Workshop on Virtual Reality Interaction and Physical Simulation VRIPHYS (2010)
J. Bender, K. Erleben, and M. Teschner (Editors)

Normal and Friction Stabilization Techniques for Interactive Rigid Body Constraint-based Contact Force Computations

Morten Silcowitz[†] Sarah Niebe [‡] Kenny Erleben [§]

eScience Center, Department of Computer Science, University of Copenhagen, Denmark

Abstract

We present a novel, yet simple, method for stabilization of normal forces. A normal stabilization term, carefully designed from hypotheses about interactive usability, is added to the contact force problem. Further, we propose friction stabilization as a completely new stabilization paradigm in interactive simulation. We present a comparison between our normal stabilization method and the Baumgarte stabilization method under extreme interactive conditions. Preliminary results on friction stabilization are presented, showing both the potential advantages of the method and that there is still work to be done.

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.5]: Physically based modeling—Computer Graphics [I.3.7]: Animation—

Keywords: Rigid Body, Contact Forces, Stabilization, Interactivity

1. Introduction

Stabilization is unavoidable in interactive simulators. Interactivity puts time constraints on the numerical methods, which often leads to terminating the methods before they reach accurate solutions. An example is the state-of-the-art projected Gauss-Seidel (PGS) method [Erl07, SNE10c, PNE10, SNE10b]. The PGS method works in an iterative manner and simulators often makes hard limits on the number of iterations allowed [Smi00, Cou05]. Often 30-50 iterations is all one can afford while maintaining interactivity. Naturally, this leads to inaccurate contact forces and thus violations of constraints in the simulated world.

We address the errors that are related to the contact points: penetration and alignment errors. We also raise the question of stabilizing sliding and sticking states and show some preliminary results on this new concept in an interactive simulation. The challenge is to create practical and robust contact stabilization methods, suitable for interactive simulation of rigid bodies. Figure 1 shows frame grabs from an interactive simulation setup applying our stabilization methods. In the example, stabilization is paramount to a successful task completion.

Penetration and separation measures can be uniquely defined from a geometric viewpoint, and many works have addressed the issue of generation contact points. In this paper we will, due to space consideration, side-step the issue of contact point generation. Our simulator uses the method presented in [SNE10a].

We treat related work in Section 1.1. Hereafter we present normal and friction stabilization in Section 2 and 3. Afterwards we show results in Section 4 and finally we conclude in Section 5.

1.1. Previous work on Stabilization

Simulation errors can be corrected on a positional level by moving objects, such that constraint violations disappear. This is often called pre-stabilization or projection error correction [Bar95, CP03, Erl05]. An alternative is to add penalty forces to the simulator, penalizing constraint violations [BB88]. This technique is widely known as Baumgarte stabilization [CP03], it is among the most frequently used in state-of-the-art simulators [Smi00, Cou05]. In position based physics [MHHR07, Mö08], an initial prediction of

† e-mail: mosh@diku.dk
‡ e-mail: niebe@diku.dk
§ e-mail: kenny@diku.dk

DOI: 10.2312/PE/vriphys/vriphys10/089-095

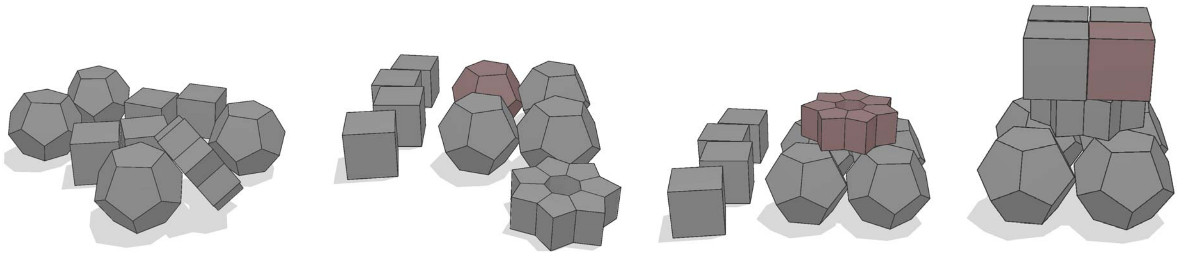


Figure 1: *An interactive simulation where the end user is performing a predefined design task. First the user builds a stack of objects, next the user tips the stack without tipping it over by pressing down on top most object. Our stabilization methods creates high fidelity in such scenarios.*

positions and velocities is followed by a position level projection. Finally, a velocity correction is applied so the velocities agree with the projected positions. Recently, stabilization on the velocity level has been popular. In [WTF05], a pre-stabilization technique is used. Here, impulses are used to update velocities to satisfy constraints on a positional level, prior to the integration of the motion. Also, a velocity post-stabilization is applied, similar to the one used in [GBF03], where velocity errors in stacks are corrected using shock-propagation. A similar idea has been applied in impulse-based simulation [BS06]. Integration schemes exist which apply both pre- and post-stabilization [SSF08]. Recently, [SSF09] presented a post stabilization technique based on energy correction. Their simulations were not interactive, although they used time steps of the same size as the frame-time. The correction problem addressed in their paper is caused by the time discretization of the gyroscopic forces. An explicit time discretization is employed, which naturally affects the energy balance greatly when taking large time-steps. This has been addressed priorly by [Lac07]. In contrast, our work addresses stabilization of contact forces in interactive simulation.

In summary: Baumgarte stabilization is the most frequently used method for stabilization. The categories, pre- and post-stabilization, refer to whether the stabilization is done before or after the actual integration. Pre- and post-stabilization can be done on a positional level, by moving objects before/after integration to make a correction that meets some invariant. It can also be on a velocity level, before/after integration impulses are applied to make a position update agree with an invariant.

All previous work have addressed error correction and stabilization techniques to combat constraint violations such as joint drifting or unwanted penetrations. We propose a novel stabilization technique that aids in interactive manipulation of rigid bodies. We also present a friction post-stabilization method. To our knowledge, stabilization of friction forces in interactive simulation is not addressed in any prior work. Our presented method can easily be appended to existing simulators, without having to rewrite the constraint solver.

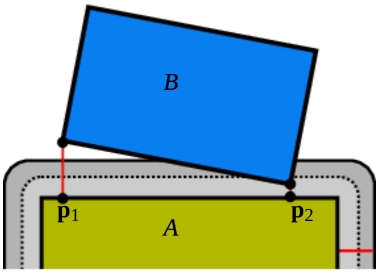
2. Normal Stabilization

Contact points can model objects being slightly misaligned against each other during resting contact, due to the collision envelope has a finite extend. Objects may drift into misaligned states, because of numerical errors. Since contacts are mostly modelled using repulsive forces, this effect can be further enlarged. We propose a simple stabilization approach, where contact points will tend to be aligned with the contact normal plane. If a contact point is farther away than the contact plane, a small correcting velocity is allowed for the contact. Care must be taken when computing the correcting velocity, so they do not introduce further instability. Such instabilities can be caused by trying to force contacts out of the collision envelope. Contacts that lie in the collision envelope are shallow penetrations, forcing them out of the envelope can cause an oscillation of the contact point, switching between being inside and outside the collision envelope. To counter this, we should to compute correction velocities such that the contact points are not pushed all the way out of the envelope.

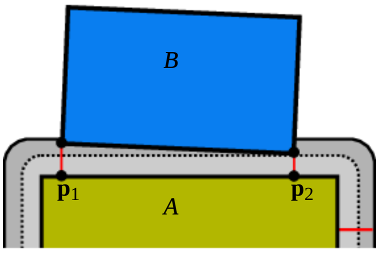
Ideally, the distance measure of a contact point is zero, however, this is not the case in a practical implementation of a simulator. Instead, we define a collision envelope surrounding the colliding objects, and decide that a contact point $\mathbf{p}$ is in true contact if its distance measure d is less than the ideal distance. The ideal distance is defined as half the size of the collision envelope. Thus, two cases are of interest:

- $d < \frac{1}{2}$, the contact point is violating the non-penetration constraint.
- $\frac{1}{2} < d <$, the non-penetration constraint is not violated, but the contact point is displaced within the collision envelope.

Figure 2 illustrates the two cases. The reason for dividing the collision envelope in two is to minimize the visual artefacts related to error correction. A repulsive force will be applied to contact points violating the non-penetration constraint to ensure separation of the overlapping objects. Contact points that are not violating the non-penetration constraint – but lie within the collision envelope – will be corrected by a force that takes the minor penetration into account. This ensures that the resulting correction is less prone to cause new penetrations in the following time step.



(a)



(b)

Figure 2: *Collision envelopes. (a) The contact point $\mathbf{p}_1$ is outside the collision envelope, and is regarded as completely separated. Conversely, contact point $\mathbf{p}_2$ is below the ideal contact plane and is regarded as a penetrating contact. (b) The distance measure of contact point $\mathbf{p}_1$ is slightly larger than half the size of the collision envelope, and is therefore seen as misaligned in the collision envelope. The contact point $\mathbf{p}_2$ is aligned with the ideal contact plane, because it coincides with the center of the collision envelope. Witness points are marked by a filled circle.*

In a velocity based setting, which is the most common in interactive simulators based on the projected Gauss–Seidel method [Erl07], the part of the contact constraint referring to the contact normal is the complementarity constraint. Here we consider a single contact point without loss of generality. Let $\mathbf{A}$ be the linear coupling of contact impulses   and contact velocities $\mathbf{w}$ and $\mathbf{b}$ the right hand side vector. We will partition the system into normal components and tangential components. That is $\ = [\ _n \quad _t^T]^T$ and $\mathbf{b} = [b_n \quad \mathbf{b}_t^T]^T$. Here $_n$ is the Lagrange multiplier corresponding to the normal contact impulses and $_t$ is the vector of Lagrange multipliers corresponding to the tangential friction impulses. The Lagrange multipliers are given by the complementarity constraint,

$$w_n = (\mathbf{A}\ )_n + b_n \geq 0, \tag{1a}$$
$$_n \geq 0, \tag{1b}$$
$$w_n \ _n = 0. \tag{1c}$$

The normal component of the right hand side vector can be written as,

$$b_n = u_n - u_{\text{collision}}, \tag{2a}$$
$$u_{\text{collision}} = -e\min\{0, u_n\}, \tag{2b}$$

where u_n is the initial normal velocity, $u_{\text{collision}}$ is the relative normal velocity after collision, and e is the coefficient of restitution.

Note, the final normal velocity at the contact point is

$$w_n = (\mathbf{A}\ )_n + b_n \tag{3}$$

and w_n is the first order time-derivative of the normal distance measure d. The value of d can be seen as a measure of geometric error, induced by fixed time stepping and numerical drift. Our strategy is to eliminate this error using a normal correction term, c_n, in the velocity constraint,

$$w_n = (\mathbf{A}\ )_n + b_n - c_n. \tag{4}$$

We define the true contact plane, placed at distance $\frac{}{2}$ from the object, so ideally $d = \frac{}{2}$. For a small displacement to be corrected during the next time step, we need a correcting normal velocity, $u_{\text{correction}}$, at the contact point equal to

$$u_{\text{correction}} = \left(\frac{}{2} - d\right)\frac{1}{\Delta t}, \tag{5}$$

where Δt is the simulation time step. In contrast, our model requires

$$w_n \geq u_{\text{collision}}. \tag{6}$$

We must therefore be careful in choosing when and how to use the correction term c_n. It makes good sense to focus on contact points that can be expected to stay within the envelope during the next time-step. Contacts that are leaving the collision envelope completely are ignored. This leads to the implication

$$u_{\text{collision}} > (\ \ - d)\frac{1}{\Delta t} \Rightarrow c_n = 0. \tag{7}$$

Next, depending on the sign of $u_{\text{correction}}$, we make the following choices

$$u_{\text{correction}} \geq 0 \Rightarrow c_n = \max\{0, u_{\text{correction}} - u_{\text{collision}}\}, \tag{8a}$$
$$u_{\text{correction}} < 0 \Rightarrow c_n = u_{\text{correction}}. \tag{8b}$$

Since $u_{\text{collision}} \geq 0$, equation (8a) ensures that we never set c_n larger than necessary. In the case of $u_{\text{correction}} < 0$ we simply ignore the value of $u_{\text{collision}}$. This means that in the case of a resting contact, where $u_{\text{collision}} = 0$, we have $c_n = u_{\text{correction}}$. Finally, we limit the magnitude of c_n in all cases to a given maximum to prevent the correction velocity from becoming too large, for instance in the case of excessive penetration. Currently a maximum value of 2 has worked well for our test cases. In comparison, for a velocity-based setting such as ours the classical Baumgarte stabilization has the form [Erl05],

$$c_n = -\frac{1}{\Delta t} k_n d, \tag{9}$$

where $0 \leq k_n < 1$ is an error reduction parameter with a typical setting around 0.8.

3. Friction Stabilization

When the projected Gauss-Seidel (PGS) method solves for friction forces, the type of friction: sticking, rolling or sliding, is determined by complementarity constraints. Being an iterative method used for interactive simulation, the PGS method will not be able to compute all friction constraints exactly. Here, we propose a post-stabilization technique that is an attempt to improve on incorrect contact friction states. From our working experience with the PGS method, we have observed that the method appears to favor sliding friction over other types of friction. Our experience suggests that this is a poor choice for stable stacks and structures, as these rely on sticking friction to "bind" the structures into an equilibrium state. Thus, our hypothesis is that if sticking friction can solve a problem, then sliding or rolling friction should not be used.

Without loss of generality, we will describe our approach in the following for a single contact point. We are interested in classifying the contact points subject to static friction. For a single contact point we know from Coulomb's friction law,

$$\| \;_t \| < \mu \;_n \quad \wedge \quad \;_n > 0 \quad \Rightarrow \quad \text{sticking} \tag{10}$$

where μ is the coefficient of friction. If we observe sliding behaviour of a contact point that has been classified as static friction, then an error has occurred. To determine the sliding behaviour, we make use of temporal contact point tracking. For instance, if we observe a contact point between rigid bodies A and B in simulation step t then we store the position $\mathbf{p}$ of the contact point with respect to the body spaces. That is,

$$\mathbf{p}_A^t = \left(\mathbf{R}_A^t\right)^T \left(\mathbf{p}^t - \mathbf{r}_A^t\right), \tag{11a}$$

$$\mathbf{p}_B^t = \left(\mathbf{R}_B^t\right)^T \left(\mathbf{p}^t - \mathbf{r}_B^t\right), \tag{11b}$$

where $\mathbf{r}_A$ and $\mathbf{r}_B$ are center of mass positions and $\mathbf{R}_A$ and $\mathbf{R}_B$ are body frame orientations, superscript t denotes the time step. Now, when time step $t+1$ has been performed, we recompute the world space positions of the local contact positions from the previous step and project them onto the current contact plane,

$$\mathbf{s}_A^{(t+1)} = \mathbf{P}_n^{(t+1)} \left(\mathbf{R}_A^{(t+1)} \mathbf{p}_A^t + \mathbf{r}_A^{(t+1)}\right), \tag{12a}$$

$$\mathbf{s}_B^{(t+1)} = \mathbf{P}_n^{(t+1)} \left(\mathbf{R}_B^{(t+1)} \mathbf{p}_B^t + \mathbf{r}_B^{(t+1)}\right), \tag{12b}$$

where $\mathbf{P}_n^{t+1} = \mathbf{I} - \mathbf{n}^{(t+1)} (\mathbf{n}^{(t+1)})^T$ is the projection matrix onto the contact plane given by the contact normal $\mathbf{n}^{t+1}$ and $\mathbf{I}$ is the identity matrix. The sliding error is now computed as

$$\mathbf{e}^{(t+1)} = \mathbf{s}_A^{(t+1)} - \mathbf{s}_B^{(t+1)}. \tag{13}$$

Given an user specified positive tolerance $\; \in \mathbb{R}_+$ to counter numerical impression we determine sliding behaviour, if

$$\| \mathbf{e}^{(t+1)} \| > \; . \tag{14}$$

To correct the sliding error we apply a Baumgarte stabilization term for the simulation step $t+2$. Baumgarte stabilization is used because there is no preferred direction, as was the case with normal stabilization. The stabilization term is computed as,

$$\mathbf{c}_t = \frac{1}{\Delta t} k_t \left(\| \mathbf{e}^{(t+1)} \|\right), \tag{15}$$

where the error reduction parameter, $k_t \approx 0.8$ is used.

In summary, our method is as follow: loop over all contact points and use (10) to determine the subset of static friction contact points, use (14) to determine if an incorrect sliding behaviour has occurred. If an error has occurred, compute a stabilization term for the next simulation step using (15).

Obviously, there are two major difficulties with our approach. One is the choice of a classification rule and the second is the need for contact point tracking [Mir98]. Our choice of static friction classifier is the least committed naive rule we could come up with. Whether this is a good or bad classifier, has yet to be examined. In our work, we experienced that static and rolling friction could be mistaken using this classifier, however, for stacking and manipulation of rigid boxes, rolling friction is unusual. The presented friction stabilization is highly usable in such scenarios. If the classification is wrong, the Baumgarte stabilization will introduce energy into contact points and in worst case cause a simulation blow-up. Contact point tracking is a well known technique in many simulators [Cou05]. However, from a practical viewpoint contact point tracking induces book keeping. It would be more attractive if sliding errors could be determined without the need for contact point tracking. We speculate this might be possible by examining the relative motion of objects projected onto the contact plane. If such a motion can be observed then sliding behaviour seems reasonable. Also It would be interesting to be able to distinguish rolling behaviour from sliding behaviour. We speculate that one may apply some fast heuristics under the assumption that all contact points in the contact set between two objects are subject to the same friction state. This is obviously not true in the general case, but might lead to good behaviour in cases of simulating structured stacks.

Test Case	Solver type / Iterations	Error reduction
Baumgarte normal stabilization	NNCG/75	0.5
New normal stabilization	NNCG/75	not used
Baumgarte normal stabilization w.o. friction stabilization	PGS/12	0.5
Baumgarte normal stabilization w. friction stabilization	PGS/12	0.5

Table 1: *Parameter value settings used in our experiments. All experiments were run using a simulation time step of* 0.1 *seconds to provoke stability problems, and a collision envelope* $= 0.0625$. *In all cases where clamping of maximum values was applied we used a clamp magnitude of* 2.

4. Experiments and Results

Our simulator is implemented in Java and was run under extreme interactive conditions as shown in Table 1 using a one core on a 2.1 GHz CPU Duo Core. We have made use of two different kind of solvers, the nonlinear nonsmooth conjugate gradient (NNCG) solver and the projected Gauss–Seidel (PGS) solver.

To demonstrate the advantages of our new normal stabilization technique we have made a subjective evaluation of interactive simulations of the same setup using the traditional setup of Baumgarte stabilization and our new normal stabilization technique. Our results can be seen from Figure 3 and the supplementary video. We have tuned all parameters in the study to make all methods work in their best possible way. Parameter settings are summarized in Table 1. The Baumgarte normal stabilization method produces visible penetration artefacts as well as jittering effects and even causes simulation blow ups. Jittering is best observed in the supplementary movie. In comparison our new normal stabilization method does not produce these artefacts.

We have applied a similar approach to show advantages and disadvantages of the suggested friction stabilization method in this paper. However, a comparison with an alternative is impossible since to our knowledge no other friction stabilization method exists. Instead, we compare results of using friction stabilization with not using it. Figure 4 shows our results. Our work on friction stabilization is in our opinion very preliminary. A more elaborate study would be interesting to determine the range of usability of friction stabilization. However, at least for our wall building and stacking test cases shown in the supplementary movie, the advantage of friction stabilization is clearly shown.

5. Discussion and Conclusion

Stabilization is well known and thoroughly studied in computational contact mechanics, scientific computing and related fields. However, in computer graphics the requirement for interactivity and high fidelity in usability may result, as we have shown here, in a different stabilization approach.

We have presented a novel method for normal stabilization which is more appropriate for interactive simulation than the Baumgarte stabilization method. The computational cost of the presented method is no worse than that of the Baumgarte method. In addition, it should prove easy to incorporate our method into existing simulation frameworks.

We have embarked upon a new research problem of friction stabilization and shown preliminary results on these ideas. Although much work still needs to be done, we have demonstrated that the idea shows great potential for creating high fidelity interactive simulations. We believe that we have taken the first few steps down a new avenue for interesting work in interactive simulation.

References

[Bar95] BARAFF D.: Interactive simulation of solid rigid bodies. *IEEE Comput. Graph. Appl. 15*, 3 (1995), 63–75. 1

[BB88] BARZEL R., BARR A. H.: A modeling system based on dynamic constraints. In *SIGGRAPH '88: Proceedings of the 15th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1988), ACM, pp. 179–188. 1

[BS06] BENDER J., SCHMITT A.: Constraint-based collision and contact handling using impulses. In *CASA '06: Proceedings of the 19th international conference on computer animation and social agents* (Geneva, Switzerland, July 2006), pp. 3–11. 2

[Cou05] COUMANS E.: The bullet physics library. http://www.continuousphysics.com, 2005. 1, 4

[CP03] CLINE M., PAI D.: Post-stabilization for rigid body simulation with contact and constraints. *Proceedings of IEEE International Conference on Robotics and Automation (ICRA) 3* (September 2003), 3744 – 3751. 1

[Erl05] ERLEBEN K.: *Stable, Robust, and Versatile Multibody Dynamics Animation.* PhD thesis, Department of Computer Science, University of Copenhagen, Denmark, Mar. 2005. 1, 4

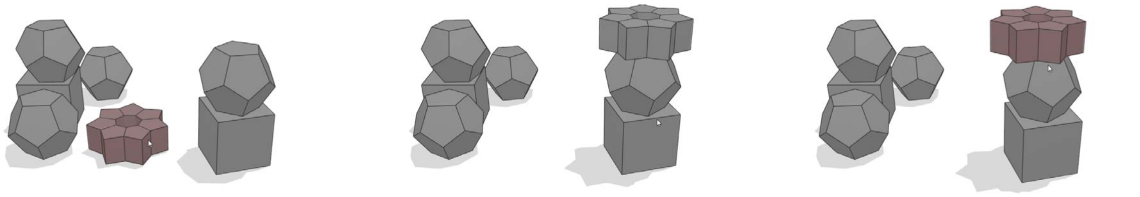
[Erl07] ERLEBEN K.: Velocity-based shock propagation for multibody dynamics animation. *ACM Trans. Graph 26*, 2 (June 2007). 1, 3

[GBF03] GUENDELMAN E., BRIDSON R., FEDKIW R.: Nonconvex rigid bodies with stacking. *ACM Trans. Graph. 22*, 3 (2003), 871–878. 2

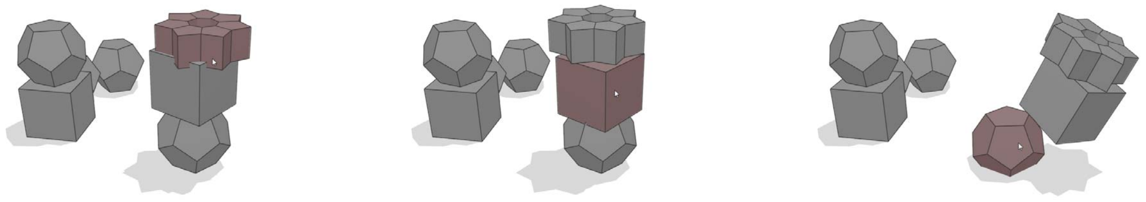
[Lac07] LACOURSIÉRE C.: Regularized, stabilized, variational methods for multibodies. In *SIMS '07: The 48th Scandinavian Conference on Simulation and Modeling* (October 2007), Linkøping Electronic Conference Proceedings, pp. 40–48. 2

[Mö8] MÜLLER M.: Hierarchical position based dynamics. In *VRIPHYS '08: Proceedings of Virtual Reality Interaction and Physical Simulation* (Nov. 2008). 1

[MHHR07] MÜLLER M., HEIDELBERGER B., HENNIX M., RATCLIFF J.: Position based dynamics. *J. Vis. Comun. Image Represent. 18*, 2 (2007), 109–118. 1

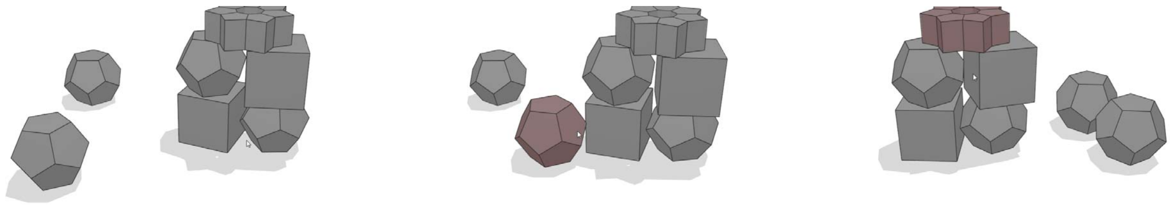


(a)

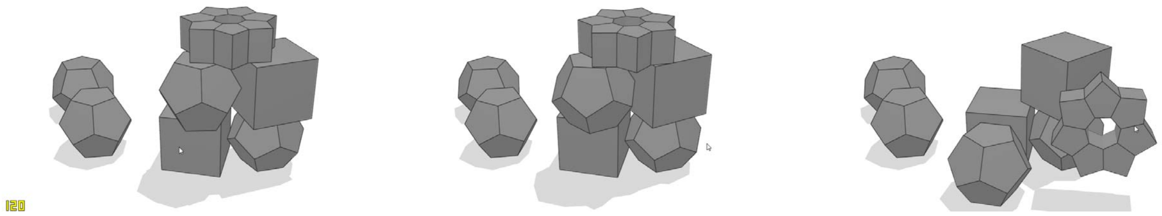


(b)

Figure 3: *Two frame grab sequences (from right to left) of interactive simulations showing a comparison of (a) our normal stabilization method with (b) Baumgarte stabilization method. Observe, when using our method there is no simulation blow ups nor is any jittering noticeable (best seen in the supplementary movie).*



(a)



(b)

Figure 4: *Two frame grab sequences (from right to left) of interactive simulations showing a comparison showing the added benefit of (a) using friction stabilization and (b) not using it. Clearly friction stabilization removes the unwanted sliding effects seen in the two first frames. The unwanted sliding results in a simulation blow up in the last frame. The supplementary movie illustrates this better.*

[Mir98] MIRTICH B.: *Rigid Body Contact: Collision Detection to Force Computation*. Tech. Rep. TR98-01, MITSUBISHI ELECTRIC RESEARCH LABORATORIES, December 1998. 4

[PNE10] POULSEN M., NIEBE S., ERLEBEN K.: Heuristic convergence rate improvements of the projected gauss-seidel method for frictional contact problems. In *Proceedings of WSCG* (2010). 1

[Smi00] SMITH R.: Open dynamics engine. http://www.ode.org, 2000. 1

[SNE10a] SILCOWITZ M., NIEBE S., ERLEBEN K.: Contact point generation for convex polytopes in interactive rigid body dynamics. SCA '10: Booklet with Poster presentation, The ACM SIGGRAPH / Eurographics Symposium on Computer Animation, Madrid, Spain, June 2010. 1

[SNE10b] SILCOWITZ M., NIEBE S., ERLEBEN K.: A non–smooth nonlinear conjugate gradient method for interactive contact force problems. *The Visual Computer Journal* (2010). 1

[SNE10c] SILCOWITZ M., NIEBE S., ERLEBEN K.: Projected gauss-seidel subspace minimization method for interactive rigid body dynamics. In *GRAPP '10: Proceedings of the Fifth International Conference on Computer Graphics Theory and Applications* (Angers, France, May 2010), INSTICC Press. 1

[SSF08] SHINAR T., SCHROEDER C., FEDKIW R.: Two-way coupling of rigid and deformable bodies. In *SCA '08: Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Aire-la-Ville, Switzerland, Switzerland, 2008), Eurographics Association, pp. 95–103. 2

[SSF09] SU J., SCHROEDER C., FEDKIW R.: Energy stability and fracture for frame rate rigid body simulations. In *SCA '09: Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (New York, NY, USA, 2009), ACM, pp. 155–164. 2

[WTF05] WEINSTEIN R., TERAN J., FEDKIW R.: Pre-stabilization for rigid body articulation with contact and collision. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Sketches* (New York, NY, USA, 2005), ACM, p. 79. 2