2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Avoiding Zeno's Paradox in Impulse-based Rigid Body Simulation

Evan Drumwright
Department of Computer Science
University of Memphis
Memphis, TN USA
edrmwrgh@memphis.edu

***Abstract*— Treating "resting" contacts (*i.e.*, contacts with zero normal relative velocity), using forces is problematic due to *inconsistent configurations*. For this reason, treating resting contacts with impulses instead of forces has become common, but this approach also suffers from a significant problem: applying impulses at the time-of-contact can keep the simulation from advancing. This scenario is analogous to one of the paradoxes devised by the philosopher Zeno, and has been referred to as a *Zeno point* in the simulation community. I describe how to avoid Zeno points without violating the theoretical dynamic behavior of the simulated bodies and without permitting interpenetration. Two experiments demonstrate that the method works as desired where alternative approaches that required accepting interpenetration or longer running times were previously required.**

I. Introduction

Classical mechanics has separated the concept of contact into two classes: resting contact and impacting contact. The former uses forces to prevent interpenetration and apply friction, while the latter uses impulses to achieve the same effect. Computer-based rigid and multibody simulation has attempted to use this dichotomy as well, but with less success: there exist *inconsistent contact configurations*, in which no set of non-impulsive forces can both prevent interpenetration and model Coulomb friction. As a result, *impulse-based methods* that treat all contacts– both resting and impacting– using impulses are the most widely used approaches for modeling contact.

Impulse-based methods generally are subject to a particular problem when modeling resting contact: *Zeno points* [1] keep the simulation from progressing. Zeno points can occur when impulses are applied to bodies in resting contact (*i.e.*, the bodies are neither moving toward nor away from each other along the contact model): impulses are applied at the time-of-contact (TOC), *which is always the current simulation time for bodies in resting contact*. Figure 2 shows that an infinite loop occurs for the simple case of a box resting on a planar surface. This loop occurs because impacts must be treated (using impulses) before the simulation can progress, which triggers a chain of events: forces are reapplied to the box (applying impulses clears force accumulators[1]), the equations of motion are integrated, and the collision detector again detects contact at the current time. Note that a simple reordering of the simulation process will fix this problem but introduce new ones; for example, if integration is done *after* contacts are treated, external forces (wind, gravity, etc.) will be ignored.

Previous approaches to avoiding Zeno points in simulation consist of methods described by Mirtich [2], Anitescu and Potra [3], and Guendelman et al [4]. These approaches exhibit excessive impulse propagation (and thus running time) for stacked bodies, require computationally expensive implicit integration, and permit interpenetration, respectively. This paper introduces a simple approach that avoids Zeno points when treating contacts– both resting and impacting– using only impulses. Unlike the previous approaches, running times are low, computationally expensive implicit integration is not required and interpenetration does not occur. Only continuous collision detection with extensive time-of-contact reporting (described later) is required. The introduced method is compared against the implicit integration approach of Anitescu and Potra [3] and against a naïve, explicit integration approach in several experiments on multiple scenarios.

II. Background

A. Previous approaches to managing Zeno points

Previous approaches to managing Zeno points treat resting contacts using superelastic (*i.e.*, energy adding) coefficients of restitution, by using implicit integration, or by not stopping the simulation at the time-of-contact.

1) Treating resting contacts as impacts with superelastic restitution: The first approach, introduced by Mirtich [2], treats bodies in resting contact using *microcollisions* (*i.e.*, superelastic coefficients of restitution). Mirtich notes that this approach is subject to problems in scenarios requiring impulse propagation (*e.g.*, for stacked objects).

2) Anitescu and Potra's method: Anitescu and Potra [3] introduced a method that utilizes implicit integration for resting contact and a Poisson collision model for impacting contact. Their method is able to avoid Zeno points, though the solution to a linear complementarity problem ($O(n^3)$ expected time complexity [5], exponential worst-case complexity [6]) is required on every time step. Additionally, extending their method to work with higher order or variable step integrators is non-trivial. Finally, the method of Anitescu and Potra requires a threshold value between resting and impacting contact; the effect of fixing this threshold parameter

[1]An impulse is a force applied over an infinitesimally small interval of time; the magnitude of any external forces within this interval is zero.

has yet to be studied.[2]

3) Stepping method of Guendelman et al.: The third approach for managing Zeno points was introduced by Guendelman et al. [4] and steps the simulation past times of contact. As a result, interpenetration is possible, though Guendelman et al. use semi-implicit integration to mitigate its occurrence. Aside from exhibiting interpenetration and requiring semi-implicit integration, their approach experiences one more issue: impact is not handled at the proper time, so the states of the bodies in contact will be different from at the true time-of-impact. Therefore, their method for treating impact will operate using incorrect dynamics.

B. Mirtich's Timewarp algorithm

Mirtich [7] introduced an approach that some researchers (*e.g.*, Lacoursière [1]) have stated can mitigate the problem of Zeno points. I note that this view is incorrect: Mirtich's "timewarp" algorithm only parallelizes the simulation such that impacts do not require the states of all bodies to be rolled back to a previous time. Mirtich's algorithm requires that bodies can be rolled back to a contact-free state before the impact occurred. For bodies in resting contact treated with impulses, part of the simulation (*i.e.*, all of the bodies in contact with the bodies in resting contact) will not be able to step forward. Not coincidentally, Mirtich uses a penalty method for treating resting contacts in that work.

C. Time stepping methods

Time-stepping methods, surveyed comprehensively by Brogliato et al. [8], avoids the problem of Zeno points altogether. Exemplified by the methods of Anitescu and Potra [9] and Stewart and Trinkle [10], these methods solve a differential algebraic equation (DAE) on every time step– existence of solutions is generally proven– and do not solve for times-of-impact. Obviating the process of solving for times-of-impact is a considerable advantage and permits the simulation to always advance. However, time stepping methods currently suffer from their own set of problems, including difficulty of efficiently determining the set of contacts and computational expense of solving the linear and nonlinear complementarity problems into which the DAEs are generally cast.

III. METHOD

The following discussion differentiates between *active contact constraints*, where the bodies in contact are either *approaching* or *resting* at a contact point, and *inactive contact constraints*, where the bodies bodies in contact are *separating* at a contact point.[3] The discussion assumes the existence of a function *treat-impacts(.)*, that ensures that none of the contact constraints will be active after treatment; this post-condition is key to the success of my approach.

[2]Such an effect may be negligible, but will certainly be measurable, as the Anitescu-Potra method consists of two slightly different methods for treating contact (corresponding to the cases of resting contact and impacting contact).

[3]Given the relative velocity $\boldsymbol{v}_{ab}$ of bodies a and b at a contact point with contact normal $\hat{n}$ pointing toward body a, the bodies are approaching or resting if $\hat{\boldsymbol{n}}^{\mathsf{T}}\boldsymbol{v}_{ab} \leq 0$ and separating if $\hat{\boldsymbol{n}}^{\mathsf{T}}\boldsymbol{v}_{ab} > 0$.

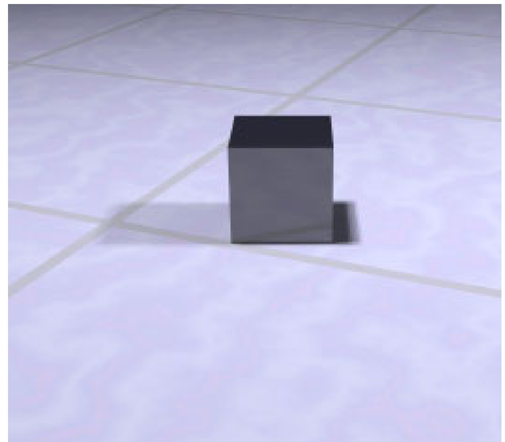


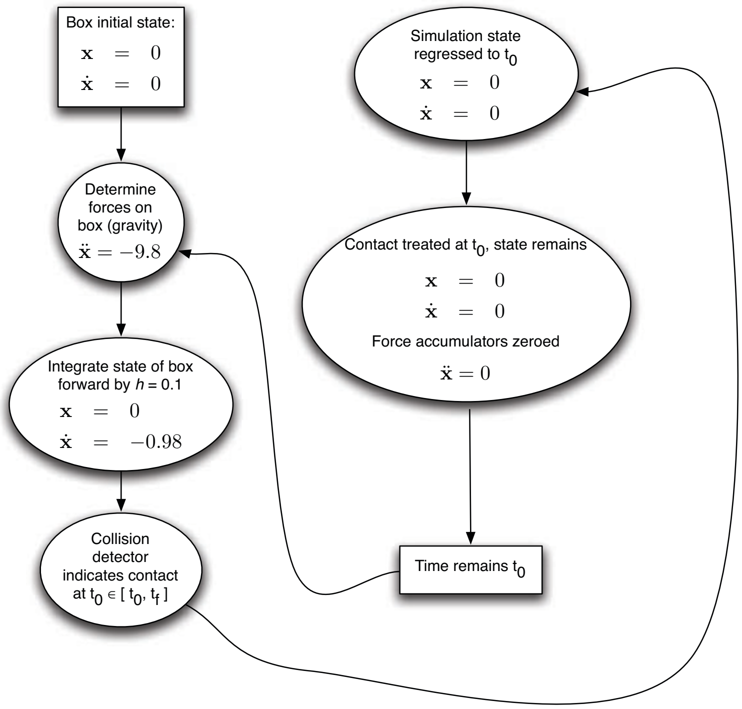
Fig. 1: *A box resting on a plane. This is a prime scenario for Zeno's method, though failure to advance the simulation will not be obvious in this example: there are no other dynamic bodies.*

A naïve approach to stepping rigid and multibody simulations is listed in Algorithm 1. Zeno's paradox may be readily observed with this approach for the scenario of a box resting on a plane (seen in Figure 1). In line 6, a contact will be reported; as the bodies are already in contact, the time-of-contact is equal to the current time (*i.e.*, $t_c = t$) on line 7. The simulation is regressed to time t_c– thus, it does not move forward– and the process repeats indefinitely. If the scenario consists only of a motionless box resting on a plane, the user may be oblivious to the occurrence of the Zeno point; however, if the box were moving– sliding along the plane, for example– or the scenario incorporated multiple dynamic bodies, the simulation would appear to freeze.

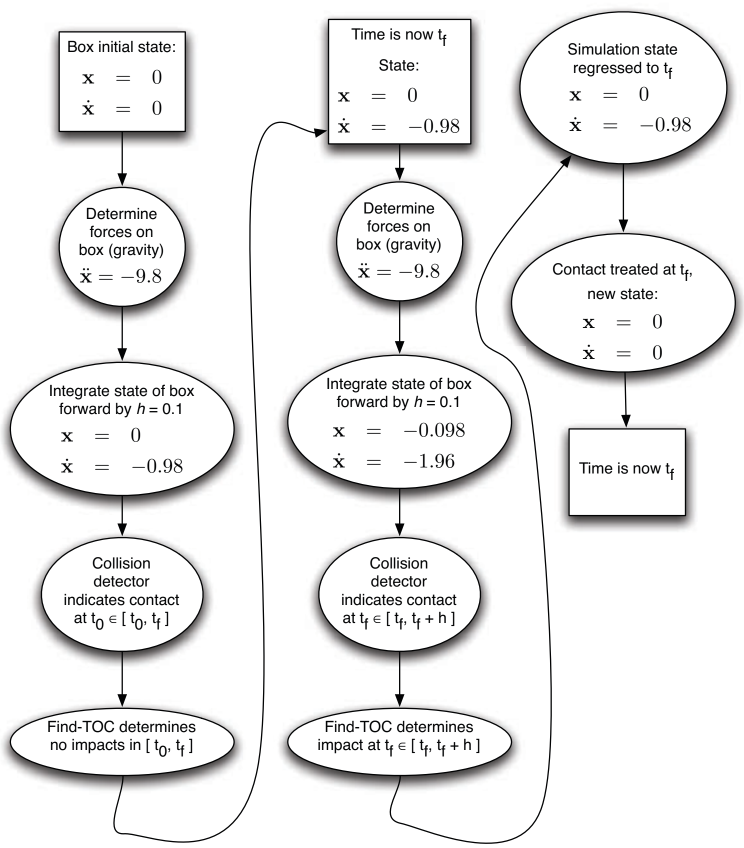
Algorithm 1 Naïve approach to stepping impulse-based simulations

1: $t \leftarrow 0$
2: **while true do**
3: determine centrifugal, Coriolis, constraint, and external forces on all bodies
4: calculate forward dynamics for all bodies
5: integrate states of all bodies forward by h
6: **if** impact $\in [t, t+h]$ **then**
7: backup body states to time-of-contact, t_c
8: *treat-impacts*(.)
9: $t \leftarrow t_c$
10: **else**
11: $t \leftarrow t + h$
12: clear force accumulators

The high-level idea behind my approach is that resting contact can be ignored: the bodies will either continue resting or begin separating (in either case, no impulses need be applied) or will start to impact in the near future; impulses can be applied at that time. A set of contacts must consist of at least a single impacting contact for contact treatment



(a) Simulating a box resting on a plane using a naïve step with a purely impulse-based method



(b) Simulating a box resting on a plane using the introduced method

Fig. 2: *The process of simulating a box resting on a plane with impulse-based methods. Figure (a) shows how the box would traditionally be simulated; the Zeno point is indicated by the loop, which keeps the simulation from progressing. Figure (b) shows how the box is simulated using the introduced method, which allows the simulation to advance correctly.*

Algorithm 2 Top-level simulation process

$t \leftarrow 0$
while true do
  determine centrifugal, Coriolis, constraint, and external forces on all bodies
  calculate forward dynamics for all bodies
  step forward by h (Algorithm 3)
  $t \leftarrow t + h$
  clear force accumulators

Algorithm 3 Steps the simulation from t_0 to $t_0 + dt$, possibly in the presence of impacts.

1: $t \leftarrow t_0$
2: $\{t_c, \mathcal{C}\}$ = *find-TOC*$(t, t + dt)$
3: integrate states of all bodies forward by dt
4: **if** $t_c - t \leq dt$ **then**
5:   **while** $dt > 0$ **do**
6:     backup all bodies to dynamic states at t
7:     integrate states of all bodies forward to t_c
8:     $treat - impacts(.)$
9:     re-apply forces to treated bodies
10:     $dt \leftarrow dt - t_c$
11:     $t \leftarrow t_c$
12:     $\{t_c, \mathcal{C}\}$ = *find-TOC*$(t, t + dt)$
13:     integrate states of all bodies forward by dt
14:     **if** $t_c > dt$ **then**
15:       **break** {no impacts in $[t, t + dt]$}

to occur; otherwise, the simulation is stepped to the next time-of-contact. Once an impact between two bodies is detected and then treated, the bodies will not impact again for some non-zero length of time (as a post-condition of *treat-impacts*(.)).

My approach requires the use of a *continuous collision detection* method (*e.g.*, [11], [12], [13]) that can compute all times-of-contact (*i.e.*, not just the first) over an interval of time. This requirement imposed on the continuous collision detection system is described below.

A. *Description of algorithms*

My method consists of Algorithms 2, 3, and 4. Algorithm 2 is the top-level replacement for Algorithm 1; lines 5–11 of this latter algorithm are replaced by a call to Algorithm 3 in the new approach.

Algorithm 3 is responsible for advancing the simulation up to, and over, times of contact. Again, this algorithm is similar to lines 5–11 of Algorithm 1. The most significant changes are on lines 2 and 12, where *find-TOC(.)* locates the next time-of-contact.

It is this function, *find-TOC(.)*, that is listed in Algorithm 4. Line 1 of the algorithm retrieves the set of pairs of times-of-contact and corresponding contact data (*i.e.*, contact point and normal) between the bodies in the simulation from the continuous collision detection method. Lines 5–10 locate the earliest time-of-contact; the case of multiple earliest times-of-contact (*i.e.*, simultaneous contact) is treated on line 9 and all such contacts are inserted into the set $\mathcal{C}$. If the earliest time-of-contact lies outside of the time interval, the function returns without any contact points (lines 11–12). If at least one contact corresponds to impacting contact, the function returns with the time-of-contact and the set of contacts. Otherwise, the search for the earliest contact repeats, after first removing the contacts just examined (*i.e.*, those contacts that do not indicate an impact has occurred) from further consideration (line 16). If no contacts are impacting at any point throughout the interval, the function returns without any contact points (line 19).

Finally, note that lines 15–17 are the motivation behind the requirement that the collision detection system return all points of contact: the earliest points of contact may indicate resting or separating contact (thereby causing those points to be removed from consideration) although there may exist other points of contact between the two bodies that, shortly thereafter, will indicate impact. If the continuous collision detection system reports only the first times of contact, there will be no means to find this impact.

IV. Experiments

Two experiments test the ability of simulator stepping methods to deal with Zeno points. The first experiment, a pair of spinning boxes, uses a bouncing ball to mark passage of time and, hence, detect any Zeno points. The second experiment utilizes two sliding boxes to mark the passage of time; if a Zeno point occurs, the boxes will not slide.

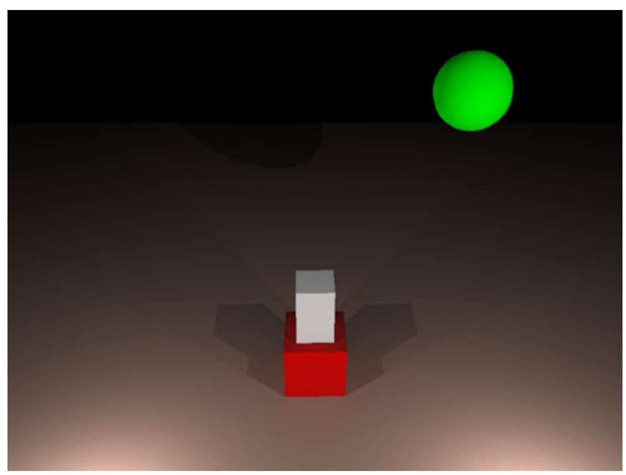


Fig. 3: *The scenario of the first example as the simulation begins. The naïve integration approach freezes the simulation at this point.*

Each experiment uses a different simulated scenario and each scenario is tested on three methods: the naïve integration approach, the implicit integration method of Anitescu and Potra,[4] and the method introduced in this paper (the method of Guendelman et al. [4] is used to treat impacts). All

[4]A sixteen sided polygon is used to approximate the friction cone. A Schur complement and Lemke's algorithm [14], [6], [5] is used to solve the mixed linear complementarity problem.

Algorithm 4 FIND-TOC(t_0, t_f)

```
T = CCD(t_0, t_f) {Get contact times and data from CCD}
C ← ∅
t_min = ∞ {Init time of first contact}
repeat
  for all (t, c) ∈ T do
    if t < t_min then {new first TOC}
      C ← {c}
      t_min ← t
    else if t = t_min then {TOC equal to first TOC (simultaneous contact)}
      C ← C ∪ {c}
  if t_min > t_f then {first contact after t_0 + dt}
    return {∞, ∅}
  if ∃c ∈ C s.t. c indicates bodies are impacting then
    return {t_min, C} {Return all contacts that occur at t_min}
  else {no contacts in C are impacting}
    for all c ∈ C do
      T ← T − (t_min, c) {Do not examine this contact further}
    t_min = ∞ {Reset t_min}
until [no contacts remaining] T = ∅
return {∞, ∅} {indicate no contact in [t_0, t_f]}
```

examples set acceleration due to gravity to -9.8m/s^2 and use a step size of 0.001 with Euler integration. The continuous collision detection method of Shell and Drumwright [13] determines times-of-contact as well as contact points and normals. The author's freely available multibody dynamics library, *Moby* [15], is used to simulate all experiments. Timing information for the Anitescu-Potra and the introduced method are provided in Table I; the timings include all aspects of the simulation process, including collision detection and contact treatment.

A. Spinning boxes with bouncing ball

The scenario in the first example consists of two spinning, stacked boxes and a bouncing ball, as seen in Figure 3. The coefficient of restitution between each of the boxes is zero (*i.e.*, fully inelastic collision), as is the coefficient of restitution between the bottom box and the ground plane; the ground plane has infinite inertia, so it is immobile. The coefficient of restitution between the ball and the ground plane is 1.0 (*i.e.*, fully elastic collision). The coefficients of friction between all pairs of objects in the scenario are zero. The top box spins with an angular speed of 20 rad/s and the bottom box spins with an angular speed of 10 rad/s. The angular velocity of the sphere is zero initially, and the linear velocities of all bodies are zero.

As can be seen from Figure 4, the boxes are able to spin as the ball falls and bounces. Given that the example begins with the boxes in resting contact, a Zeno point occurs immediately and the simulation freezes when modeling this scenario with naïve integration: the boxes do not rotate and the ball remains suspended in mid-air. Both the Anitescu and Potra approach and the method introduced in this paper are able to simulate this scenario properly– as seen in Figure 4– though the LCP solver used by the former method experiences numerical difficulties and fails after one second of simulation time.[5]

B. Sliding boxes

The second experiment consists of two 1m^3, 1kg boxes (see Figure 5). The coefficients of kinetic restitution between the boxes and the ground plane (which is immobile) are zero, and the coefficient of restitution between the boxes is zero. The coefficients of friction between all pairs of bodies are zero. When the simulation starts, one box moves with a linear velocity of 10 m/s while the other is stationary; as the simulation begins, the boxes are about to collide. According to nonrelativistic classical mechanics and Newton's model of restitution, both boxes will move together at 5 m/s after the collision.

This simple example tests three bodies in fully connected contact: each box is in contact with the ground, and the boxes are also in contact with each other. As can be seen from Figure 5, the algorithm keeps the boxes from falling through the floor while correctly handling the impact between the boxes. As with the previous example, the naïve integration approach does not allow the simulation to progress; the boxes rest on the ground but do not slide. Both the Anitescu-Potra method and the introduced method are able to correctly simulate the sliding boxes for ten seconds of simulation time.

V. Conclusions

I have presented an algorithm that provides an alternative means to deal with the issue of Zeno points in impulse-based rigid body simulation. Interpenetration is prevented and any type of integration scheme can be readily employed.

[5] Increasing the robustness of LCP solvers is the subject of current research; see *e.g.*, [16].

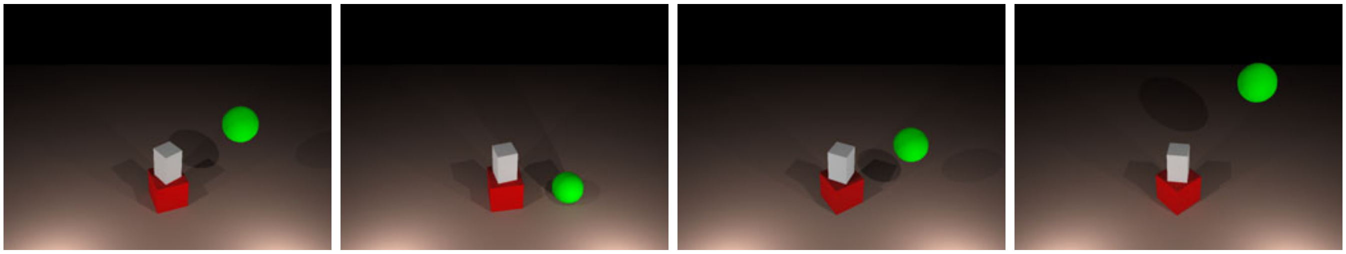


Fig. 4: *Snapshots taken over the execution of the bouncing ball example. Note that both boxes are spinning at the same time the ball is bouncing and that the boxes are undergoing resting contact (one box is resting on the other, and that box is resting on the ground).*

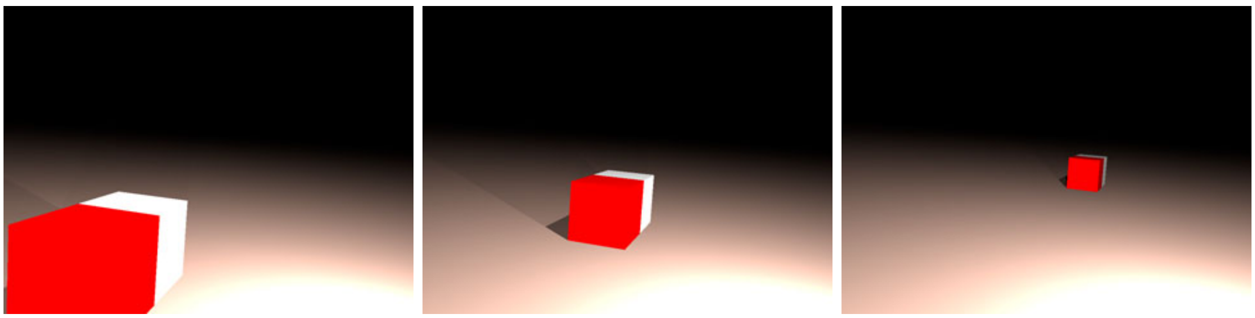


Fig. 5: *Snapshots taken over the execution of the sliding boxes example*

TABLE I: *"Wall clock" timings for methods on the scenarios. The naïve method is unable to step the simulation through the desired amount of simulation time because of Zeno points.*

	Introduced method	Anitescu-Potra method	naïve method
Boxes and sphere (1.25s simulation time)	3.20s	25.73s	∞
Sliding boxes (10s simulation time)	5.17s	6.29s	∞

The *Moby* simulation library uses the introduced approach to simulate resting contact for Newton, Mirtich, Anitescu-Potra, and convex optimization based impact models (among others). The primary advantage over the implicit integration method of Anitescu and Potra is the lower running time that such alternative methods can yield, as the results in Table I can testify. The experiments illustrate that the algorithm both prevents Zeno points from occurring and correctly models the dynamics of rigid bodies.

References

[1] C. Lacoursière, "Ghosts and machines: Regularized variational methods for interactive simulations of multibodies with dry frictional contacts," Ph.D. dissertation, Umeå University, 2007.

[2] B. Mirtich, "Impulse-based dynamic simulation of rigid body systems," Ph.D. dissertation, University of California, Berkeley, 1996.

[3] M. Anitescu and F. A. Potra, "Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems," *Nonlinear Dynamics*, vol. 14, pp. 231–247, 1997.

[4] E. Guendelman, R. Bridson, and R. Fedkiw, "Nonconvex rigid bodies with stacking," *ACM Trans. on Graphics*, vol. 22, no. 3, pp. 871–878, 2003.

[5] R. W. Cottle, J.-S. Pang, and R. Stone, *The Linear Complementarity Problem*. Boston: Academic Press, 1992.

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

[7] B. Mirtich, "Timewarp rigid body simulation," Mitsubishi Electronic Research Laboratory, Tech. Rep. TR-2000-17, April 2000.

[8] B. Brogliato, A. ten Dam, L. Paoli, F. Génot, and M. Abadie, "Numerical simulation of finite dimensional multibody nonsmooth mechanical systems," *ASME Appl. Mech. Reviews*, vol. 55, no. 2, pp. 107–150, March 2002.

[9] M. Anitescu, F. Potra, and D. Stewart, "Time-stepping for three dimensional rigid body dynamics," *Computer Methods in Applied Mechanics and Engineering*, vol. 177, pp. 183–197, 1999.

[10] D. Stewart and J. Trinkle, "An implicit time-stepping scheme for rigid body dynamics with coulomb friction," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, San Francisco, CA, April 2000.

[11] S. Redon, A. Kheddar, and S. Coquillart, "Fast continuous collision detection between rigid bodies," in *Proc. of Eurographics (Computer Graphics Forum)*, 2002.

[12] X. Zhang, M. Lee, and Y. J. Kim, "Interactive continuous collision detection for non-convex polyhedra," in *The Visual Computer (Proc. of Pacific Graphics)*, 2006.

[13] D. Shell and E. Drumwright, "Precise generalized contact point and normal determination for rigid body simulation," in *Proc. of the ACM Symp. on Applied Computing*, Honolulu, HI, March 2009, pp. 2107–2108.

[14] C. E. Lemke, "Bimatrix equilibrium points and mathematical programming," *Management Science*, vol. 11, pp. 681–689, 1965.

[15] E. Drumwright, "Moby," http://physsim.sourceforge.net.

[16] K. Yamane and Y. Nakamura, "A numerically robust LCP solver for simulating articulated rigid bodies in contact," in *Proc. of Robotics: Science and Systems*, Zurich, Switzerland, June 2008.