy $(\mathbf{w}_i^k)(\mathbf{z}_i^h) = 0$, it follows that $(\mathbf{u}^k)^T \mathbf{x}^h = 0$, so

$$(\mathbf{x}^h)^T \mathbf{u}^k = 0 = -(\mathbf{x}^h)^T \mathbf{b_1} + (\mathbf{x}^h)^T z_0^k \mathbf{e} \tag{B–15}$$

or equivalently

$$-(\mathbf{x}^h)^T\mathbf{b_1} = -(\mathbf{x}^h)^T z_0^k \mathbf{e}. \tag{B–16}$$

Since $\mathbf{x}^h \geq 0$, $\mathbf{e} > 0$ and $z_0^k > 0$, and $x_i^h > 0$ for some i,

$$-(\mathbf{x}^h)^T\mathbf{b_1} < 0. \tag{B–17}$$

Letting I denote the $k \times k$ identity matrix, and $(I \mid -M)$ the $k \times 2k$ matrix formed by catenating I and $-M$, we have

$$(\mathbf{x}^h)^T(I \mid -M) = (\mathbf{x}^h)^T - (\mathbf{x}^h)^T\mathbf{M} = (\mathbf{x}^h)^T > \mathbf{0} \quad \text{and}$$
$$-(\mathbf{x}^h)^T\mathbf{b_1} < 0. \tag{B–18}$$

By Farkas' lemma[40, appendix 1] this implies that the system

$$(I \mid -M)\begin{pmatrix} \mathbf{a_N} \\ \boldsymbol{f_N} \end{pmatrix} = -\mathbf{b_1} \qquad \text{and} \qquad \begin{pmatrix} \mathbf{a_N} \\ \boldsymbol{f_N} \end{pmatrix} \geq \mathbf{0} \tag{B–19}$$

is *infeasible*; that is, no $\boldsymbol{f_N}$ and $\mathbf{a_N}$ satisfying Equation (B–19) exists. This means that

$$(I \mid -M)\begin{pmatrix} \mathbf{a_N} \\ \boldsymbol{f_N} \end{pmatrix} = \mathbf{a_N} - \mathbf{M}\boldsymbol{f_N} = -\mathbf{b_1} \qquad \text{and} \qquad \begin{pmatrix} \mathbf{a_N} \\ \boldsymbol{f_N} \end{pmatrix} \geq \mathbf{0} \tag{B–20}$$

or equivalently

$$\mathbf{a_N} = \mathbf{M}\boldsymbol{f_N} - \mathbf{b_1} \qquad \text{and} \qquad \begin{pmatrix} \mathbf{a_N} \\ \boldsymbol{f_N} \end{pmatrix} \geq \mathbf{0} \tag{B–21}$$

is infeasible. However, this contradicts the fact that Equation (B–2) always has a solution, since it is the LCP of a frictionless configuration. Thus, $z_0^h = 0$ leads to a contradiction.

Since both $z_0^h = 0$ and $z_0^h \neq 0$ lead to contradictions, the original assumption that $\mathbf{y}^h = 0$ must in fact be false. Thus, at least one element z_i^h of $\mathbf{z}^h$ corresponding to a contact point with dynamic friction satisfies $z_i^h > 0$, and $\mathbf{z}^h$ satisfies both properties (1) and (2). □

Appendix C

Iterative Solution Methods for Static Friction

In this appendix, a Jacobi-like iterative method for solving Equation (14–1) is derived. As stated in Section 14.2.3, the iterative method is applied long enough to estimate the signs of the unknown variables. Gauss-Seidel, Jacobi, and successive overrelaxation (SOR) iterations have been developed for solving linear complementarity problems[9,33]. The Jacobi-like iteration for solving Equation (14–1) developed in this appendix can easily be made into either a Gauss-Seidel- or SOR-like iterative method.

Linear equations

Consider the linear equation

$$\mathbf{A}\mathbf{z} + \mathbf{q} = \mathbf{0} \tag{C–1}$$

where $\mathbf{A}$ is an $n \times n$ matrix, $\mathbf{q}$ is an n-vector, and $\mathbf{z}$ is the unknown n-vector being solved for. Iterative methods generate a sequence of solutions $\{\mathbf{z}^{(0)}, \mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \dots\}$ that (hopefully) converge to a vector $\mathbf{z}$ satisfying Equation (C–1).

The Jacobi iteration for computing $\mathbf{z}^{(k+1)}$ is the following.

$$\begin{aligned}
&\mathbf{e}^{(k)} = \mathbf{A}\mathbf{z}^{(k)} + \mathbf{q} \\
&\textbf{for } i = 1 \textbf{ to } n \\
&\qquad z_i^{(k+1)} = z_i^{(k)} - e_i^{(k)}/A_{ii} \\
&\textbf{done}
\end{aligned}$$

Note that if $\mathbf{z}^{(k)}$ exactly solves Equation (C–1), then $\mathbf{e}^{(k+1)} = \mathbf{A}\mathbf{z} + \mathbf{q} = \mathbf{0}$, and $\mathbf{z}^{(k+1)} = \mathbf{z}^{(k)}$. In practice, the iteration is stopped when $\mathbf{e}^{(k)}$ becomes suitably small. The Jacobi iteration is known to converge if $\mathbf{A}$ is positive definite (PD); in this case, the diagonal elements A_{ii} are all positive.

Linear complementarity

Now suppose that instead of solving Equation (C–1), we wish to find $\mathbf{z}$ such that

$$\mathbf{A}\mathbf{z} + \mathbf{q} \geq \mathbf{0}, \quad \mathbf{z} \geq \mathbf{0} \qquad \text{and} \qquad \mathbf{z}^T(\mathbf{M}\mathbf{z} + \mathbf{q}) = 0. \tag{C–2}$$

The Jacobi iteration for solving Equation (C–2) involves a *projection* of $\mathbf{z}^{(k)}$, to maintain the constraint $\mathbf{z} \geq \mathbf{0}$. The projection of a scalar x so that it is non-negative is written x_+ and is defined as

$$x_+ = \max(0, x). \tag{C–3}$$

The Jacobi iteration for solving Equation (C–2) is simply the Jacobi iteration for solving Equation (C–1), with the addition of a projection; that is,

$$\begin{aligned}
&\mathbf{e}^{(k)} = \mathbf{A}\mathbf{z}^{(k)} + \mathbf{q} \\
&\textbf{for } i = 1 \textbf{ to } n \\
&\qquad z_i^{(k+1)} = (z_i^{(k)} - e_i^{(k)}/A_{ii})_+ \\
&\textbf{done}
\end{aligned}$$

This is called the projected Jacobi iteration.

If $\mathbf{z}^{(k)}$ satisfies Equation (C–2), then $z_i^{(k)} > 0$ implies

$$e_i^{(k)} = (\mathbf{A}\mathbf{z}^{(k)} + \mathbf{q})_i = 0.$$

Thus $z_i^{(k+1)} = (z_i^{(k)})_+ = z^{(k)}$. Conversely, if $z_i^{(k)} = 0$, then

$$e_i^{(k)} = (\mathbf{A}\mathbf{z}^{(k)} + \mathbf{q})_i > 0, \tag{C–4}$$

so

$$z_i^{(k+1)} = (-e_i^{(k)}/A_{ii})_+ = 0$$

assuming $A_{ii} > 0$. (As a minimum, iterative methods generally assume the diagonal elements of $\mathbf{A}$ are positive.) Thus, if $\mathbf{z}^{(k)}$ satisfies Equation (C–2), then $\mathbf{z}^{(k+1)} = \mathbf{z}^{(k)}$. If $\mathbf{A}$ is PD then the projected Jacobi iteration will always converge[33].

Static friction equations

We will use the notation of Equations (14–6), (14–7), and (14–8) in defining the vector of accelerations, $\mathbf{a}$, and the vector of forces, $\boldsymbol{f}$, along with the known matrix $\mathbf{A}$ and vector $\mathbf{b}$. (To be consistent, we will write $\mathbf{q} = -\mathbf{b}$, so that $\mathbf{a} = \mathbf{A}\boldsymbol{f} + \mathbf{q}$.) These equations define f_{xi}, f_{y_i}, a_{xi} and a_{y_i} for every contact point. If the configuration of bodies is such that the ith contact point does not have static friction, then the variables f_{xi}, f_{y_i}, a_{xi}, and a_{y_i} are left out of Equations (14–7) and (14–8); the corresponding steps for computing f_{xi} and f_{y_i} in the Jacobi-like iteration are ignored as well. For simplicity, if $f_{N\,i}$, f_{xi} and f_{y_i} are the pth, qth and rth elements of $\boldsymbol{f}$, then the diagonal elements of $\mathbf{A}$ corresponding to these indices are denoted $D_{Ni} = A_{pp}$, and $D_{xi} = A_{qq}$ and $D_{y_i} = A_{rr}$. If the ith contact point has dynamic friction, it is possible that D_{Ni} may be zero or negative if the value of μ_i (the coefficient of friction at the contact point) is non-zero. If this happens, then

it is trivial to apply a valid contact impulse to the system by imposing an impulse at just the ith contact point, converting the dynamic friction to static friction. Otherwise, the ith contact has static friction and D_{Ni} is positive. Thus, without loss of generality, we may assume $D_{Ni} > 0$ in all cases. Additionally, both D_{xi} and D_{y_i} are positive[28]. The Jacobi-like iteration to solve Equation (14–1) is

$$\mathbf{a}^{(k)} = \mathbf{A}\boldsymbol{f}^{(k)} + \mathbf{q}$$

for $i = 1$ **to** n

$$f_{N\,i}{}^{(k+1)} = (f_{N\,i}{}^{(k)} - a_{N\,i}{}^{(k+1)}/D_{Ni})_+$$

$$v_f = ((f_{xi}{}^{(k)})^2 + (f_{y_i}{}^{(k)})^2)^{1/2}$$

$$p = f_{xi}{}^{(k)} a_{xi}{}^{(k)} + f_{xi}{}^{(k)} a_{xi}{}^{(k)}$$

if $v_f < \mu_i f_{N\,i}{}^{(k)}$ **or** $p > 0$ **then**

$$f_{xi}{}^{(k+1)} = f_{xi}{}^{(k)} - a_{xi}{}^{(k)}/D_{xi}$$

$$f_{y_i}{}^{(k+1)} = f_{y_i}{}^{(k)} - a_{y_i}{}^{(k)}/D_{y_i}$$

end

$$v'_f = ((f_{xi}{}^{(k+1)})^2 + (f_{y_i}{}^{(k+1)})^2)^{1/2}$$

if $v'_f > \mu_i f_{N\,i}{}^{(k)}$ **then**

$$f_{xi}{}^{(k+1)} = f_{xi}{}^{(k+1)}(\mu_i f_{N\,i}{}^{(k)}/v'_f)$$

$$f_{y_i}{}^{(k+1)} = f_{y_i}{}^{(k+1)}(\mu_i f_{N\,i}{}^{(k)}/v'_f)$$

end

done

The assignment to $f_{N\,i}{}^{(k+1)}$ is the same as in the linear complementarity case. After this, the iteration proceeds as follows. The value v_f is the magnitude of the friction force. The value p is the dot product of the friction force and the tangential acceleration. If the magnitude of the friction force, v_f, has not exceeded its upper bound of $\mu_i f_{N\,i}{}^{(k)}$, or if the friction and the acceleration are not opposite pointing

($p > 0$), then the friction forces are modified. The assignments

$$\begin{aligned} {f_{xi}}^{(k+1)} &= {f_{xi}}^{(k)} - {a_{xi}}^{(k)}/D_{xi} \\ {f_{y_i}}^{(k+1)} &= {f_{y_i}}^{(k)} - {a_{y_i}}^{(k)}/D_{y_i} \end{aligned}$$

attempt to modify the friction forces so that the tangential accelerations will be zero. Following this, the magnitude v_f' of the (possibly) modified friction forces is computed. If the friction force magnitude exceeds its upper bound of $\mu_i {f_{N\,i}}^{(k)}$, the friction forces are suitably scaled (by $\mu_i {f_{N\,i}}^{(k)}/v_f'$) so that their magnitude exactly attains this upper bound. As in the previous two Jacobi algorithms, when $\boldsymbol{f}^{(k)}$ satisfies Equation (14–1), then $\boldsymbol{f}^{(k+1)} = \boldsymbol{f}^{(k)}$. As stated, the purpose of this algorithm is to examine the signs of the variables after some number of iterations, as a prediction to solve the QP of Section 14.2.3.

Appendix D

Simulation Examples

This appendix contains images from simulations produced using the methods described in this thesis. Figure D.1 shows a simulation of a jack rolling down stairs. Figure D.2 shows a simulation of superquadric dice falling through a lattice. Figure D.3 shows a simulation of a collapsing structure with friction.

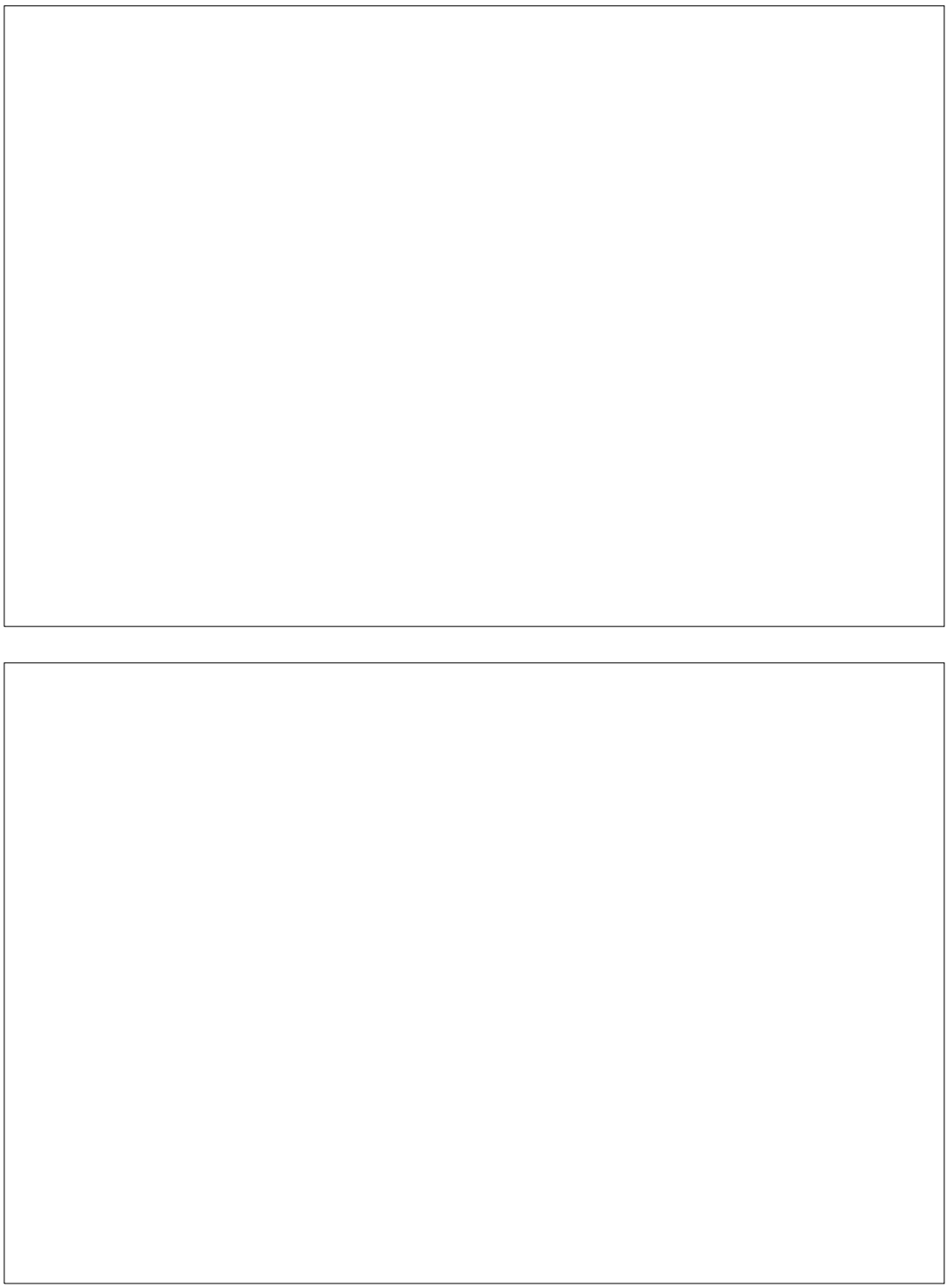
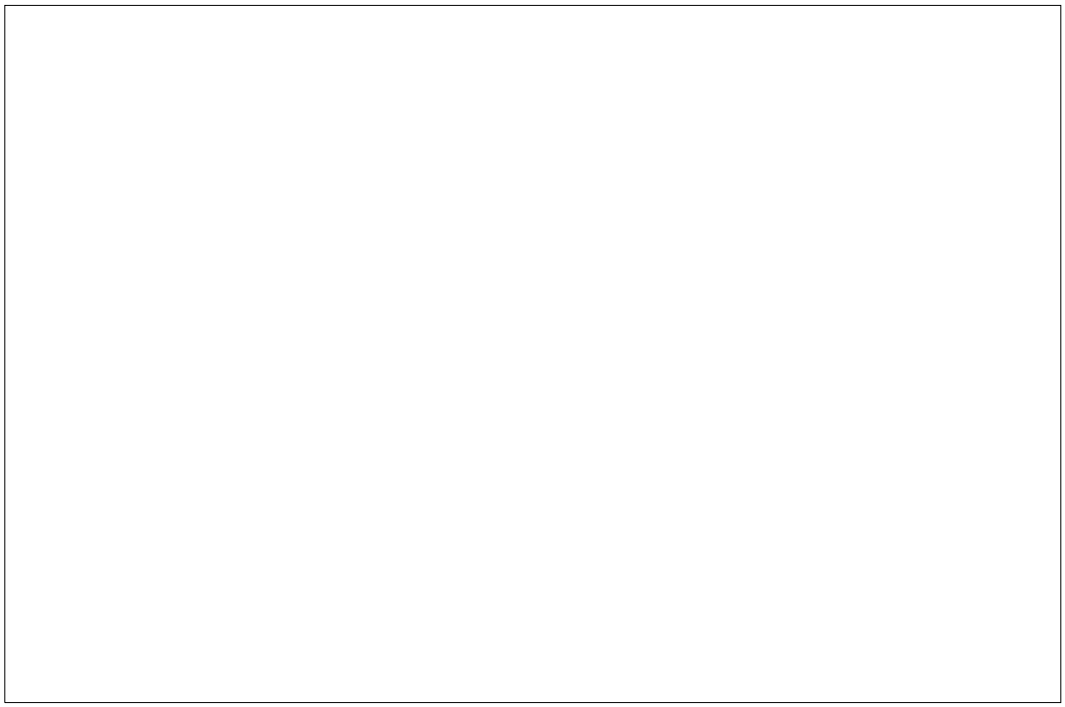


Figure D.1: Jack falling down stairs.

Figure D.1 (Continued)



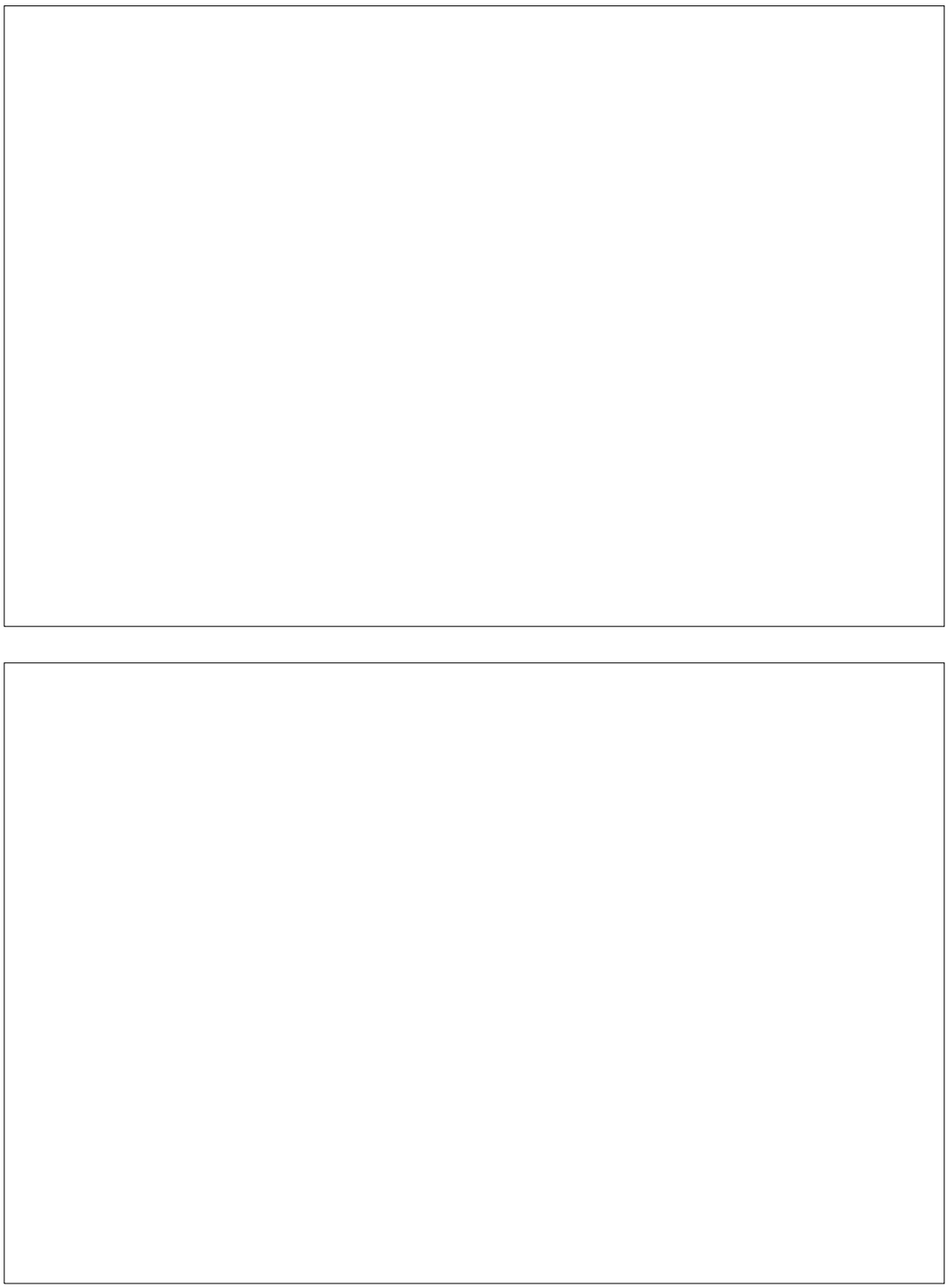
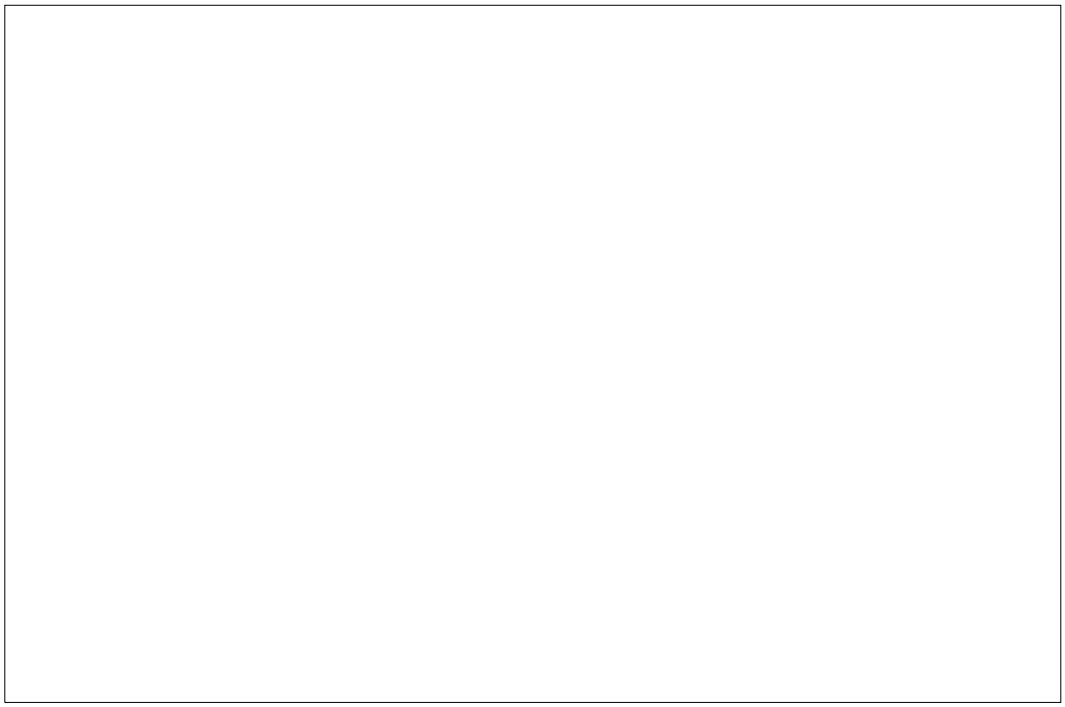
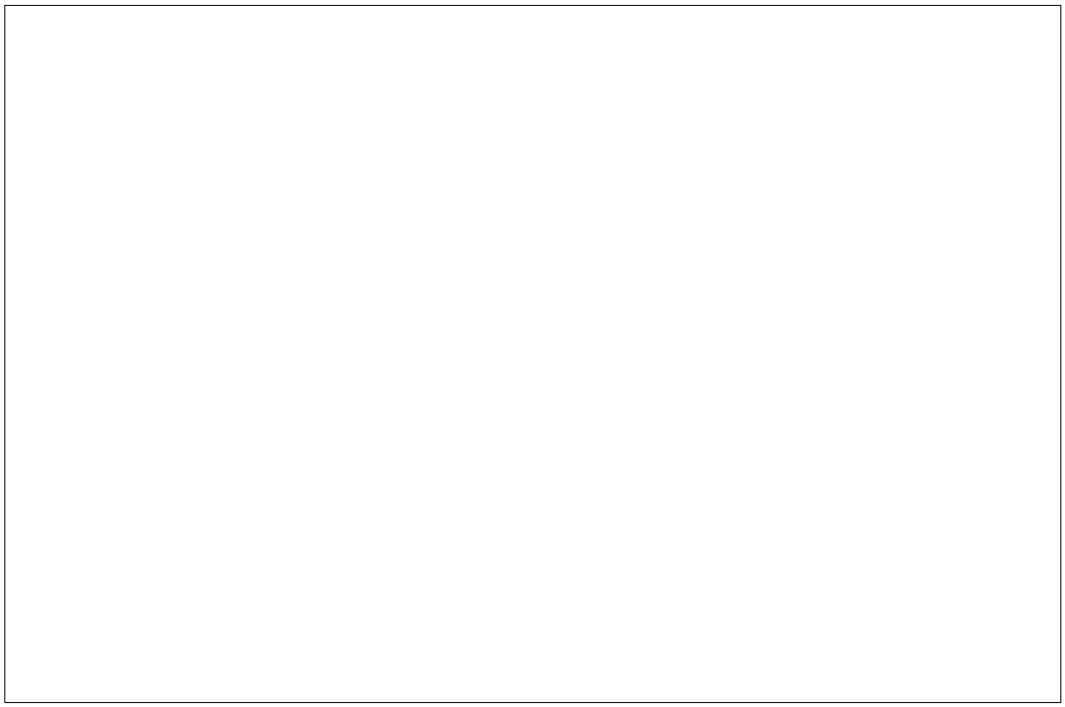


Figure D.2: Falling superquadric dice.

Figure D.2 (Continued)





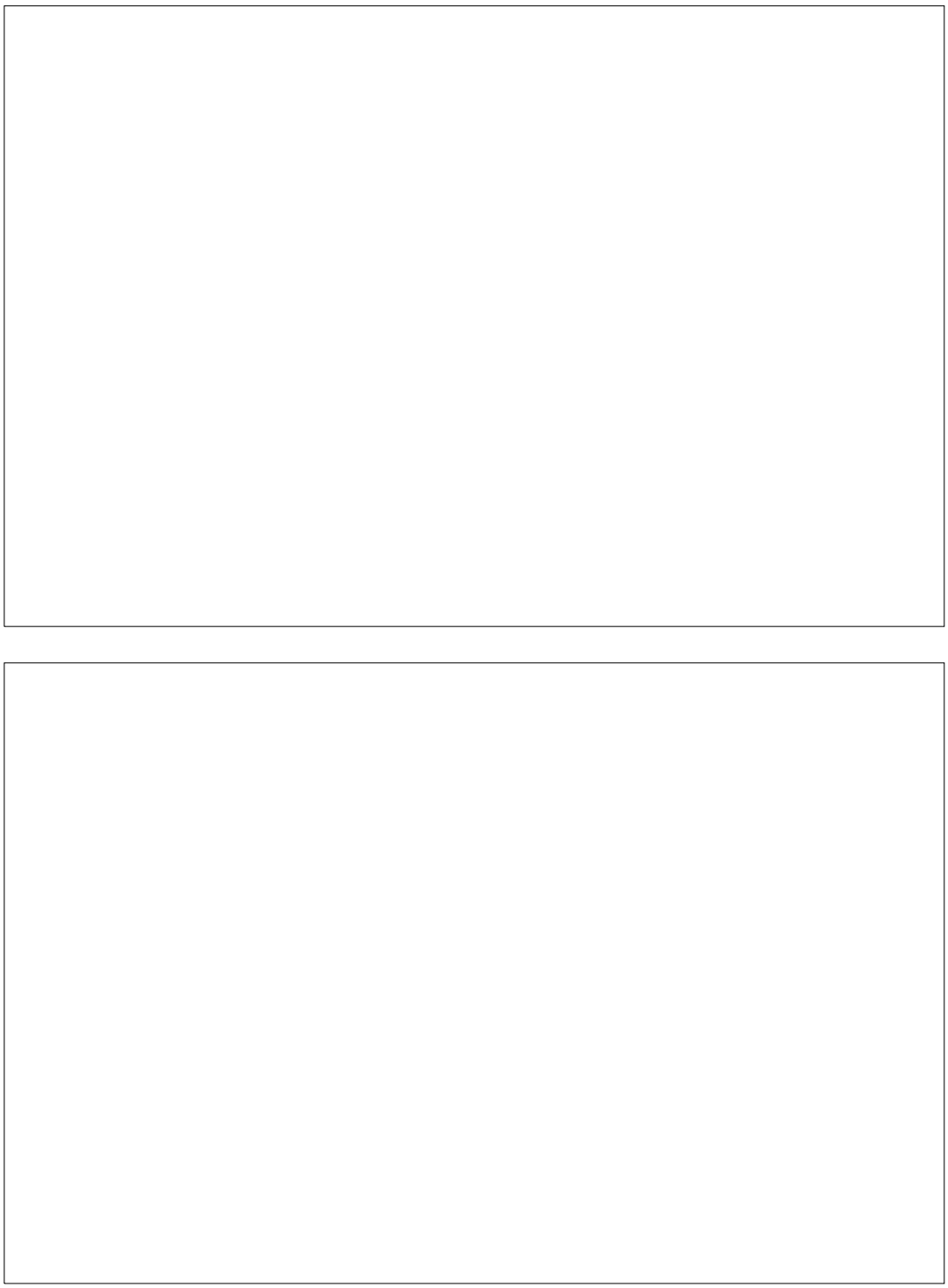
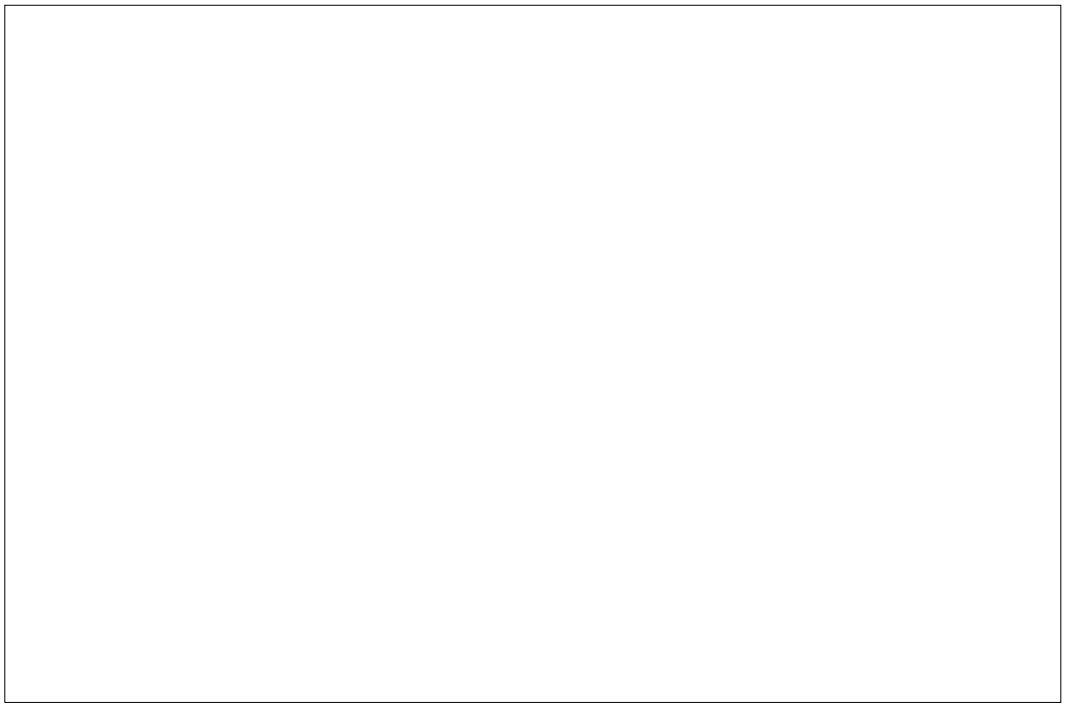
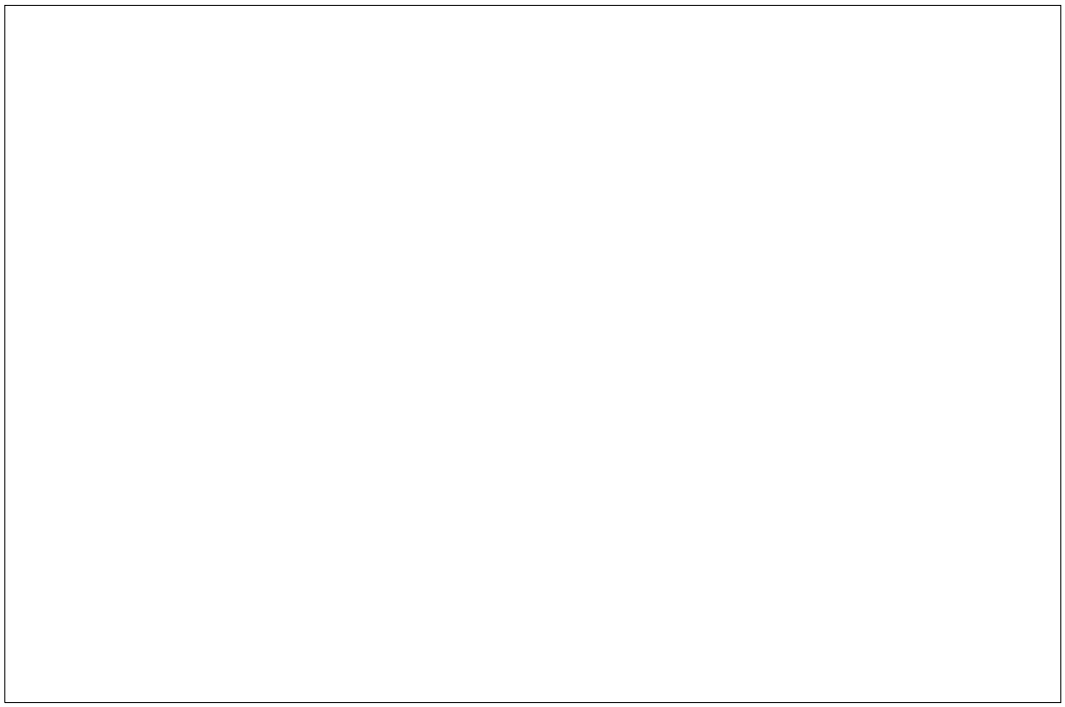


Figure D.3: Collapsing structure.

Figure D.3 (Continued)





Bibliography

[1] V.I. Arnold. *Mathematical Methods of Classical Mechanics.* Springer-Verlag, 1978.

[2] D. Baraff and A. Witkin. Dynamic simulation of non-penetrating flexible bodies. Submitted to *Computer Graphics* (*Proc. SIGGRAPH*), 1992.

[3] M.R. Brach. Rigid body collisions. *Journal of Applied Mechanics*, pages 164–170, 1989.

[4] C. Cai and B. Roth. On the spatial motion of a rigid body with point contact. In *International Conference on Robotics and Automation*, pages 686–695. IEEE, 1987.

[5] S.A. Cameron and R.K. Culley. Determining the minimum translational distance between two convex polyhedra. In *International Conference on Robotics and Automation*, pages 591–596. IEEE, 1986.

[6] J. Canny. Collision detection for moving polyhedra. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(2), 1986.

[7] R.W. Cottle. On a problem in linear inequalities. *Journal of the London Mathematical Society*, 43:378–384, 1968.

[8] J.F. Cremer. *An Architecture for General Purpose Physical System Simulation.* PhD thesis, Cornell University, April 1989.

[9] C.W. Cryer. The solution of a quadratic programming problem using systematic overrelaxation. *SIAM Journal on Control*, 9(3):385–392, 1971.

[10] P.A. Cundall. Formulation of a three-dimensional distinct element model—Part I. A scheme to represent contacts in a system composed of many polyhedral blocks. *International Journal of Rock Mechanics, Mineral Science and Geomechanics*, 25, 1988.

[11] J.E. Dennis, Jr. and R.B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice Hall, Inc., 1983.

[12] M.A. Erdmann. On motion planning with uncertainty. Master's thesis, Massachusetts Institute of Technology, 1984.

[13] R. Featherstone. *Robot Dynamics Algorithms.* Kluwer, 1987.

[14] R. Fletcher. *Practical Methods of Optimization*, volume 2. John Wiley & Sons, 1981.

[15] G.E. Forsythe, M.A. Malcolm, and C.B. Moler. *Computer Methods for Mathematical Computations.* Prentice Hall, Inc., 1977.

[16] M.R. Garey and D.S. Johnson. *Computers and Intractability.* Freeman, 1979.

[17] E.G. Gilbert and C.P. Foo. Computing the distance between smooth objects in three dimensional space. In *International Conference on Robotics and Automation*, pages 158–163. IEEE, 1989.

[18] E.G. Gilbert and S.M. Hong. A new algorithm for detecting the collision of moving objects. In *International Conference on Robotics and Automation*, pages 8–13. IEEE, 1989.

[19] E.G. Gilbert, D.W. Johnson, and S.S. Keerthi. A fast procedure for computing the distance between complex objects in three space. In *International Conference on Robotics and Automation*, pages 1883–1889. IEEE, 1987.

[20] B.J. Gilmore and R.J. Cipra. Simulation of planar dynamic mechanical systems with changing topologies: Part 1—characterization and prediction of the kinematic constraint changes; Part 2—implementation strategy and simulation results for example dynamic systems. In *Advances in Design Automation*, pages 369–388. ASME, 1987.

[21] H. Goldstein. *Classical Mechanics.* Addison-Wesley, Reading, 1983.

[22] J.P. Gourret, N.M. Thalmann, and D. Thalmann. Simulation of object and human skin deformations in a grasping task. In *Computer Graphics* (*Proc. SIGGRAPH*), volume 23, pages 21–30. ACM, July 1989.

[23] W. Ingleton. A problem in linear inequalities. *Proceedings of the London Mathematical Society*, 16(3):519–536, 1966.

[24] J.B. Keller. Impact with friction. *Journal of Applied Mechanics*, 53(1):1–4, 1986.

[25] W. Kilmister and J.E. Reeve. *Rational Mechanics*. Longman's, 1966.

[26] B. Lafleur, N. Magnenat-Thalmann, and D. Thalmann. Cloth animation with self-collision detection. In T.L. Kunii, editor, *Modeling in Computer Graphics*. Springer-Verlag, 1991.

[27] C.E. Lemke. Bimatrix equilibrium points and mathematical programming. *Management Science*, 11:681–689, 1965.

[28] P. Lötstedt. Coulomb friction in two-dimensional rigid body systems. *Zeitschrift für Angewandte Mathematik un Mechanik*, 61:605–615, 1981.

[29] P. Lötstedt. Mechanical systems of rigid bodies subject to unilateral constraints. *SIAM Journal of Applied Mathematics*, 42(2):281–296, 1982.

[30] P. Lötstedt. Numerical simulation of time-dependent contact friction problems in rigid body mechanics. *SIAM Journal of Scientific Statistical Computing*, 5(2):370–393, 1984.

[31] T. Lozano-Pérez. Automatic planning of manipulator transfer movements. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-11(10):681–698, 1981.

[32] T. Lozano-Pérez. Spatial planning: A configuration space approach. *IEEE Transactions on Computers*, C-32(2):108–120, 1983.

[33] O.L. Mangasarian. Solution of symmetric linear complementarity problems by iterative methods. *Journal of Optimization Theory and Applications*, 22(4):465–485, 1977.

[34] M.T. Mason and Y. Wang. Modeling impact dynamics for robotic operations. In *International Conference on Robotics and Automation*, pages 678–685. IEEE, 1987.

[35] M.T. Mason and Y. Wang. On the inconsistency of rigid-body frictional planar mechanics. In *International Conference on Robotics and Automation*, pages 524–528. IEEE, 1988.

[36] M. McKenna and D. Zeltzer. Dynamic simulation of autonomous legged locomotion. In *Computer Graphics* (*Proc. SIGGRAPH*), volume 24, pages 29–38. ACM, August 1990.

[37] N. Megiddo. Linear time algorithms for linear programming in R^3 and related problems. *SIAM Journal of Computing*, 12(4):759–776, 1983.

[38] P.M. Moore. A flexible object animation system. Master's thesis, University of California, Santa Cruz, 1987.

[39] P.M. Moore and J. Wilhelms. Collision detection and reponse for computer animation. In *Computer Graphics* (*Proc. SIGGRAPH*), volume 22, pages 289–298. ACM, August 1988.

[40] K.G. Murty. *Linear Complementarity, Linear and Nonlinear Programming.* Heldermann Verlag, 1988.

[41] Ju.I. Neĭmark and N.A. Fufaev. *Dynamics of Nonholonomic Systems.* American Mathematical Society, 1972.

[42] R.S. Palmer. *Computational Complexity of Motion and Stability of Polygons.* PhD thesis, Cornell University, January 1987.

[43] F.P. Preparata and M.I. Shamos. *Computational Geometry.* Springer-Verlag, New York, 1985.

[44] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes.* Cambridge University Press, 1986.

[45] H. Rubin and P. Ungar. Motion under a strong constraining force. *Communications on Pure and Applied Mathematics*, 10:65–87, 1957.

[46] R. Sedgewick. *Algorithms.* Addison-Wesley, 1983.

[47] L.F. Shampine and M.K. Gordon. *Computer Solution of Ordinary Differential Equations: The Initial Value Problem.* Freeman, 1975.

[48] K. Shoemake. Animating rotation with quaternion curves. In *Computer Graphics* (*Proc. SIGGRAPH*), volume 19, pages 245–254. ACM, July 1985.

[49] A.E. Taylor and R.M. Mann. *Advanced Calculus.* John Wiley & Sons, Inc., 1983.

[50] D. Terzopoulos and K. Fleischer. Deformable models. *Visual Computer*, 4:306–331, 1988.

[51] D. Terzopoulos, J.C. Platt, and A.H. Barr. Elastically deformable models. In *Computer Graphics* (*Proc. SIGGRAPH*), volume 21, pages 205–214. ACM, August 1987.

[52] S.A. Vavasis. Quadratic programming is in *NP.* Technical Report TR 90-1099, Department of Computer Science, Cornell University, 1990.

[53] B. Von Herzen, A. Barr, and H. Zatz. Geometric collisions for time-dependent parametric surfaces. In *Computer Graphics* (*Proc. SIGGRAPH*), volume 24, pages 39–48. ACM, August 1990.

[54] Y. Wang and M.T. Mason. Two dimensional rigid body collisions with friction. *Journal of Applied Mechanics*, (to appear).

[55] J. Wilhelms. Toward automatic motion control. *IEEE Computer Graphics and Applications*, 7(4):11–22, 1987.

[56] A. Witkin and W. Welch. Fast animation and control of nonrigid structures. In *Computer Graphics* (*Proc. SIGGRAPH*), volume 24, pages 243–252. ACM, August 1990.