Constraint-based collision and contact handling using impulses

Article · January 2006

CITATIONS
13

READS
348

2 authors, including:

Jan Bender
RWTH Aachen University
50 PUBLICATIONS **278** CITATIONS

SEE PROFILE

Constraint-based collision and contact handling using impulses

Jan Bender
Universität Karlsruhe
jbender@ira.uka.de
http://i31www.ira.uka.de

Alfred Schmitt
Universität Karlsruhe
aschmitt@ira.uka.de
http://i31www.ira.uka.de

Abstract

In this paper a new method for handling collisions and permanent contacts between rigid bodies is presented. Constraint-based methods for computing contact forces with friction provide a high degree of accuracy. The computation is often transformed into an optimization problem and solved with techniques like linear or quadratic programming. Impulse-based methods compute impulses to prevent colliding bodies from interpenetrating. The determination of these impulses is simple and fast. The impulse-based methods are very efficient but they are less accurate than the constraint-based methods because they resolve only one contact between two colliding bodies at the same time. The presented method uses a constraint-based approach. It can handle multiple contacts between two colliding bodies at the same time. For every collision and contact a non-penetration constraint is defined. These constraints are satisfied by iteratively computing impulses. In the same iteration loop impulses for dynamic and static friction are determined. The new method provides the accuracy of a constraint-based method and is efficient and easy to implement like an impulse-based one.

Keywords: physically-based animation, rigid bodies, collision, contact, friction

1 Introduction

The resolution of collisions and permanent contacts with friction is an important part of the dynamic simulation of rigid bodies. A rigid body is defined by its mass m, the position of its centre of mass $\mathbf{c}(t)$, its velocity $\mathbf{v}(t)$, its inertia tensor $\mathbf{J}$, its actual orientation represented by a unit quaternion $\mathbf{q}(t)$ [1] and its angular velocity $\boldsymbol{\omega}(t)$.

There exist two main approaches for collision and contact handling: the impulse-based and the constraint-based approach. The impulse-based method works as follows. For every pair of colliding bodies one pair of closest points is determined and an impulse is computed and applied to resolve the collision at these points. Due to the impulses new collisions can occur. Because of this, the computation of collision impulses is continued in a loop until all collisions are resolved. Since the determination of such impulses runs fast, the method is very efficient. If two bodies are resting on each other, in general they have multiple contacts at the same time but only one contact is handled per simulation step. This results in a vibratory motion of the bodies. There exist some approaches to cope with this problem but if a high degree of accuracy is required, all existing contacts must be resolved at once. This is done by constraint-based methods. These methods define a non-penetration constraint for each contact that occurs. The defined constraints are unilateral and can be formulated as a linear complementarity

problem (LCP). The contact forces can be computed by solving this LCP. Alternatively the constraints can be written as a quadratic program and solved by using quadratic programming algorithms [2].

The goal of the presented method is to combine the advantages of the constraint-based and impulse-based approaches. To avoid problems with vibratory motion between bodies resting on each other and to achieve a higher degree of accuracy, a constraint-based approach was chosen for this method. But instead of transforming the collision and contact resolution into an optimization problem, the non-penetration constraints are satisfied by iteratively computing impulses. The advantage of using impulses is, that the computation is simple and fast. Impulses change the velocities of the bodies directly, so no numerical integration is needed to determine the velocity change due to contact forces.

2 Previous work

Analytical methods define the determination of contact forces as a linear complementarity problem [3]. Either two colliding rigid bodies are accelerating away from each other or a normal force is needed to prevent interpenetration. So either the normal acceleration or the normal force must be zero. Per Löstedt was the first to handle contacts in this way [4]. David Baraff presented a heuristic approach to compute the contact forces analytically [5]. Victor Milenkovic and Harald Schmidl solved the problem of computing non-interpenetrating positions, momenta and accelerations by using quadratic programming [6]. In [7] they used impulses to resolve collisions and contacts with friction and determined the necessary impulses by the solution of quadratic programs. To accelerate the simulation of many objects, Victor Milenkovic proposed a simplified position-based physics and used linear programming [8]. Danny Kaufman et al. presented an algorithm for the simulation of large sets of non-convex rigid bodies [9]. Contacts are solved simultaneously by quadratic programming. The solution is linear in the number of contacts detected in each iteration. David Baraff presented an algorithm to determine the contact forces with friction without solving an optimization problem [10]. This algorithm is faster, simpler and more robust than methods based on optimization techniques.

Brian Mirtich and John Canny proposed an impulse-based method to handle collisions and contacts with friction [11, 12]. They identify the occurrence of a contact by using a velocity threshold. During the simulation of a block resting on a plane small impulses are applied to different corners of the block to prevent interpenetration. These impulses lead to a vibratory motion. In the case of static friction a block slides erroneously down an inclined plane due to these vibrations. A microcollision model was introduced to reduce this error [13]. Eran Guendelman et al. also use an impulse-based approach for collision and contact handling [14]. They proposed a new order of the simulation loop to cope with the vibration problem. After all collisions are processed, only the velocities of the rigid bodies are updated. With the new velocities the contact handling is done and at last the positions are updated. The change of order is not physically correct but with a sufficiently small time step size the results are plausible.

3 Collision response

A dynamic simulation system for non-penetrating rigid bodies needs a collision detection. The collision detection controls the time step size of the dynamic simulator to determine the time of impact. At this point of time the colliding bodies just have contact and do not penetrate each other. Especially in the case of bodies resting on each other, two colliding bodies can have contact in more than one point. In the case of collision or contact, the collision detection determines all contact points and a contact normal $\mathbf{n}$ for each pair of colliding bodies. A contact normal is always directed from the second body to the first one. The collision detection will not be discussed in further detail in this paper.

The collision response of a dynamic simulation system must handle collisions and resting contacts. A collision occurs if two bodies are in contact and their velocity towards each other in

the contact point is greater than zero. If the relative velocity of the bodies in the contact point is equal to zero, then the bodies are in resting contact. In both cases an impulse is computed to prevent the bodies from interpenetrating.

Let us assume that gravity is the only external force that acts on the bodies. In the case of contact between two bodies, let $\mathbf{a}$ be the contact point in the first body and $\mathbf{b}$ the one in the second body and let $\mathbf{u}_a$ and $\mathbf{u}_b$ be the corresponding point velocities. The velocity of a point $\mathbf{a}$ of a rigid body can be computed by

$$\mathbf{u}_a(t) := \mathbf{v}(t) + \boldsymbol{\omega}(t) \times (\mathbf{a}(t) - \mathbf{c}(t)).$$

The relative velocity of the contact points is defined as $\mathbf{u}_{rel} := \mathbf{u}_a - \mathbf{u}_b$. The relative point velocity in normal direction $u'_{rel,n} = \mathbf{u}_{rel} \cdot \mathbf{n}$ is determined to differentiate between three cases. If $u'_{rel,n}$ is greater than zero the bodies are separating and no impulse need to be applied. Otherwise the bodies are in resting contact or a collision occurs. The criterion for a resting contact is

$$-u'_{rel,n} < \sqrt{2\,g\,\varepsilon_c}$$

where ε_c is the tolerance of the collision detection and g is the acceleration of gravity. This criterion was proposed by Brian Mirtich in [13].

The difference between resolving a collision and handling a resting contact is that the duration of a collision is infinitesimal whereas a resting contact lasts for a whole simulation step. Because of this, the resolution of all collisions is done before the simulation step and the contacts are handled during the simulation step (see figure 1). The collision resolution has to be done in a loop because the applied impulses that prevent the bodies from interpenetrating can cause further collisions. The loop ends if no pair of contact points satisfies the criterion for a collision. After all collisions are resolved, impulses for the resting contacts are computed, so that a simulation step can be done without interpenetration. In the following first the resolution of collisions is discussed and then the handling of resting contacts.

3.1 Collision resolution

The impact law of Newton is an approximation of the collision process without friction. It

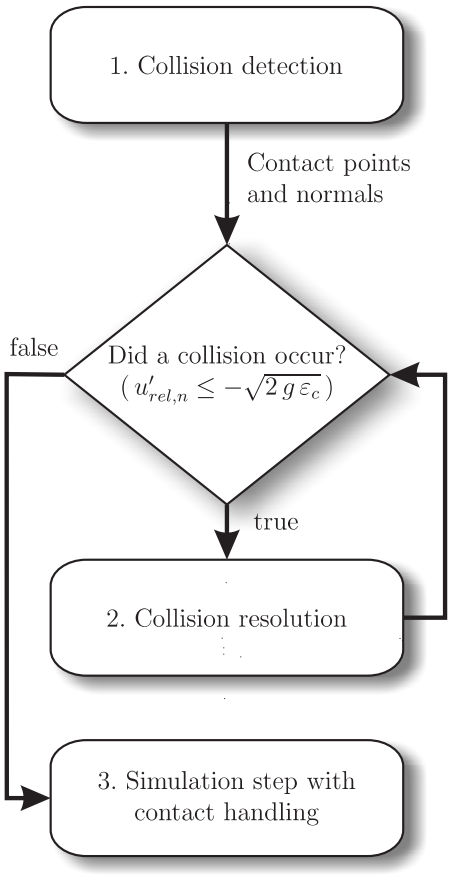


Figure 1: Simulation step with collision and contact handling

states that the relative velocity of a pair of contact points in the direction of the normal after a collision is

$$\mathbf{u}^c_{rel,n} = -e \cdot \mathbf{u}_{rel,n}$$

where e is the coefficient of restitution and $\mathbf{u}_{rel,n} := u'_{rel,n} \cdot \mathbf{n}$ is the normal velocity before the collision. If two bodies collide, an impulse $\mathbf{p}$ has to be applied to the first body and an equal impulse in opposite direction $-\mathbf{p}$ has to be applied to the second body to prevent interpenetration. These impulses must act in the direction of the normal $\mathbf{n}$. To resolve the collision the impulses must change the relative velocity of the contact points by

$$\boldsymbol{\Delta}\mathbf{u}_{rel,n} = \mathbf{u}^c_{rel,n} - \mathbf{u}_{rel,n}.$$

Let $\mathbf{x}$ and $\mathbf{y}$ be two arbitrary points of a rigid body and let $\mathbf{r}_{xc} := \mathbf{x} - \mathbf{c}$ and $\mathbf{r}_{yc} := \mathbf{y} - \mathbf{c}$ be the vectors from the centre of mass to these points. The velocity change $\boldsymbol{\Delta}\mathbf{u}_x$ of the point $\mathbf{x}$

if an impulse $\mathbf{p}$ is applied at the point $\mathbf{y}$ can be determined with the following matrix $\mathbf{K}_{x,y}$:

$$\mathbf{K}_{x,y} := \begin{cases} \frac{1}{m}\,\mathbf{I}_3 - \mathbf{r}^*_{xc}\,\mathbf{J}^{-1}\,\mathbf{r}^*_{yc} & \text{if the body} \\ & \text{is dynamic} \\ \mathbf{0} & \text{otherwise} \end{cases}$$

$$\mathbf{\Delta u}_x \;=\; \mathbf{K}_{x,y} \cdot \mathbf{p}$$

where $\mathbf{r}^*_{xc}$ and $\mathbf{r}^*_{yc}$ are the cross product matrices of the vectors $\mathbf{r}_{xc}$ and $\mathbf{r}_{yc}$ and $\mathbf{I}_3$ is the 3×3 identity matrix. The impulse in normal direction that changes the relative velocity of the pair of contact points $\mathbf{a}$ and $\mathbf{b}$ by $\mathbf{\Delta u}_{rel,n}$ is determined by

$$\mathbf{p}_n = \frac{1}{\mathbf{n}^T\mathbf{K}\mathbf{n}}\mathbf{\Delta u}_{rel,n} \tag{1}$$

where $\mathbf{K} := \mathbf{K}_{a,a} + \mathbf{K}_{b,b}$. The resulting impulse $\mathbf{p}_n$ is applied at point $\mathbf{a}$ and the impulse $-\mathbf{p}_n$ at point $\mathbf{b}$ in order to resolve the collision. If there occur multiple collisions at the same time or two colliding bodies have multiple pairs of contact points, then for each pair an impulse must be computed and applied. Since in general the collision impulses depend on each other this process must be continued in an iterative loop until all collisions are resolved correctly.

The collision resolution with multiple pairs of contact points works as follows. In the beginning the relative velocity $\mathbf{u}^c_{rel,n}$ of each pair of contact points after the collision is computed with Newton's law. Then the computation of the collision impulses is done in an iterative loop. In the i-th iteration for each pair of contact points it is tested if the relative velocity $\mathbf{u}_{rel,n}$ of the pair has already reached the corresponding collision velocity $\mathbf{u}^c_{rel,n}$. If this is true, the computation continues with the next contact. Otherwise an impulse $\mathbf{p}_{n,i}$ is determined with equation 1 to eliminate the actual velocity difference $\mathbf{\Delta u}_{rel,n}$. To prevent the colliding bodies from sticking together it must be ensured that in each iteration the sum of all impulses that are applied to a pair of contact points is in positive normal direction:

$$\mathbf{n} \cdot \sum_{j=1}^{i} \mathbf{p}_{n,j} \geq 0. \tag{2}$$

If condition 2 holds in iteration $i-1$ and is not satisfied anymore in the i-th iteration, then $-\mathbf{n}\cdot\mathbf{p}_{n,i} > \mathbf{n}\cdot\sum_{j=1}^{i-1}\mathbf{p}_{n,j}$ must hold. The following impulse $\mathbf{p}'_{n,i}$ is applied at the contact points instead of $\mathbf{p}_{n,i}$ to ensure that sticking is prevented:

$$\mathbf{p}'_{n,i} = \begin{cases} \mathbf{p}_{n,i} & \text{if } \mathbf{n}\sum_{j=1}^{i}\mathbf{p}_{n,j} \geq 0 \\ -\sum_{j=1}^{i-1}\mathbf{p}_{n,j} & \text{otherwise.} \end{cases} \tag{3}$$

The iteration loop ends, if every pair of contact points has reached its corresponding collision velocity.

The friction that occurs if two bodies collide depends on the collision impulse in normal direction. Because of this, friction impulses are computed in the same iteration loop as the collision impulses. The computation of static and dynamic friction is based on the friction law of Coulomb. Dynamic friction occurs, if the relative tangential velocity $\mathbf{u}_{rel,t} = \mathbf{u}_{rel} - \mathbf{u}_{rel,n}$ between two colliding bodies at the contact point is not zero. To simulate dynamic friction an impulse is computed that acts in the opposite direction of the tangent $\mathbf{t} = \mathbf{u}_{rel,t}/|\mathbf{u}_{rel,t}|$. The friction impulse of a pair of contact points in the i-th iteration is computed by

$$\mathbf{p}_{t,i} = -\mu_d|\mathbf{p}'_{n,i}|\mathbf{t} \tag{4}$$

where μ_d is the coefficient of dynamic friction. Before this impulse is applied to the contact points, it must be ensured that it does not turn the relative tangential velocity $\mathbf{u}_{rel,t}$ in the opposite direction because friction impulses are only allowed to decelerate the bodies. The maximal allowed friction impulse is determined by

$$\mathbf{p}_{t,max} = -\frac{1}{\mathbf{t}^T\mathbf{K}\mathbf{t}}\,\mathbf{u}_{rel,t}. \tag{5}$$

This impulse eliminates the relative tangential velocity completely. If the impulse $\mathbf{p}_{t,max}$ is taken into account, the resulting impulse for dynamic friction is given by

$$\mathbf{p}'_{t,i} = \begin{cases} \mathbf{p}_{t,i} & \text{if } \mathbf{p}_{t,max}\cdot\mathbf{t} < \mathbf{p}_{t,i}\cdot\mathbf{t} \\ \mathbf{p}_{t,max} & \text{otherwise.} \end{cases}$$

In the case that the impulse $\mathbf{p}_{t,max}$ is applied, the relative tangential velocity vanishes and static friction may occur.

Let μ_s be the coefficient of static friction. If in the i-th iteration the relative tangential velocity $\mathbf{u}_{rel,t}$ of a pair of contact points is zero and the condition

$$|\mathbf{p}_t| \leq \mu_s|\mathbf{p}_n| \tag{6}$$

is satisfied for the sum of the tangential impulses $\mathbf{p}_t = \sum_{j=1}^{i} \mathbf{p}'_{t,j}$ and the sum of the normal impulses $\mathbf{p}_n = \sum_{j=1}^{i} \mathbf{p}'_{n,j}$, static friction occurs between the colliding bodies. The sum of friction impulses $\mathbf{p}_t$ is exactly the impulse that has eliminated the relative tangential velocity of the two colliding bodies at the contact point. As long as condition 6 is satisfied, the corresponding pair of contact points is marked to have static friction. The friction impulses computed in the iteration process for such a marked contact must eliminate the relative tangential velocity. An impulse computed with equation 5 eliminates $\mathbf{u}_{rel,t}$ and is therefore exactly the impulse that must be applied to simulate static friction. In the case that a pair of contact points is marked to have static friction but friction is not strong enough to maintain sticking, the bodies start to slide at this position. This happens, if condition 6 is not satisfied anymore. Then the corresponding mark is removed and in the following iteration steps impulses for dynamic friction are applied. To guarantee that contacts where static friction occurs have no relative tangential velocity at the end of the collision resolution, the iteration process continues until this condition is satisfied.

3.2 Contact handling

After the collisions are resolved, all pairs of contact points are tested, if they satisfy the criterion for resting contacts. The contact handling works similar to the resolution of collisions. The only difference is that in the case of a resting contact the bodies remain in contact for a whole simulation step. Hence an impulse in normal direction has to be computed that prevents interpenetration during the whole step in order to satisfy the non-penetration constraint. Let us assume that at the time t_0 all collisions are resolved and for one pair of contact points $\mathbf{a}$ and $\mathbf{b}$ the criterion for a resting contact is satisfied. If the simulation would continue without regarding the non-penetration constraint, in general the rigid bodies would interpenetrate due to the external forces acting on the bodies as shown in figure 2. The distance of the contact points after a simulation step of size h has to be determined. Let $\mathbf{r}_{ac} = \mathbf{a} - \mathbf{c}_1$ and $\mathbf{r}_{bc} = \mathbf{b} - \mathbf{c}_2$ be the vectors from the centres of mass of the colliding rigid bodies to the corresponding contact points at the time of contact t_0. The directions the vectors will have after a simulation step of size h can be computed by numerically integrating the following differential equation:

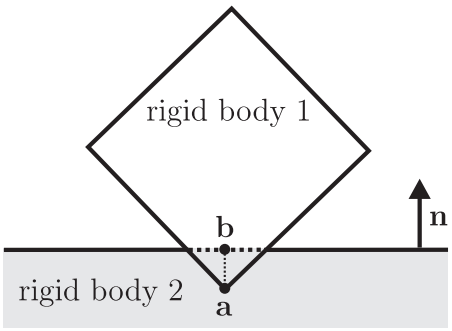


Figure 2: Interpenetration of two bodies

$$\dot{\mathbf{r}} = \boldsymbol{\omega} \times \mathbf{r}. \tag{7}$$

The fourth order Runge-Kutta method is used to do the numerical integration. Since gravity is the only external force, the position of the centre of mass $\mathbf{c}$ of a rigid body after a simulation step is determined by:

$$\mathbf{c}(t_0 + h) = \mathbf{c}(t_0) + \mathbf{v}(t_0)\, h + \frac{1}{2}\, \mathbf{g}\, h^2.$$

If the resulting vectors are added to the new centres of mass, the distance of the contact points after the simulation step can be computed:

$$\begin{aligned} \mathbf{a}(t_0 + h) &= \mathbf{r}_{ac}(t_0 + h) + \mathbf{c}_1(t_0 + h) \\ \mathbf{b}(t_0 + h) &= \mathbf{r}_{bc}(t_0 + h) + \mathbf{c}_2(t_0 + h) \\ \mathbf{d} &= \mathbf{a}(t_0 + h) - \mathbf{b}(t_0 + h). \end{aligned}$$

By numerically integrating differential equation 7, the new direction of the contact normal is determined. The distance and the new contact normal $\mathbf{n}(t_0 + h)$ can be used to compute the penetration depth after the simulation step:

$$d_n(t_0 + h) = -\mathbf{d} \cdot \mathbf{n}(t_0 + h).$$

The impulse that eliminates exactly this penetration depth within one simulation step can be determined by solving a non-linear equation [15]. In the presented method a simplification was used. An impulse is determined that causes a change of the relative velocity of the contact points in normal direction by $d_n(t_0 + h)/h$:

$$\mathbf{p}_n = \frac{1}{\mathbf{n}^T \mathbf{K} \mathbf{n}} \cdot \frac{1}{h} d_n(t_0 + h) \cdot \mathbf{n}. \tag{8}$$

This impulse must be applied at time t_0. Since in general the contact points have no linear motion, the resulting impulse will possibly not resolve the contact directly but it is a good approximation. The contact can be completely resolved by computing impulses with equation 8 in an iterative loop until $d_n(t_0 + h)$ is zero within a tolerance. In practice the time step size h is normally at most 0.04 s (25 frames per second). In several tests with $h = 0.04$ s the desired impulse could always be computed in one or two iteration steps even when a small tolerance of 10^{-6} m was used. The advantage of the used simplification is that the equation for the required impulse is linear and can be solved easily. In the case that multiple resting contacts occur at the same time in the system, the iterative computation of the impulses solves emerging dependencies between the contacts. In this iterative process it must be guaranteed that rigid bodies in contact do not stick together due to the computed impulses. Hence condition 2 must be satisfied for the impulses $\mathbf{p}_{n,i}$ that are determined by using equation 8. Analogous to the resolution of collisions, an impulse $\mathbf{p}'_{n,i}$ computed by equation 3 is applied instead of $\mathbf{p}_{n,i}$ to satisfy the condition in every iteration step.

If a stack of bodies lies on the floor, the body at the top pushes all other bodies in the floor and many iterations are needed to handle all the contacts. Eran Guendelman et al. proposed an algorithm in [14] called shock propagation to reduce the number of iterations needed for such situations. A contact graph is used to describe the dependencies between the contacts in a system of rigid bodies. Every body is represented by a node in the graph. Static bodies are marked as root nodes. For every contact between two bodies the corresponding nodes are connected by an edge. After a given number of iterations the shock propagation starts with the bodies whose corresponding nodes are connected to the root nodes. All contacts in this level of the contact graph are handled. Then the bodies of this level are assigned infinite mass. In the same way the shock propagation method traverses the contact graph level by level. When the traversal ends, all contacts are resolved and the simulation can continue. In this way the contact handling of systems with stacks is more efficient.

The determination of impulses to simulate dynamic friction works exactly in the same way as described for the resolution of collisions. If the relative tangential velocity of a pair of contact points is zero and condition 6 is satisfied, then static friction occurs. The distance the contact points have after a simulation step of size h is given by

$$\mathbf{d}_t(t_0 + h) = -\mathbf{d} - d_n(t_0 + h)\mathbf{n}(t_0 + h).$$

In the case that $\mathbf{d}_t(t_0 + h)$ has zero length, no impulse has to be applied in tangential direction. Otherwise the tangent $\mathbf{t}$ can be determined by normalising the vector $\mathbf{d}_t(t_0 + h)$. The friction impulse that eliminates the distance of the contact points in tangential direction is determined by

$$\mathbf{p}_t = \frac{1}{\mathbf{t}^T\mathbf{K}\mathbf{t}} \cdot \frac{1}{h}\mathbf{d}_t(t_0 + h).$$

With this impulse static friction can be simulated. It assures that the contact points are still at the same position after the simulation step. The iteration process ends, if interpenetration is prevented for all contacts and the distance in tangential direction $\mathbf{d}_t(t_0 + h)$ after the simulation step is zero for all contacts where static friction occurs.

4 Results

The presented method for collision and contact handling was implemented in C++. Different models have been simulated to test the method. First of all, blocks with different coefficients of friction were dropped on an inclined plane. The blocks with a higher coefficient stopped their movement due to static friction and were able to hold their positions until the end of the simulation without any erroneous sliding. The distances the blocks needed for stopping in the case of static friction have been compared to analytically computed stopping distances to verify that the simulation delivers correct results.

To measure the performance of the collision and contact handling a model with 1000 cubes was created. The cubes fall through a funnel in order to cause many collisions and contacts (see figure 3). The performance tests were run on a PC with a 3.4 GHz Intel Pentium 4 processor. The cubes had a side length of 1 m. The restitution coefficient was set to $e = 0.48$ and the

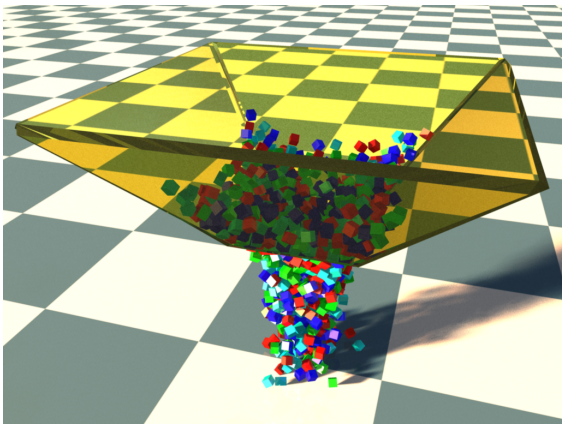
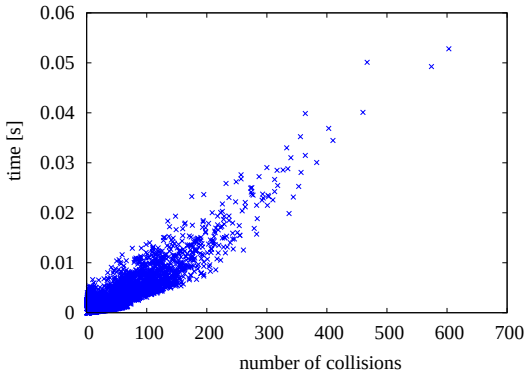


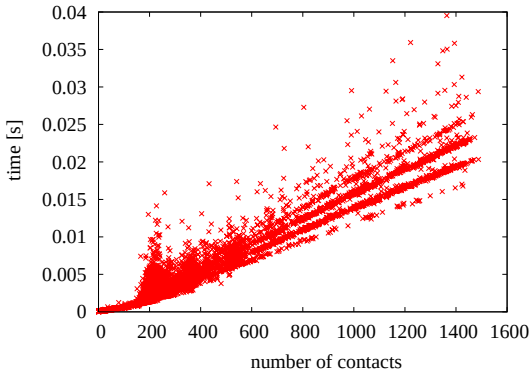
Figure 3: 1000 cubes falling through a funnel.

coefficients of static and dynamic friction were $\mu_s = \mu_d = 0.1$. Resting contacts were resolved until they had at most a penetration depth of 10^{-4} m. 6000 simulation steps were performed with a time step size of $h = 0.001$ s. Shock propagation was not used in this simulation in order to produce very accurate results. The results of the performance tests are presented in figures 4(a) and 4(b). During the simulation the rigid bodies had 41 collisions and 426 resting contacts per time step on average. The simulation method needed 3.09 ms on average for the resolution of all collisions. The average time needed for contact handling was 6.30 ms. Since the time step size and the maximal allowed penetration depth were very small, accurate results could be achieved. The simulation of 1000 cubes with friction was less than ten times slower than real-time. This is a good result for an accurate simulation. If only plausible results are required, the parameters can be changed and the simulation runs much faster.

The next model that was simulated was a tippe top (see figure 5). This is a top that spins first on its spherical body and then after a while it turns around and spins on its stem. The tippe top turns around due to the friction between the top and the plane on which it spins. The simulation of this model is not trivial [16]. It is a good example for showing the capabilities of the presented method concerning frictional effects. The model was simulated with different coefficients of friction. The tippe top always turned around without any problems. In simulations with a small friction coefficient it spun longer on its spherical body before it turned around due to



(a)



(b)

Figure 4: Computation times for collision and contact handling

the lower friction.

For the last simulation the dynamic simulation method of Bender et al. [17] was implemented to simulate joints by iteratively computing impulses. Since the computation of the joint and the contact impulses both work in an iterative loop, the iterative processes can be easily combined. In this way the collisions and contacts for a system with articulated rigid bodies can be resolved. The office toy that is shown in figure 6 was simulated to test if the collision impulses are propagated correctly through the balls. Every ball had a restitution coefficient of $e = 1.0$. When the two balls on the left hit the rest of the balls, they stopped and the two balls on the right bounced off. After the collision the balls on the right had the same velocity as the balls on the left had before the collision. The energy of the system was measured during the simulation to verify that it is conserved. The results showed that the energy was constant except for a small numerical error during the whole simu-

Figure 5: Tippe top

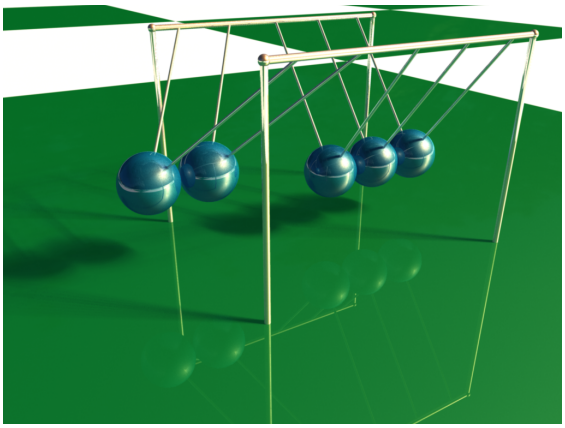


Figure 6: Office toy

lation.

5 Conclusion

A method for collision and contact handling with friction was presented. This method is based on a constrained-based approach and is able to handle collisions and resting contacts between rigid bodies. The method works by iteratively computing impulses. The equations to determine these impulses are simple and no numerical integration has to be done to compute the resulting velocities. Therefore this method is easy to implement. The results have shown that the method is also very fast and even complex frictional effects can be simulated. Contacts can be simulated even faster, if the shock propagation algorithm is used. The presented method can handle multiple contact points between two colliding bodies at the same time. Hence no erroneous vibratory motion occurs and very accurate results can be achieved. Finally, the method was successfully integrated in a dynamic simulation system for articulated rigid bodies.

References

[1] Ken Shoemake. Animating rotation with quaternion curves. In *SIGGRAPH '85: Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 245–254. ACM Press, 1985.

[2] Nick Gould and Philippe Toint. A quadratic programming bibliography, 2001.

[3] Richard W. Cottle, Jong-Shi Pang, and Richard E. Stone. *The linear complementarity problem*. Academic Press, 1992.

[4] Per Lötstedt. Numerical simulation of time-dependent contact and friction problems in rigid body mechanics. 5(2):370–393, June 1984.

[5] David Baraff. Analytical methods for dynamic simulation of non-penetrating rigid bodies. *Computer Graphics*, 23(3):223–232, 1989.

[6] Victor J. Milenkovic and Harald Schmidl. Optimization-based animation. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 37–46, New York, NY, USA, 2001. ACM Press.

[7] Harald Schmidl and Victor J. Milenkovic. A fast impulsive contact suite for rigid body simulation. *IEEE Transactions on Visualization and Computer Graphics*, 10(2):189–197, 2004.

[8] Victor J. Milenkovic. Position-based physics: simulating the motion of many highly interacting spheres and polyhedra. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 129–136, New York, NY, USA, 1996. ACM Press.

[9] Danny M. Kaufman, Timothy Edmunds, and Dinesh K. Pai. Fast frictional dynamics for rigid bodies. *ACM Trans. Graph.*, 24(3):946–956, 2005.

[10] David 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 Press.

[11] Brian Mirtich and John Canny. Impulse-based dynamic simulation. In *WAFR: Proceedings of the workshop on Algorithmic foundations of robotics*, pages 407–418, Natick, MA, USA, 1994. A. K. Peters, Ltd.

[12] Brian Mirtich and John Canny. Impulse-based simulation of rigid bodies. In *SI3D '95: Proceedings of the 1995 symposium on Interactive 3D graphics*, pages 181–ff., New York, NY, USA, 1995. ACM Press.

[13] Brian Vincent Mirtich. *Impulse-based dynamic simulation of rigid body systems*. PhD thesis, University of California, Berkeley, 1996.

[14] Eran Guendelman, Robert Bridson, and Ronald Fedkiw. Nonconvex rigid bodies with stacking. *ACM Transactions on Graphics*, 22(3):871–878, July 2003.

[15] Rachel Weinstein, Joseph Teran, and Ron Fedkiw. Dynamic simulation of articulated rigid bodies with contact and collision. In *IEEE Transactions on Visualization and Computer Graphics*, volume 12, pages 365–374, 2006.

[16] Jörg Sauer and Elmar Schömer. A constraint-based approach to rigid body dynamics for virtual reality applications. In *VRST '98: Proceedings of the ACM symposium on Virtual reality software and technology*, pages 153–162, New York, NY, USA, 1998. ACM Press.

[17] Jan Bender, Dieter Finkenzeller, and Alfred Schmitt. An impulse-based dynamic simulation system for VR applications. In *Proceedings of Virtual Concept 2005*, Biarritz, France, 2005. Springer.