Collapse

Günter Rote* Uri Zwick†

Abstract

The problem of checking whether a given tower of bricks is stable can be easily answered by checking whether a system of linear inequalities has a feasible solution. A more challenging problem is to determine *how* an *unstable* tower of bricks collapses. We use Gauß' *principle of least restraint* to show that this, and more general rigid-body simulation problems in which many parts touch each other, can be reduced to solving a sequence of convex quadratic programs, with linear constraints, corresponding to a discretization of time. The first of these quadratic programs gives an exact description of initial infinitesimal collapse. The results of the subsequent programs need to be integrated over time to yield an approximation of the global motion of the system.

1 Introduction

Paterson and Zwick (SODA'06) [16, 17] (see also Paterson et al. (SODA'08) [14, 15]) have asked for the maximum *overhang* that can be achieved with a stable tower of n identical bricks (without rotating them), see Figure 1. Before one can go about designing such a tower, the basic question is whether a given tower of bricks is stable. It is known that this question can be reduced to solving a system of linear inequalities, whose variables model the forces exchanged between bricks lying on top of each other.

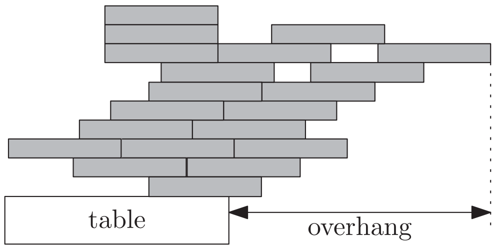

Figure 1: The overhang of a tower of bricks

We go here one step further: for the case of an *unstable* tower, as in Figure 2, we want to know *how* it will collapse. This problem has two versions. In the easier version, we are just interested in the *initial collapse*, i.e., in the accelerations of the different bricks when the tower, which is initially at rest, starts collapsing. We want to determine which bricks remain in contact and which bricks start to lift off from each other. We show that an exact solution of the initial collapse problem can be obtained by solving a single convex quadratic optimization problem. In the more demanding version, we are interested in an accurate description of the positions, velocities and accelerations of all the bricks as a function of time. This is, of course, a much more challenging task. We get accurate approximations of all these quantities by discretizing time and solving a sequence of convex quadratic problems.

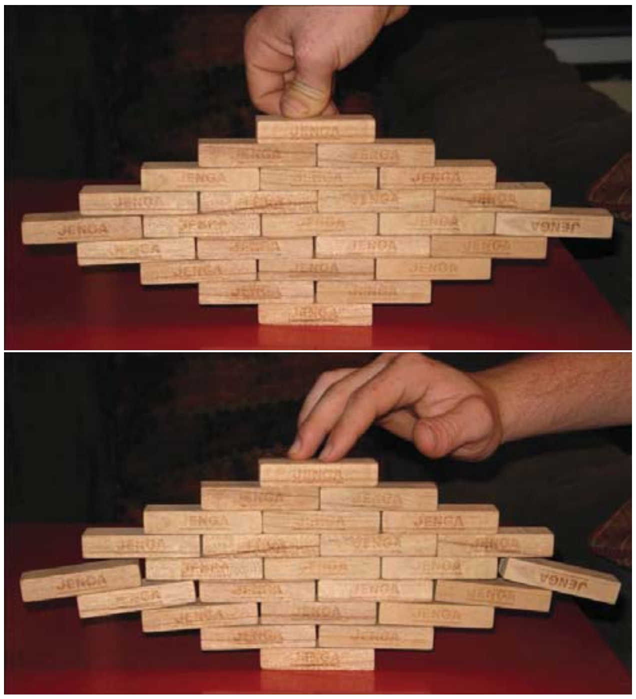

Figure 2: A tower of 5×5 bricks, which is unstable

This is a problem of rigid-body simulation. The motion (velocity and acceleration) at any particular instant can be computed by solving a quadratic optimization problem subject to linear constraints that is obtained using Gauß' principle of least restraint, which dates back to 1829 [11], see Section 3. The constraints

*Freie Universität Berlin. E-mail: rote@inf.fu-berlin.de.
†Tel Aviv University, E-mail: zwick@tau.ac.il.

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

of this system model the fact that the bricks are rigid and are not allowed to overlap, see Section 2. The particular constraints that have to be imposed at some instant depend on the current positions and velocities of the bricks (for example, which pairs of bricks are in contact). The solution of the optimization problem results in infinitesimal motions (accelerations and velocities). These motions define a system of ordinary differential equations. Integrating this system over time produces the global motion.

Related Work. Physical simulations arise in a variety of areas. In many applications like computer games, a realistic impression is all that is needed. In other circumstances, closeness to reality is important, for example in industrial settings when mixing granular materials. According to the well-known Brazil nut effect [21], in a mixture that is shaken, larger grains will rise to the top. There is, however, sometimes the opposite effect [23], and our understanding of this phenomenon is far from complete.

When rigid-body simulation is treated in computer graphics, the focus lies on fast computation of a motion that *looks* natural enough. Algorithms are typically illustrating by a large number of balls running through an hourglass or objects falling on the ground. Interactions between touching objects are usually handled on a pairwise basis, cf. [18, 10, 9, 3, 24, 22].

By contrast, our goal is to compute a mathematically correct answer soundly based on physical principles, apart from the necessity to perform discrete time steps instead of a continuous motion.

A few papers have used Gauß' principle for rigid-body simulation, see Bruyninckx and Khatib [4] and Redon et al. [20]. They were impeded by the fact that software for quadratic programming has not readily been available. Baraff [1, 2] has formulated the necessary consistency conditions on contact forces as a linear complementarity problem, noting that they can be solved as convex quadratic programming problems (and thus implicitly rederiving Gauß' principle, without mentioning it). Similarly, Lötstedt [12] used a quadratic programming formulation, without referring to Gauß' principle. He solves the quadratic optimization problems using an active set method. His paper includes an error analysis of the numerical methods for solving the system of differential equations resulting from the description of the motion.

Milenkovic and Schmidl [13] intuitively proposed a variety of quadratic programming formulations in terms of translation and rotation parameters (without a connection to Gauß' principle or any other physical justification). They performed simulations with up to 1000 parts.

The typical algorithm loops along the steps shown in Figure 3. In this feedback loop, the result of the

a) Determine the constraints at the current time step.

b) Solve the quadratic optimization problem, resulting in accelerations for each vertex.

c) Update the velocities and positions for the next time step.

d) (Optionally) Correct for errors resulting from the discretization.

Figure 3: The basic loop of rigid-body simulation

motion computed in step (c) in turn determines the constraints of the optimization problem (b) whose solution gives the next infinitesimal motion. The time discretization of the differential equation leads to systematic errors. If they are not corrected, they can lead to a deformation of rigid objects. The determination of the active constraints in (a) is actually very delicate.

Our contribution. We provide (in Section 5) a purely position-based simulation step that can be used to replace the simulation step given in Figure 3. (In other words, the variables of the quadratic optimization problem are now positions, and not velocities or accelerations.) This greatly simplifies the process and produces, we believe, more accurate results. An additional advantage of our formulation is that it deals automatically with *collisions* and *impulses*, if we decide to accept totally *inelastic* collisions where the total energy of the impulse is destroyed, see Section 7. The algorithm has been implemented as a prototype, see Section 8.

Outlook. For simplicity, we restrict our attention to a two-dimensional world. The extension to three (or higher) dimensions is straightforward. Completely rigid objects are of course an idealization, and we abstract from many aspects that are relevant in actual materials. The algorithm can accommodate any model of friction, as, for example, in [2, 12], and it can be extended to handle collisions (see Section 7), but we ignore friction completely, and we do only a rudimentary handling of (completely inelastic) collisions. Still, we view our work as a step towards the goal of realistic and reliable simulations.

2 Constraints on the Motion

We assume that the objects are rigid polygons that may be attached to each other at vertices. We model them as a bar-and-joint framework as shown in Figure 4.

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

Additional diagonal bars are inserted into polygons

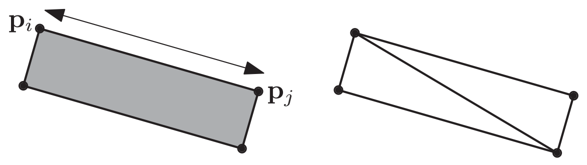

Figure 4: A brick made of rigid bars

(triangulation) to make them rigid. Each vertex i has a position $\mathbf{p}_i(t) = \begin{pmatrix} x_i(t) \\ y_i(t) \end{pmatrix}$ which depends on the time t, a velocity $\mathbf{v}_i(t) = \dot{\mathbf{p}}_i(t) = \begin{pmatrix} \dot{x}_i(t) \\ \dot{y}_i(t) \end{pmatrix}$ and an acceleration $\mathbf{a}_i(t) = \dot{\mathbf{v}}_i(t) = \ddot{\mathbf{p}}_i(t) = \begin{pmatrix} \ddot{x}_i(t) \\ \ddot{y}_i(t) \end{pmatrix}$.

The constraints are of two types. (i) Length constraints: each bar (i,j) has a fixed length: $\|\mathbf{p}_i(t) - \mathbf{p}_j(t)\|$ is constant, and consequently the derivatives are zero. It is more convenient look at the derivatives of the *squared* lengths:

$$\text{(L1)} \qquad \tfrac{\mathrm{d}}{\mathrm{d}t}\|\mathbf{p}_i(t) - \mathbf{p}_j(t)\|^2 = 0$$

$$\text{(L2)} \qquad \tfrac{\mathrm{d}^2}{\mathrm{d}t^2}\|\mathbf{p}_i(t) - \mathbf{p}_j(t)\|^2 = 0$$

(ii) Non-penetration constraints: A vertex is $\mathbf{p}_k$ not allowed to enter another brick by crossing an edge $\mathbf{p}_i\mathbf{p}_j$, see Figure 5a. We model this by a *sidedness constraint* saying that the (doubled) signed area $A(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k)$ of the triangle $\mathbf{p}_i\mathbf{p}_j\mathbf{p}_k$ is nonnegative (Figure 5b).

$$A(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k) = \begin{vmatrix} 1 & 1 & 1 \\ x_i & x_j & x_k \\ y_i & y_j & y_k \end{vmatrix} \geq 0$$

Note that this is a *local* condition, which has to prevent penetration when $\mathbf{p}_k$ is near the edge $\mathbf{p}_i\mathbf{p}_j$. The point $\mathbf{p}_k$ may cross the line through $\mathbf{p}_i\mathbf{p}_j$ by making a "global" detour (Figure 5c), and then $A(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k)$ becomes negative.

The first-order condition that prevents $A(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k) = 0$ from becoming negative is as follows:

(S1) If the point $\mathbf{p}_k$ lies on the segment $\mathbf{p}_i\mathbf{p}_j$ (and hence $A(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k) = 0$), then $\frac{\mathrm{d}}{\mathrm{d}t}A(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k) \geq 0$.

For given positions $\mathbf{p}_i$, this is a linear inequality in the velocities (first derivatives).

In terms of accelerations (second derivatives), the condition is written as follows:

(S2) If the point $\mathbf{p}_k$ lies on the segment $\mathbf{p}_i\mathbf{p}_j$ (and hence $A(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k) = 0$) *and* $\frac{\mathrm{d}}{\mathrm{d}t}A(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k) = 0$, then $\frac{\mathrm{d}^2}{\mathrm{d}t^2}A(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k) \geq 0$.

Thus, for given positions $(\mathbf{p}_i)$ and velocities $(\mathbf{v}_i)$ at some time instant, we have the following linear constraints that govern the possible accelerations $(\mathbf{a}_i)$: the length equations,

$$\text{(L2)} \qquad \frac{\mathrm{d}^2}{\mathrm{d}t^2}\|\mathbf{p}_i - \mathbf{p}_j\|^2 = 0, \text{ for all bars } (i,j),$$

and the sidedness inequalities:

$$\text{(S2)} \quad \frac{\mathrm{d}^2}{\mathrm{d}t^2}A(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k) \geq 0, \text{ for certain triples } (i,j,k).$$

After expanding the derivatives, the constraints become linear inequalities in the variables $(\mathbf{a}_i)$, see Section 4.

3 The Principle of Least Restraint

We now assume that each vertex i is a point of mass m_i, and some external force field $(\mathbf{g}_i)$ is applied at the vertex. In our case, we have gravity, which is uniform everywhere and so we just write $\mathbf{g}$. If there were no constraints, each point would just follow the external force and we would set $\mathbf{a}_i = \mathbf{g}$. Gauß' principle of least restraint (Prinzip des kleinsten Zwanges) [11], see also [4, 20], states that the system will minimize the weighted sum of the squared differences between the desired accelerations $\mathbf{g}$ and the actual accelerations $\mathbf{a}_i$:[1]

$$\text{(Q)} \qquad \text{minimize} \sum_{i=1}^{n} m_i \cdot \|\mathbf{a}_i - \mathbf{g}\|^2$$

subject to (L2) and (S2).

This is an optimization problem with a convex quadratic objective function and linear constraints. It turns out that the dual variables for the constraints can be interpreted as internal forces between vertices that are joined by bars or push against other bars. These internal forces balance the external forces with the acceleration forces and maintain an equilibrium at every vertex [19]. Gauß derived his principle from d'Alembert's *method of virtual velocities* (virtual displacements).

Gauß' principle is stated in terms of *accelerations* (see [8]), and this is also how it is commonly used in the robotic simulation literature [4, 20]. The algorithm shown in Figure 6 follows the outline of Figure 3. It takes current positions $\mathbf{p}_i^{(\ell)}$ and velocities $\mathbf{v}_i^{(\ell)}$ for step ℓ, computes new accelerations $\mathbf{a}_i$ and uses them for generating new positions $\mathbf{p}_i^{(\ell+1)}$ and velocities $\mathbf{v}_i^{(\ell+1)}$ for iteration $\ell + 1$ for all vertices i. This is just the simplest update scheme; more refined schemes might for example update the positions by $(\mathbf{v}_i^{(\ell)} + \mathbf{v}_i^{(\ell+1)})/2 \cdot \Delta t$.

[1]in the original: Ablenkung jedes Punktes von seiner freien Bewegung

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

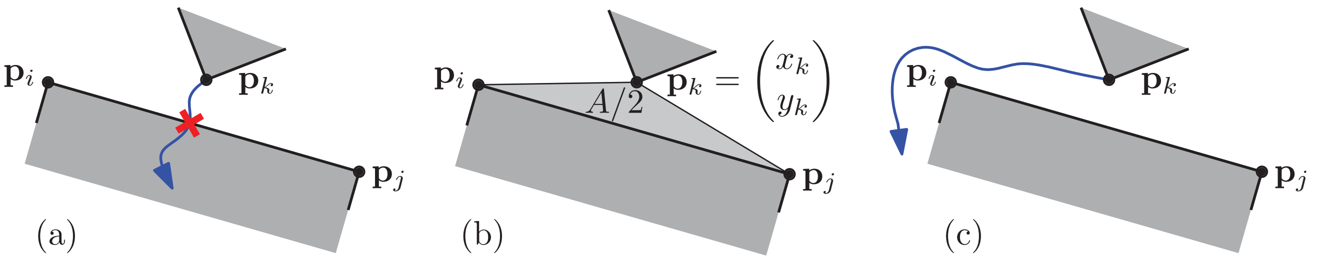

Figure 5: Avoiding overlap between the brick containing $\mathbf{p}_k$ and the brick containing the edge $\mathbf{p}_i\mathbf{p}_j$

The constraints (L2) and (S2) are formulated in terms of the position functions $\mathbf{p}_i(t)$ and their derivatives. In the context where we consider the motion in discrete time steps of length Δt, we have to formulate these constraints in terms of velocities $\mathbf{v}_i$ and accelerations $\mathbf{a}_i$, which are now distinct quantities at every time step. (They only *approximate* the derivatives of the positions.) In the next section we will work out the resulting constraints. The precise form of these constraints is not so interesting per se, but it is necessary to state them in order to give meaning to the theorem about the algorithm that we will state below.

a) Determine the active sidedness constraints.

b) Solve the quadratic optimization problem (Q) in the variables $(\mathbf{a}_i)$, subject to second-order length and sidedness constraints (L2) and (S2).

c) Advance velocities and positions:

$$\mathbf{v}_i^{(\ell+1)} := \mathbf{v}_i^{(\ell)} + \mathbf{a}_i \cdot \Delta t, \qquad (3.1)$$
$$\mathbf{p}_i^{(\ell+1)} := \mathbf{p}_i^{(\ell)} + \mathbf{v}_i^{(\ell+1)} \cdot \Delta t,$$

where Δt is the time increment between successive steps.

Figure 6: The classical acceleration-based loop of rigid-body simulation

4 The Explicit Form of the Linear Constraints

The derivatives of the squared length $\|\mathbf{p}_i(t) - \mathbf{p}_j(t)\|^2 = \langle \mathbf{p}_i(t) - \mathbf{p}_j(t), \mathbf{p}_i(t) - \mathbf{p}_j(t)\rangle$ are expressed as follows:

$$\begin{aligned} \tfrac{\mathrm{d}}{\mathrm{d}t}\|\mathbf{p}_i - \mathbf{p}_j\|^2 &= \tfrac{\mathrm{d}}{\mathrm{d}t}\langle \mathbf{p}_i - \mathbf{p}_j, \mathbf{p}_i - \mathbf{p}_j\rangle \\ &= 2\langle \mathbf{v}_i - \mathbf{v}_j, \mathbf{p}_i - \mathbf{p}_j\rangle \\ \tfrac{\mathrm{d}^2}{\mathrm{d}t^2}\|\mathbf{p}_i - \mathbf{p}_j\|^2 &= 2\langle \mathbf{a}_i - \mathbf{a}_j, \mathbf{p}_i - \mathbf{p}_j\rangle \\ &\quad + 2\langle \mathbf{v}_i - \mathbf{v}_j, \mathbf{v}_i - \mathbf{v}_j\rangle \end{aligned} \qquad (4.2)$$

For the sidedness constraints, taking derivatives row-wise, we get

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} A(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k) &= \frac{\mathrm{d}}{\mathrm{d}t} \begin{vmatrix} 1 & 1 & 1 \\ x_i & x_j & x_k \\ y_i & y_j & y_k \end{vmatrix} \\ &= \begin{vmatrix} 1 & 1 & 1 \\ \dot{x}_i & \dot{x}_j & \dot{x}_k \\ y_i & y_j & y_k \end{vmatrix} + \begin{vmatrix} 1 & 1 & 1 \\ x_i & x_j & x_k \\ \dot{y}_i & \dot{y}_j & \dot{y}_k \end{vmatrix} \end{aligned}$$

For the second derivative, we obtain

$$\begin{aligned} \frac{\mathrm{d}^2}{\mathrm{d}t^2} A(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k) &= \begin{vmatrix} 1 & 1 & 1 \\ \ddot{x}_i & \ddot{x}_j & \ddot{x}_k \\ y_i & y_j & y_k \end{vmatrix} \\ &\quad + 2 \begin{vmatrix} 1 & 1 & 1 \\ \dot{x}_i & \dot{x}_j & \dot{x}_k \\ \dot{y}_i & \dot{y}_j & \dot{y}_k \end{vmatrix} + \begin{vmatrix} 1 & 1 & 1 \\ x_i & x_j & x_k \\ \ddot{y}_i & \ddot{y}_j & \ddot{y}_k \end{vmatrix} \end{aligned}$$

We will now make use of the assumption that the three points $\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k$ are initially aligned whenever we wish to test the derivatives of the sidedness predicate. By translation and rotation, we assume without loss of generality that $\mathbf{p}_i = \binom{0}{0}$, $\mathbf{p}_j = \binom{\ell}{0}$, and $\mathbf{p}_k = \binom{\alpha\ell}{0}$ for some $0 \le \alpha \le 1$, where $\ell = \|\mathbf{p}_i - \mathbf{p}_j\|$. Then we have $\mathbf{p}_k = \alpha\mathbf{p}_j + (1-\alpha)\mathbf{p}_i$, and the formulas simplify:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} A(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k) &= \begin{vmatrix} 1 & 1 & 1 \\ 0 & \ell & \ell\alpha \\ \dot{y}_i & \dot{y}_j & \dot{y}_k \end{vmatrix} \\ &= (\dot{y}_k - [\alpha\dot{y}_j + (1-\alpha)\dot{y}_i]) \cdot \ell \\ \frac{\mathrm{d}^2}{\mathrm{d}t^2} A(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k) &= 2 \begin{vmatrix} 1 & 1 & 1 \\ \dot{x}_i & \dot{x}_j & \dot{x}_k \\ \dot{y}_i & \dot{y}_j & \dot{y}_k \end{vmatrix} + \begin{vmatrix} 1 & 1 & 1 \\ 0 & \ell & \alpha\ell \\ \ddot{y}_i & \ddot{y}_j & \ddot{y}_k \end{vmatrix} \\ &= 2 \cdot A(\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k) \\ &\qquad + \big(\ddot{y}_k - [\alpha\ddot{y}_j + (1-\alpha)\ddot{y}_i]\big) \cdot \ell \end{aligned}$$

The expressions $\dot{y}_i$, $\ddot{y}_i$, etc., are proportional to the projections of the vectors $\mathbf{v}_i$, $\mathbf{a}_i$ on the perpendicular direction $\mathbf{n} = (\mathbf{p}_j - \mathbf{p}_i)^\perp = \binom{0}{\ell}$ to the bar $\mathbf{p}_i\mathbf{p}_j$. For example, $\langle \mathbf{v}_i, \mathbf{n}\rangle = \ell\dot{y}_i$ and $\langle \mathbf{a}_i, \mathbf{n}\rangle = \ell\ddot{y}_i$. We can thus

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

write the derivatives of $A(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k)$ in a coordinate-free form:

$$(4.3)\quad \frac{\mathrm{d}}{\mathrm{d}t}A(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k) = \big\langle \mathbf{v}_k - [\alpha\mathbf{v}_j + (1-\alpha)\mathbf{v}_i], (\mathbf{p}_j - \mathbf{p}_i)^{\perp}\big\rangle$$

$$(4.4)\quad \frac{\mathrm{d}^2}{\mathrm{d}t^2}A(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k) = 2\cdot A(\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k) + \big\langle \mathbf{a}_k - [\alpha\mathbf{a}_j + (1-\alpha)\mathbf{a}_i], (\mathbf{p}_j - \mathbf{p}_i)^{\perp}\big\rangle$$

In the discretized setting, the second derivatives in the constraints (L2) and (S2) must be interpreted as the right-hand sides of (4.2) and (4.4), respectively. In determining for which triples (i, j, k) the constraint (S2) is imposed, according to condition (S2), (4.3) must be used for the first derivatives $\frac{\mathrm{d}}{\mathrm{d}t}A(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k)$. Summarizing, we have the following discretized versions of the constraints:

$$(\mathrm{L1}^+)\qquad \langle \mathbf{v}_i - \mathbf{v}_j, \mathbf{p}_i - \mathbf{p}_j\rangle = 0, \text{ for all bars } (i, j),$$

$$(\mathrm{L2}^+)\quad \langle \mathbf{a}_i - \mathbf{a}_j, \mathbf{p}_i - \mathbf{p}_j\rangle + \langle \mathbf{v}_i - \mathbf{v}_j, \mathbf{v}_i - \mathbf{v}_j\rangle = 0, \text{ for all bars } (i, j),$$

and

$(\mathrm{S2}^+)$

$$2\cdot A(\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k) \geq -\big\langle \mathbf{a}_k - [\alpha\mathbf{a}_j + (1-\alpha)\mathbf{a}_i], (\mathbf{p}_j - \mathbf{p}_i)^{\perp}\big\rangle,$$

for all triples (i, j, k) consisting of an edge $\mathbf{p}_i\mathbf{p}_j$ and a vertex $\mathbf{p}_k$ with

$\mathbf{p}_k = \alpha\mathbf{p}_j + (1-\alpha)\mathbf{p}_i$, for some $0 \leq \alpha \leq 1$, and

$\big\langle \mathbf{v}_k - [\alpha\mathbf{v}_j + (1-\alpha)\mathbf{v}_i], (\mathbf{p}_j - \mathbf{p}_i)^{\perp}\big\rangle = 0.$

5 Elimination of Acceleration and Velocity

While the length constraints (L2) are always there, the sidedness constraints (S2) are required only for certain triples. The conditions under which the sidedness constraint (S2) is imposed are equality conditions. This leads to problems in practice when testing these conditions. Imagine a brick B_1 whose edge $\mathbf{p}_i\mathbf{p}_j$ slides along the corner $\mathbf{p}_k$ of another brick B_2, keeping constant contact. Since the motion described by accelerations and velocities over discrete time steps is only a second-order approximation of the true motion, it is unavoidable that the point $\mathbf{p}_k$ will slightly drift away from the brick B_1, or worse, creep into the brick B_1. If the point $\mathbf{p}_k$ loses contact with B_1 and the distance becomes so large that the corresponding non-penetration constraint (S2) is no longer enforced, it means that it will "fall into" B_1 in the next step, like in a collision. It is thus necessary to constantly monitor the length and non-penetration constraints and correct the positions $\mathbf{p}_i$ and velocities $\mathbf{v}_i$ (Step (d) of Figure 3).

The position-based procedure of Figure 7, which is our main contribution, eliminates the need to perform these corrections. It starts by estimating the velocities from the positions at the two previous iterations $\ell - 1$ and ℓ and computes new positions $\mathbf{p}_i^{(\ell+1)}$ for iteration $\ell + 1$.

a) Linear extrapolation: $\bar{\mathbf{p}}_i := \mathbf{p}_i^{(\ell)} + [\mathbf{p}_i^{(\ell)} - \mathbf{p}_i^{(\ell-1)}]$

b) Apply external forces: $\tilde{\mathbf{p}}_i := \bar{\mathbf{p}}_i + \mathbf{g}\cdot(\Delta t)^2$, where Δt is the time increment between successive steps.

c) Solve the quadratic optimization problem in the variables $(\mathbf{p}_i)$.

$$(\mathrm{Q}')\qquad \text{minimize} \sum_{i=1}^{n} m_i \cdot \|\mathbf{p}_i - \tilde{\mathbf{p}}_i\|^2$$

subject to adapted length and sidedness constraints (L′) and (S′).

d) Use the solution $\mathbf{p}_i$ as positions $\mathbf{p}_i^{(\ell+1)}$ for the next iteration.

Figure 7: The purely position-based simulation iteration

The new length constraints that correspond to (L2) are

$$(\mathrm{L}')\quad \langle \mathbf{p}_i - \mathbf{p}_j, \mathbf{p}_i^{(\ell)} - \mathbf{p}_j^{(\ell)}\rangle = \|\mathbf{p}_i^{(\ell)} - \mathbf{p}_j^{(\ell)}\|^2 - \|(\mathbf{p}_i^{(\ell)} - \mathbf{p}_i^{(\ell-1)}) - (\mathbf{p}_j^{(\ell)} - \mathbf{p}_j^{(\ell-1)})\|^2 \quad \text{for all bars } (i, j).$$

If we look only at the first term on the right-hand side, this condition says that the *projection* of the new segment $\mathbf{p}_i\mathbf{p}_j$ on the line through the old segment $\mathbf{p}_i^{(\ell)}\mathbf{p}_j^{(\ell)}$ is equal to the previous length of this segment. $\mathbf{p}_i^{(\ell)}\mathbf{p}_j^{(\ell)}$. Since in one step, we assume the segment is rotated only slightly, this projected length is a good (first-order) substitute for the true length of $\mathbf{p}_i\mathbf{p}_j$. The last term offsets the small effects of the rotation. The precise form of this term results from the computation in Theorem 5.1 below.

The new sidedness constraints that correspond to

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

(S2) are

$$(\mathrm{S}') \quad \langle \mathbf{p}_k - (\alpha \mathbf{p}_i + (1-\alpha)\mathbf{p}_j), (\mathbf{p}_j^{(\ell)} - \mathbf{p}_i^{(\ell)})^\perp \rangle \geq \frac{-2 \cdot A(\mathbf{p}_i^{(\ell)} - \mathbf{p}_i^{(\ell-1)}, \mathbf{p}_j^{(\ell)} - \mathbf{p}_j^{(\ell-1)}, \mathbf{p}_k^{(\ell)} - \mathbf{p}_k^{(\ell-1)})}{\Delta t}, \quad \text{for certain triples } (i,j,k),$$

where $(\mathbf{p}_j^{(\ell)} - \mathbf{p}_i^{(\ell)})^\perp$ denotes the vector $\mathbf{p}_j^{(\ell)} - \mathbf{p}_i^{(\ell)}$ rotated by 90 degrees, and the scalar $\alpha = \alpha_{ijk}$ is defined by the condition that the projection of $\mathbf{p}_k$ on the segment $\mathbf{p}_i^{(\ell)}\mathbf{p}_j^{(\ell)}$ divides this segment in the ratio $(1-\alpha) : \alpha$, that is,

$$\langle \mathbf{p}_k^{(\ell)}, \mathbf{p}_j^{(\ell)} - \mathbf{p}_i^{(\ell)} \rangle = \langle \alpha \mathbf{p}_i^{(\ell)} + (1-\alpha)\mathbf{p}_j^{(\ell)}, \mathbf{p}_j^{(\ell)} - \mathbf{p}_i^{(\ell)} \rangle.$$

We give a geometric interpretation for this condition for the case when the segment $\mathbf{p}_i\mathbf{p}_j$ (initially) does not rotate. In this case, the right-hand side is zero. Nonnegativity of the left-hand term says that that the point $\mathbf{p}_k$ remains on the correct side of the edge $\mathbf{p}_i\mathbf{p}_j$ if we just consider the component of the "motion" $\mathbf{p}_i - \mathbf{p}_i^{(\ell)}$, $\mathbf{p}_j - \mathbf{p}_j^{(\ell)}$, $\mathbf{p}_k - \mathbf{p}_k^{(\ell)}$ *perpendicular* to the direction of the edge $\mathbf{p}_i^{(\ell)}\mathbf{p}_j^{(\ell)}$, see Figure 8. This is a first-order approximation of the true sidedness condition for $\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k$. The formula on the right-hand side gives a correction term for the case when the segment rotates.

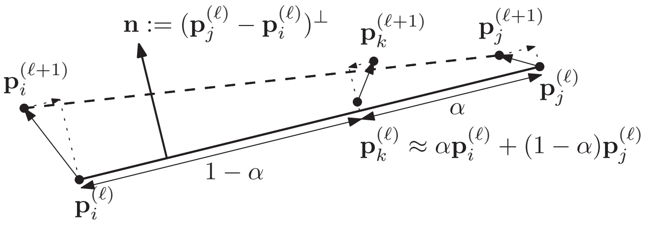
$\mathbf{n} := (\mathbf{p}_j^{(\ell)} - \mathbf{p}_i^{(\ell)})^\perp$
$\mathbf{p}_k^{(\ell+1)}$
$\mathbf{p}_j^{(\ell+1)}$
$\mathbf{p}_i^{(\ell+1)}$
$\mathbf{p}_j^{(\ell)}$
α
$\mathbf{p}_k^{(\ell)} \approx \alpha \mathbf{p}_i^{(\ell)} + (1-\alpha)\mathbf{p}_j^{(\ell)}$
$1-\alpha$
$\mathbf{p}_i^{(\ell)}$

Figure 8: Geometric interpretation of the linearized sidedness constraints

The relation between this model and the original acceleration-based approach of Figure 6 is expressed in the following theorem:

THEOREM 5.1. *Assume that the current velocities* $\mathbf{v}_i^{(\ell)}$ *satisfy the first-order length constraints* (L1$^+$)*, and the active constraints in* (S2$^+$) *are determined by evaluating the conditions of* (S2$^+$) *exactly. Then the acceleration-based algorithm of Figure 6 yields the same positions* $\mathbf{p}_i^{(\ell+1)}$ *as the optimization of* (Q$'$) *subject to constraints* (L$'$) *and the corresponding inequalities* (S$'$).

The proof of this theorem is a straightforward substitution. It is given in the next section.

The advantage of the new approach using (Q$'$) is that it is less sensitive to the selection of active sidedness constraints. If the point $\mathbf{p}_k$ is currently very far from the line through the edge $\mathbf{p}_i\mathbf{p}_j$, but on the correct side, then the corresponding condition (S$'$) is fulfilled by a wide margin, and is therefore effectively inactive. It is therefore safe to include this constraint in the model. As $\mathbf{p}_k$ approaches the segment $\mathbf{p}_i\mathbf{p}_j$, we need not worry when it has actually reached it. The constraint (S$'$) will be automatically active whenever it is necessary.

When $\mathbf{p}_k$ "goes around" the edge $\mathbf{p}_i\mathbf{p}_j$ and crosses the line through $\mathbf{p}_i\mathbf{p}_j$ far from the edge (Figure 5c), we must deactivate (S$'$), but this decision is a "macroscopic" test and not based on a precise comparison with 0. Care must be taken when $\mathbf{p}_k$ approaches one of the vertices $\mathbf{p}_i$ or $\mathbf{p}_j$; then the activation of the constraint must be coordinated with that of the adjacent edge, so that $\mathbf{p}_k$ does not accidentally slip through the corner of a brick. Section 8.5 contains a method that works correctly for convex obstacles.

6 Proof of Theorem 5.1

In the acceleration-based algorithm of Figure 6, by definition, we have $\mathbf{p}_i^{(\ell)} - \mathbf{p}_i^{(\ell-1)} = \mathbf{v}_i^{(\ell)}\Delta t$, and thus $\bar{\mathbf{p}}_i = \mathbf{p}_i^{(\ell)} + \mathbf{v}_i^{(\ell)}\Delta t$. By (3.1), the new position is obtained as follows.

$$\begin{aligned}
\mathbf{p}_i^{(\ell+1)} &:= \mathbf{p}_i^{(\ell)} + \mathbf{v}_i^{(\ell+1)} \cdot \Delta t \\
&= \mathbf{p}_i^{(\ell)} + (\mathbf{v}_i^{(\ell)} + \mathbf{a}_i \cdot \Delta t) \cdot \Delta t \\
(6.5) \qquad &= \mathbf{p}_i^{(\ell)} + \mathbf{v}_i^{(\ell)} \cdot \Delta t + \mathbf{a}_i \cdot (\Delta t)^2 \\
&= \bar{\mathbf{p}}_i + \mathbf{a}_i \cdot (\Delta t)^2 \\
(6.6) \qquad &= \tilde{\mathbf{p}}_i + (\mathbf{a}_i - \mathbf{g}) \cdot (\Delta t)^2
\end{aligned}$$

From the last expression we get

$$\mathbf{p}_i^{(\ell+1)} - \tilde{\mathbf{p}}_i = (\mathbf{a}_i - \mathbf{g}) \cdot (\Delta t)^2$$

It is now obvious that the objective functions (Q) and (Q$'$) of the two models are equivalent since they are the weighted sums of the squared lengths of these vectors: the objective function (Q$'$) is expressed in term of the the unknowns $\mathbf{p}_i = \mathbf{p}_i^{(\ell+1)}$ on the left-hand side, whereas the objective function (Q) is expressed in term of the the unknowns $\mathbf{a}_i$ on the right-hand side.

We shall now prove equivalence of the length constraints (L2$^+$) and (L$'$) for the two models. By (6.5), we have

$$\begin{aligned}
\langle \mathbf{p}_i^{(\ell+1)} - \mathbf{p}_j^{(\ell+1)}, \mathbf{p}_i^{(\ell)} - \mathbf{p}_j^{(\ell)} \rangle &= \langle \mathbf{p}_i^{(\ell)} - \mathbf{p}_j^{(\ell)}, \mathbf{p}_i^{(\ell)} - \mathbf{p}_j^{(\ell)} \rangle \\
&\quad + \langle \mathbf{v}_i^{(\ell)} - \mathbf{v}_j^{(\ell)}, \mathbf{p}_i^{(\ell)} - \mathbf{p}_j^{(\ell)} \rangle \cdot \Delta t \\
&\quad + \langle \mathbf{a}_i - \mathbf{a}_j, \mathbf{p}_i^{(\ell)} - \mathbf{p}_j^{(\ell)} \rangle \cdot (\Delta t)^2
\end{aligned}$$

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

By assumption, the velocities fulfill (L1$^+$), and therefore, the second term on the right-hand side is 0. By rearranging and adding the term $\|\Delta t \cdot (\mathbf{v}_i^{(\ell)} - \mathbf{v}_j^{(\ell)})\|^2 = \|(\mathbf{p}_i^{(\ell)} - \mathbf{p}_i^{(\ell-1)}) - (\mathbf{p}_j^{(\ell)} - \mathbf{p}_j^{(\ell-1)})\|^2$ to both sides, we obtain

$$\begin{aligned}\langle \mathbf{p}_i^{(\ell+1)} - \mathbf{p}_j^{(\ell+1)}, \mathbf{p}_i^{(\ell)} - \mathbf{p}_j^{(\ell)}\rangle - \|\mathbf{p}_i^{(\ell)} - \mathbf{p}_j^{(\ell)}\|^2 \\ + \|(\mathbf{p}_i^{(\ell)} - \mathbf{p}_i^{(\ell-1)}) - (\mathbf{p}_j^{(\ell)} - \mathbf{p}_j^{(\ell-1)})\|^2 \\ = (\langle \mathbf{a}_i - \mathbf{a}_j, \mathbf{p}_i^{(\ell)} - \mathbf{p}_j^{(\ell)}\rangle + \|(\mathbf{v}_i^{(\ell)} - \mathbf{v}_j^{(\ell)})\|^2) \cdot (\Delta t)^2\end{aligned}$$

The new length constraint (L$'$) requires that the left-hand side is zero, whereas the original length constraint (L2$^+$) requires that the right-hand side is zero.

Finally, we shall prove equivalence between the sidedness constraints (S2$^+$) and (S$'$). By (6.5), we have

$$\begin{aligned}&\mathbf{p}_k^{(\ell+1)} - \Big[(1-\alpha)\mathbf{p}_i^{(\ell+1)} + \alpha \mathbf{p}_j^{(\ell+1)}\Big] \\ &= \mathbf{p}_k^{(\ell)} + \mathbf{v}_k^{(\ell)} \cdot \Delta t + \mathbf{a}_k \cdot \Delta t^2 \\ &\quad - \Big[(1-\alpha)[\mathbf{p}_i^{(\ell)} + \mathbf{v}_i^{(\ell)} \cdot \Delta t + \mathbf{a}_i \cdot \Delta t^2] \\ &\quad + \alpha[\mathbf{p}_j^{(\ell)} + \mathbf{v}_j^{(\ell)} \cdot \Delta t + \mathbf{a}_j \cdot \Delta t^2]\Big] \\ &= \Big(\mathbf{p}_k^{(\ell)} - \Big[(1-\alpha)\mathbf{p}_i^{(\ell)} + \alpha \mathbf{p}_j^{(\ell)}\Big]\Big) \\ &\quad + \Big(\mathbf{v}_k^{(\ell)} - \Big[(1-\alpha)\mathbf{v}_i^{(\ell)} + \alpha \mathbf{v}_j^{(\ell)}\Big]\Big) \cdot \Delta t \\ &\quad + \Big(\mathbf{a}_k - [(1-\alpha)\mathbf{a}_i + \alpha \mathbf{a}_j]\Big) \cdot (\Delta t)^2\end{aligned}$$

Since $\mathbf{p}_k^{(\ell)} = (1-\alpha)\mathbf{p}_i^{(\ell)} + \alpha \mathbf{p}_j^{(\ell)}$, the first term on the right-hand side is zero. Let us now take the scalar product with the vector $\mathbf{n} = (\mathbf{p}_j - \mathbf{p}_i)^\perp$ perpendicular to the bar $\mathbf{p}_i\mathbf{p}_j$. By assumption, the velocities $\mathbf{v}_i^{(\ell)}$ fulfill (L1$^+$), and this means that the second term vanishes. Thus we obtain

$$\begin{aligned}\Big\langle \mathbf{p}_k^{(\ell+1)} - \Big[(1-\alpha)\mathbf{p}_i^{(\ell+1)} + \alpha \mathbf{p}_j^{(\ell+1)}\Big], \mathbf{n}\Big\rangle = \\ \Big\langle \mathbf{a}_k - [(1-\alpha)\mathbf{a}_i + \alpha \mathbf{a}_j], \mathbf{n}\Big\rangle \cdot (\Delta t)^2.\end{aligned}$$

This equation together with the identity

$$\begin{aligned}A(\mathbf{p}_i^{(\ell)} - \mathbf{p}_i^{(\ell-1)}, \mathbf{p}_j^{(\ell)} - \mathbf{p}_j^{(\ell-1)}, \mathbf{p}_k^{(\ell)} - \mathbf{p}_k^{(\ell-1)}) \\ = A(\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k) \cdot (\Delta t)^3\end{aligned}$$

transforms (S2$^+$) into (S$'$).

7 Collisions

Collisions of rigid bodies incur a *discontinuous* change of velocities and thus are not captured by Gauß' principle.

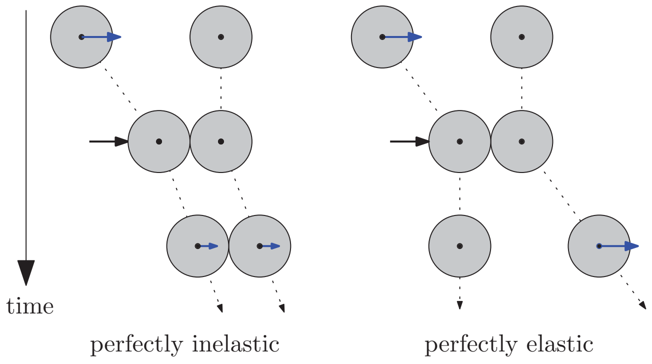

Figure 9: Inelastic and elastic collisions

The behavior at a collision is governed by the *coefficient of restitution* c, which determines how much of the kinetic energy resulting from the collision is conserved as kinetic energy. Figure 9 shows two equal colliding balls and the two extremes of perfectly inelastic collisions ($c = 0$, the relative speed is reduced to zero) and perfectly elastic collisions ($c = 1$). Like in every closed system, the total momentum is always conserved.

The standard way to handle collisions (after detecting them) is to "stop" the motion, compute new velocities according to some model of collision, and continue the motion from there. A poorly understood phenomenon, however, is the handling of multiple collisions, which involve not just the two colliding objects but other objects that are directly or indirectly in touch with them. Chatterjee and Ruina [5] speak of "the ill-posed nature of simultaneous multiple collisions".

Interestingly, we found that, for the case of totally inelastic collisions, our position-based algorithm of Figure 7 seems to automatically model collisions correctly without any special treatment. What we can prove about our algorithm is formally expressed in the following theorem, which follows by a straightforward computation.

THEOREM 7.1. *Assume that there are no external forces:* $\mathbf{g} = 0$. *Let* $\mathbf{v}_i^{\text{old}} := (\mathbf{p}_i^{(\ell)} - \mathbf{p}_i^{(\ell-1)})/\Delta t$ *and let* $\mathbf{v}_i^{\text{new}} := (\mathbf{p}_i^{(\ell+1)} - \mathbf{p}_i^{(\ell)})/\Delta t$, *where* $\mathbf{p}_i^{(\ell+1)}$ *has been computed by the position-based algorithm of Figure 7. Then the "new velocities"* $\mathbf{v}_i^{\text{new}}$ *are equal to the solutions of the following quadratic optimization problem in the variables* $\mathbf{v}_i$:

$$(\mathrm{Q}^*) \qquad \textit{minimize} \sum_{i=1}^{n} m_i \cdot \|\mathbf{v}_i - \mathbf{v}_i^{\text{old}}\|^2$$

subject subject to length constraints (L1) *and sidedness constraints* (S1) *on the variables* $\mathbf{v}_i$.

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

We have in mind the following application to collisions: suppose a collision occurs at some time step ℓ. Let $\Delta t \to 0$, (but let ℓ continue to refer to the time of collision). Then the quantities $\mathbf{v}_i^{\text{old}} := (\mathbf{p}_i^{(\ell)} - \mathbf{p}_i^{(\ell-1)})/\Delta t$ and $\mathbf{v}_i^{\text{new}} := (\mathbf{p}_i^{(\ell+1)} - \mathbf{p}_i^{(\ell)})/\Delta t$ converge to the velocities before and after the collision. It follows from the above theorem that, as $\Delta t \to 0$, the new velocities $\mathbf{v}_i^{\text{new}}$ converge to the solution of (Q*). This is true even in the presence of an external force $\mathbf{g}$, since it enters the calculation with a quadratic factor $(\Delta t)^2$. Thus the relative influence of $\mathbf{g}$ goes to zero when $\Delta t \to 0$.

We conjecture that the optimization problem (Q*) correctly models the velocity change for perfectly inelastic collision among an arbitrary set of mutually touching rigid bodies. We have checked that the result coincides with the standard model of inelastic collision for the case of two isolated colliding bodies. For the general case, we have so far not been able to find this optimization problem, whose form is analogous to Gauß' principle, in the physics literature, nor have we been able to derive it from other physical principles.

The difficulty in applying Theorem 7.1 is that, when Δt is a fixed positive constant, the effect of the collision is not expressed in pure form, but it is mixed with the smooth velocity change according to Gauß' principle. But this is even a potential advantage of the model: if we are just interested in inelastic collisions, we do not have to stop for collisions or even notice them in the simulation.

8 Implementation

8.1 Continuous Mass Distributions. So far, we have described the method for point masses. Continuous mass distributions μ can be replaced by appropriately chosen discrete mass distributions which behave identically with respect to first and second moments. For example, a rectangular brick with homogeneous mass distribution μ can be simulated by a rigid structure of 9 point masses. The mass distribution in Figure 10 ensures that in terms of momentum of inertia and rotational momentum, the discrete structure is indistinguishable from the homogeneous rectangle. One has to check that

$$\iint f(x,y)\,\mathrm{d}\mu(x,y) = \sum_{i=1}^{9} m_i f(x_i, y_i)$$

holds for all polynomial functions f of degree at most 2. It is sufficient to check this for the six functions $f = 1$, x, y, x^2, xy, and y^2.

8.2 Reducing the Number of Variables. In practice, to speed up the calculations, one would traditionally model a rectangle or any other rigid plane shape directly with three variables for translation and rotation (six variables for three dimensions), instead of 18 variables with 16 equations, one of which is redundant. The constraints would have to be rewritten in terms of the new variables. Our simple point-based model, on the other hand, is more uniform, and it extends in a straightforward way to articulated objects whose rigid parts are connected by joints. In terms of running time for the optimization problem, it is not a priori clear that a model with more variables and equations but a sparse structure would be outperformed by a smaller but denser system that arises after elimination of variables.

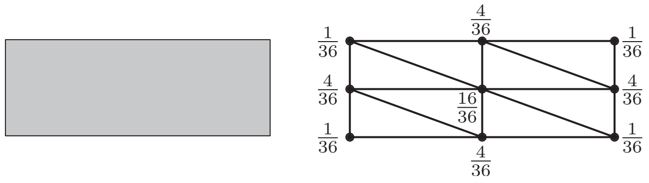

Figure 10: A homogeneous rectangle can be replaced by a rigid framework of 9 discrete points. (One bar is redundant.)

8.3 Maintaining Lengths. In the program, we have actually used another modification of the length constraint:

(L$''$)
$$\langle \mathbf{p}_i - \mathbf{p}_j, \mathbf{p}_i^{(\ell)} - \mathbf{p}_j^{(\ell)} \rangle = \|\mathbf{p}_i^{(0)} - \mathbf{p}_j^{(0)}\| \cdot \|\mathbf{p}_i^{(\ell)} - \mathbf{p}_j^{(\ell)}\| \qquad \text{for all bars } (i,j).$$

Geometrically, this condition says that the projection of the new segment $\mathbf{p}_i\mathbf{p}_j$ should be equal to the *original* length of the segment, instead of the previous length of the segment $\mathbf{p}_i^{(\ell)}\mathbf{p}_j^{(\ell)}$. This is obviously better than the constraint (L$'$) that is used in the theorem, since it automatically corrects changes in length due to discretization errors instead of accumulating them. This improvement incorporates thus elements of Step (d) of Figure 3. In the limit, as $\Delta t \to 0$, and without numerical errors, the length of each bar would be constant and there is no difference to the original constraint (L$'$).

We have implemented our algorithm in a simple prototype system. We have provided no special collision handling mechanism and thus our collision model defaults to totally inelastic collision of Section 7: two bricks that hit each other will stick together (for some short time, at least). For the quadratic optimization problems, we used the commercial optimization pack-

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

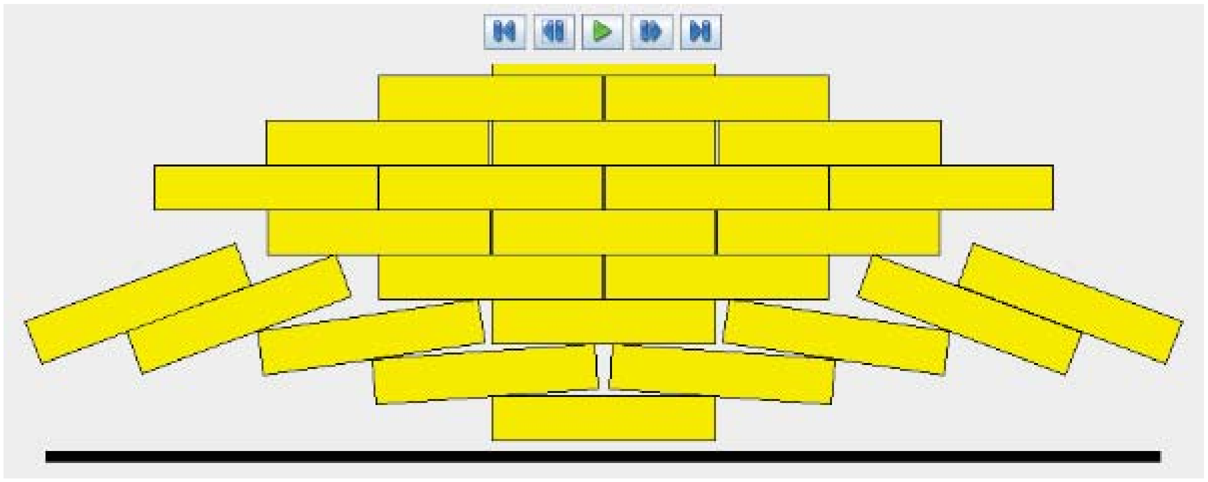

Figure 11: 5×5 bricks that were originally positioned as in Fig. 2.

age Cplex.[2] Figure 11 shows a snapshot of an animation that was produced by the program.[3] A simulation with 1000 time steps takes about 5 minutes on a moderate workstation. In the initial phases, each quadratic programming problem takes about 1/3 second to solve. Later, when the tower is dispersed into separate pieces, this time is reduced to a small fraction of a second.

8.4 Equality Constraints for Redundant Bars.

Equality constraints pose a problem for structures with redundant bars such as the "overbraced" framework of Figure 10. While the exact length constraints for these bars are consistent, numerical errors will lead to an overdetermined system of equations that is inconsistent. This problem equally affects the classical acceleration-based approach of Figure 6 and our new position-based algorithm. One possibility to avoid this problem is to remove redundant bars, such that each brick is represented as a *minimally rigid* framework. Another option is to remove the length constraints and turn them into a penalty term in the objective function, by adding the term

$$M \cdot \sum_{\text{all bars } (i,j)} \Big[\langle \mathbf{p}_i - \mathbf{p}_j, \mathbf{p}_i^{(\ell)} - \mathbf{p}_j^{(\ell)} \rangle - \|\mathbf{p}_i^{(0)} - \mathbf{p}_j^{(0)}\| \cdot \|\mathbf{p}_i^{(\ell)} - \mathbf{p}_j^{(\ell)}\|\Big]^2$$

with a large penalty factor M. In our simulations we have taken $M = 100$. This approach is no longer directly related to Gauß' principle, but it may have the advantage of avoiding numerical instabilities in degenerate situations like when a chain of bars becomes aligned.

8.5 Determining the Active Sidedness Constraints.

To find the sidedness constraints that are active according to (S2$^+$), one has to check which of the pairs of bricks that were in contact in the previous iteration remains in contact, and one has to detect new contacts. The first task is straightforward, but the second task effectively amounts to *collision detection* among moving objects. Detecting collisions efficiently is a research direction of its own, and there are many different algorithms in the area of computational geometry and in robotics.

In our implementation, we have not made an effort to make the collision test efficient. For each pair of bricks, we have selected one *critical* separating line that goes through an edge $\mathbf{p}_i\mathbf{p}_j$ of one of the bricks. We have then included a sidedness constraint between this edge and all vertices $\mathbf{p}_k$ of the other brick. The separating line for a pair of bricks B_1 and B_2 is chosen as follows. First, we look at the line through each edge of B_1 and compute the sidedness constraint that is

[2] `http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/`

[3] The complete animation can be viewed at `http://page.mi.fu-berlin.de/rote/Papers/slides/collapse/Animation/applet/5x5-bricks.html`. In this simulation, each rectangle had four equal point masses at the corners. The astute reader may wonder about the asymmetry towards the end of the animation, despite the symmetric starting configuration shown in Fig. 2. Our version of Cplex could solve quadratic optimization problems only by an interior-point ("barrier") method that stops when a sufficiently close approximation of the optimum is reached. The initial errors caused by this approximation were amplified during the process and eventually produced a gross asymmetry. We have checked that this asymmetry is not due to the asymmetric insertion of one diagonal as in Figure 4: the asymmetry persists when we insert both diagonals in each rectangle. The horizontal gaps in the upper part of Figure 11 are caused by the same phenomenon.

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

"most satisfied" or "least violated" by B_2, i.e., has the largest signed distance separating it from B_2. We then exchange the roles of B_1 and B_2 and compute the "best separating" edge of B_2 in a similar way. From the two edges thus selected, we now pick the "least satisfied" (or "most violated") one as the critical edge that is used for the sidedness constraints.

This procedure works for all convex polygons, and it will safely prevent the bricks from overlapping too much, even if, due to numeric errors or discretization errors, a small penetration occurs.

9 Conclusion

Our contribution is to show how the motion of an arbitrary number of non-overlapping rigid bodies can be calculated by Gauß' principle. The novelty lies in integrating the computation of acceleration, velocity, and position into one quadratic programming problem. At the same time we integrate also non-elastic collisions in the same model.

Our purely position-based approach to rigid-body simulation may not be entirely new. In fact, when one reads Gauß' original paper [11] one gets the impression that he may have had such a formulation in mind, since he explicitly talks about new *positions* of point masses. However, these are positions after *infinitely* small time steps, and the modification of positions is associated with velocities and forces.

The ad-hoc optimization problems of Milenkovic and Schmidl [13] are also based purely on new positions, and the optimization step merely corrects penetrations that have occurred.

Note that the situation on the right part of Figure 2 represents a different situation: it is the *static* problem of finding the state with minimum potential energy. If we abstract from the vertical thickness of the bricks and assume that they cannot slide away sideways, this problem can be solved as a linear programming problem.

Future work includes the use of a reliable quadratic programming solver and verifier[4], making use of the coherence of data between successive problems, elimination of unnecessary constraints (which is related to efficient collision detection), and an investigation of the piecewise smoothness of the motion.

It is perhaps interesting in this context that the solution of the Carpenter's Rule Problem—unfolding a given polygonal chain without self-overlap—by Connelly, Demaine and Rote originally also used a quadratic objective function to define the global motion, as described in the proceedings version [6]. However, the solution of a (strictly convex) quadratic optimization problem does not necessarily depend smoothly on the data, and therefore it was finally modified by a logarithmic barrier function [7].

[4]see `http://www.zib.de/Optimization/Projects/MIP/ExactIP/ExactIPlong.en.html`, `http://scip.zib.de/`

References

[1] D. Baraff. Analytical methods for dynamic simulation of non-penetrating rigid bodies. *SIGGRAPH Comput. Graph.*, 23(3):223–232, 1989. doi:10.1145/74334.74356.

[2] D. Baraff. Fast contact force computation for nonpenetrating rigid bodies. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 23–34, New York, NY, USA, 1994. ACM. doi:10.1145/192161.192168.

[3] J. Bender and A. Schmitt. Fast dynamic simulation of multi-body systems using impulses. In *Virtual Reality Interactions and Physical Simulations (VRIPhys), Madrid, November 6–7*, 2006.

[4] H. Bruyninckx and O. Khatib. Gauss' principle and the dynamics of redundant and constrained manipulators. In *Proc. 2000 IEEE Int. Conf. on Robotics and Animation (ICRA)*, pages 470–475. IEEE, 2000.

[5] A. Chatterjee and A. Ruina. A new algebraic rigid-body collision law based on impulse space considerations. *J. Appl. Mech.*, 65:939–951, 1998.

[6] R. Connelly, E. D. Demaine, and G. Rote. Straightening polygonal arcs and convexifying polygonal cycles. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science, Redondo Beach, California*, pages 432–442. IEEE Computer Society Press, 2000. doi:10.1109/SFCS.2000.892131.

[7] R. Connelly, E. D. Demaine, and G. Rote. Straightening polygonal arcs and convexifying polygonal cycles. *Discrete and Computational Geometry*, 30:205–239, 2003.

[8] Y. Y. Fan, R. E. Kalaba, H. H. Natsuyama, and F. E. Udwadia. Reflections on the Gauss principle of least constraint. *Journal of Optimization Theory and Applications*, 127:475484, 2005. doi:10.1007/s10957-005-7496-7.

[9] J.-A. Ferrez. *On the behaviour of spherical and non-spherical grain assemplies, its modeling and numerical simulation.* Ph.D. thesis, EPFL Lausanne, 2005.

[10] J.-A. Ferrez and T. M. Liebling. Dynamic triangulations for efficient collision detection among spheres with applications in granular media simulations. *Phil. Mag. B*, 82(8):905–929, 2002.

[11] C. F. Gauß. Über ein neues allgemeines Grundgesetz der Mechanik. *J. reine angew. Math. (Crelle's Journal)*, 4:232–235, 1829.

[12] P. Lötstedt. Numerical simulation of time-dependent contact and friction problems in rigid body dynamics. *SIAM J. Sci. Stat. Comput.*, 5(2):370–393, 1984.

[13] V. J. Milenkovic and H. Schmidl. Optimization-based animation. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and*

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

interactive techniques, pages 37–46, New York, NY, USA, 2001. ACM. doi:10.1145/383259.383263.

[14] M. Paterson, Y. Peres, M. Thorup, P. Winkler, and U. Zwick. Maximum overhang. In S.-H. Teng, editor, *Proc. of 19th SODA*, pages 756–765. SIAM, 2008.

[15] M. Paterson, Y. Peres, M. Thorup, P. Winkler, and U. Zwick. Maximum overhang. *Amer. Math. Monthly*, 116(9):763–787, 2009.

[16] M. Paterson and U. Zwick. Overhang. In *Proc. of 17th SODA*, pages 231–240. ACM Press, 2006.

[17] M. Paterson and U. Zwick. Overhang. *Amer. Math. Monthly*, 116(1):19–44, 2009.

[18] L. Pournin, T. M. Liebling, and A. Mocellin. Molecular-dynamics force models for better control of energy dissipation in numerical simulations of dense granular media. *Phys. Rev. E*, 65:011302, 7 pp., 2002. doi:10.1103/PhysRevE.65.011302.

[19] A. Prékopa. On the development of optimization theory. *Amer. Math. Monthly*, 87(7):527–542, 1980.

[20] S. Redon, A. Kheddar, and S. Coquillart. Gauss' least constraints principle and rigid body simulations. In *Proc. 2002 IEEE Int. Conf. on Robotics and Animation (ICRA)*, pages 517–522. IEEE, 2002.

[21] A. Rosato, K. J. Strandburg, F. Prinz, and R. H. Swendsen. Why the Brazil nuts are on top—size segregation of particle matter by shaking. *Phys. Rev. Lett.*, 58:1038–1040, Mar. 1987.

[22] T. Shinar, C. Schroeder, and R. Fedkiw. Two-way coupling of rigid and deformable bodies. In D. James and M. Gross, editors, *ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA)*, pages 95–103, 2008.

[23] T. Shinbrot. Granular materials: The brazil nut effect—in reverse. *Nature*, 429:352–353, May 2004. doi:10.1038/429352b.

[24] R. Weinstein, J. Teran, and R. Fedkiw. Dynamic simulation of articulated rigid bodies with contact and collision. *IEEE Transactions on Visualization and Computer Graphics*, 12:365–374, 2006.

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.