Adaptive Sub-stepping for Constrained Rigid Body Simulations

Chris Giles
Roblox
USA
cgiles@roblox.com

Sheldon Andrews
École de technologie supérieure
Canada
Roblox
USA
sheldon.andrews@etsmtl.ca

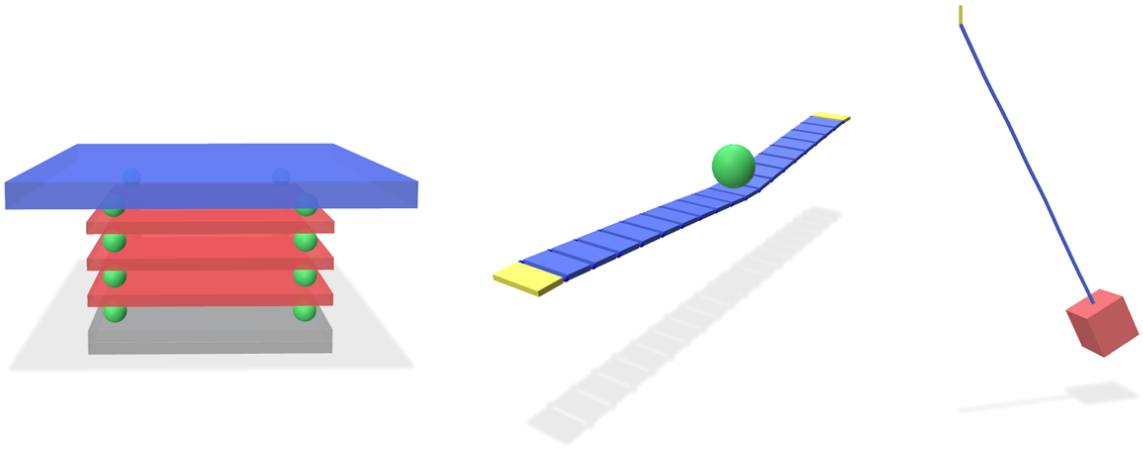

Figure 1: A heavy box sits atop a tower of spheres and thin plates (left); a boulder rolls across a swinging bridge (middle); a massive box is suspended from a chain of rigid bodies (right). Each scenario involves mass ratios of 10,000:1, 1000:1, and 1000:1, respectively. Our proposed adaptive sub-stepping scheme allows efficient and stable simulation of these challenging examples.

Abstract

Achieving stable simulation of constrained rigid body systems is a primary concern for many computer graphics applications, such as video games, robotic planning, and virtual reality training. In this paper, we present a novel adaptive sub-stepping scheme that achieves stable simulation by adaptively reducing the time step as needed. Our approach employs a diagonalized geometric stiffness matrix as a heuristic to determine when smaller time steps are required, and adjusts the number of sub-steps accordingly. Our method is straightforward to integrate into existing rigid body simulators, and further eliminates manually tuning the number of sub-steps required. We demonstrate the ability of our method to produce stable simulates at real-time frame rates using a number of challenging, complex examples.

CCS Concepts

• **Computing methodologies** → **Real-time simulation**; **Interactive simulation**; **Physical simulation**.

MIG '24, Nov. 21-23, 2024, Arlington, VA

Keywords

geometric stiffness, sub-stepping, rigid body dynamics, constraints, physics-based animation

ACM Reference Format:
Chris Giles and Sheldon Andrews. 2024. Adaptive Sub-stepping for Constrained Rigid Body Simulations. In *The 16th ACM SIGGRAPH Conference on Motion, Interaction, and Games (MIG '24), November 21–23, 2024, Arlington, VA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3677388.3696331

1 Introduction

Physics simulation is a mainstay of interactive computer graphics. In particular, constrained rigid body simulation has seen widespread adoption in video games and virtual reality applications. A primary objective of simulations in this setting is to maintain real-time framerates, while also remaining robust to user inputs and a wide variety of simulation parameters. There has thus been a large body of work that focuses on efficient methods for producing stable simulations.

One popular technique proposed in recent years is that of sub-stepping [Macklin et al. 2019b], which breaks down each visual step into N simulation time steps. This strategy not only improves accuracy, e.g., collision detection for fast moving objects, but also improves stability due to more frequent linearization of the non-linear system modeling dynamics.

Other approaches have focused on improving stability by explicit damping [Cline 2002]. Damping improves the conditioning of stiff systems, allowing semi-implicit and even explicit integrators to be used. The non-linear constraints that characterize many rigid body simulations may introduce damping by a Baumgarte-type stabilization, which effectively introduces feedback of the constraint error and generates constraint forces from implicit damped springs. However, Tournier et al. [2015] observed that additional instabilities due to geometric non-linearities that are ignored by many first-order integrators are a principle cause of instability for constrained simulations. Including the geometric stiffness, a first-order approximation of constraint force changes, helps to improve stability. Andrews et al. [2017] later proposed using a diagonalized version of the geometric stiffness to compute an adaptive damping term. Their approach was based on a stability analysis of a first-order integrator that computed just enough damping to meet the stability criterion. [Macklin et al. 2019a] found that this diagonal approximation also acts as a good preconditioner for a non-smooth Newton method. They compute the geometric stiffness using finite differences, which avoids explicit computation of the matrix. They found this geometric stiffness approximation to be quite effective on particle-based objects, but less so on rigid articulated mechanisms, and in the worst case can cause some jitter at joint limits.

In this paper, we propose a technique that combines the strategy of sub-stepping with a stability analysis based on the geometric stiffness. Specifically, an adaptive sub-stepping technique is demonstrated that shows improved stability compared to a baseline simulator, improved performance compared to previous sub-stepping methods, and improved energy conservation versus adaptive damping techniques. We demonstrate our method with a number of challenging constrained rigid body simulations involving high mass ratios and complex configurations of joints and contacts.

2 Background and Related Work

The report by Bender et al. [2014] provides an excellent background on the simulation of rigid bodies with constraints. Here, we briefly introduce the fundamentals of rigid body simulation and provide the lead-up to our adaptive sub-stepping scheme.

2.1 Constrained Rigid-body Dynamics

The Newton-Euler equations $\mathbf{f}(\mathbf{q}(t), \mathbf{u}(t)) = \mathbf{M}(t)\dot{\mathbf{u}}(t)$ describe the dynamical behavior of a system at time t with forces $\mathbf{f}(\mathbf{q}(t), \mathbf{u}(t))$ acting on a collection of bodies with generalized masses $\mathbf{M}(t)$, positions $\mathbf{q}(t)$, and velocities $\mathbf{u}(t)$. Introducing a time step h and approximating $\mathbf{u}(t + \Delta t) \approx \mathbf{u}(t) + h\dot{\mathbf{u}}(t)$, the equations can be discretized and a numerical integrator used to advance the state of the system, giving the familiar velocity level equations of motion:

$$\mathbf{M}\mathbf{u}^+ = \mathbf{M}\mathbf{u} + h\mathbf{f}\,.$$

We use $\mathbf{u}^+$ to denote values at the end of a time step, and assume that the mass of bodies does not change, giving constant mass matrix $\mathbf{M}$.

Kinematic constraints couple the motion of rigid bodies making it possible to model the behavior of real-world joints, e.g., hinges, ball-and-socket, and even contact. A constraint equation of the form $\phi(\mathbf{q}) = 0$ implicitly defines the permissible configurations for the degrees of freedom $\mathbf{q}$ for a system of rigid bodies. While position-based frameworks for simulating rigid bodies have been proposed [Müller et al. 2020], it is more common to resolve the constraint equations at the velocity level, such that

$$\mathbf{J}\mathbf{u} = -\frac{1}{h}\phi\,,$$

where $\mathbf{J} = \frac{\partial \phi(\mathbf{q})}{\partial q}$ is the gradient of the constraint function $\phi(\mathbf{q})$, and $-\frac{1}{h}\phi$ on the right-hand side is a Baumgarte-style stabilization[Baumgarte 1972] term. Note that ϕ is sometimes also referred to as the constraint error.

Constraints are enforced in the dynamical system through the inclusion of impulses with magnitude λ^+ acting in the direction of the constraint gradient, $\mathbf{J}^T\lambda^+$, giving the revised equations of motion:

$$\mathbf{M}\mathbf{u}^+ - \mathbf{J}^T\lambda^+ = \mathbf{M}\mathbf{u} + h\mathbf{f}\,.$$

Simulations involving contact also require solving a mixed linear complementarity problem (MLCP), since the system may include both bilateral and unilateral constraints. Our rigid body simulations use a box Coulomb friction model and follow the contact formulation outlined by Andrews et al. [2022].

2.2 Stable Constrained Simulations

Observe that the constraint forces are generated by a linearization of ϕ, which does not account for changes in the constraint force direction between time steps. This can lead to simulation instabilities, in particular when ϕ represents a stiff potential. However, Tournier et al. [2015] showed that by including the *geometric stiffness*, even in its explicit form, simulation stability is significantly improved. The geometric stiffness is a tensor encoding variations in the constraint force directions, and has the form

$$\tilde{\mathbf{K}} = \frac{\partial \mathbf{J}^T}{\partial \mathbf{q}}\lambda^+\,. \tag{1}$$

Tournier et al. [2015] proposed to use an explicit version of Eq. 1 that is computed using constraint impulses from the start of the time step, or $\tilde{\mathbf{K}} = \frac{\partial \mathbf{J}^T}{\partial \mathbf{q}}\lambda$. We use this form of the geometric stiffness in our experiments and include it as a stiffness in the velocity-level discretization of the constrained dynamical equations, such that

$$\begin{bmatrix} \mathbf{M} - h^2\tilde{\mathbf{K}} & -\mathbf{J}^T \\ \mathbf{J} & \frac{1}{h^2}\mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{u}^+ \\ h\lambda^+ \end{bmatrix} = \begin{bmatrix} \mathbf{M}\mathbf{u} + h\mathbf{f} \\ -\frac{1}{h}\phi \end{bmatrix}, \tag{2}$$

where the diagonal matrix $\mathbf{C}$ contains the compliance (inverse stiffness) of each scalar constraint. Forming the Schur complement of the upper left block in Eq. 2 gives the reduced system

$$\left(\mathbf{J}\tilde{\mathbf{M}}^{-1}\mathbf{J}^T + \frac{1}{h^2}\mathbf{C}\right) h\lambda^+ = -\frac{1}{h}\phi - \mathbf{J}\tilde{\mathbf{M}}^{-1}(\mathbf{M}\mathbf{u} + h\mathbf{f})\,, \tag{3}$$

where the augmented mass matrix is $\tilde{\mathbf{M}} = \mathbf{M} - h^2\tilde{\mathbf{K}}$.

Andrews et al. [2017] noted that inclusion of the geometric stiffness in Eq. 3 leads to numerical difficulties when solving the linear system. Mainly, that symmetry and positive definiteness of the lead matrix are not assured. They proposed to diagonalize the $\tilde{\mathbf{K}}$ matrix, where each diagonal element $\mathbf{K}_{\mathrm{d}i,i}$ is computed as the 2-norm of the corresponding column $\tilde{\mathbf{K}}_i$ of the geometric stiffness, such that

$$\mathbf{K}_{\mathrm{d}i,i} = \|\tilde{\mathbf{K}}_i\|\,. \tag{4}$$

This diagonalization facilitates a stability analysis of a semi-implicit integrator, where

$$\frac{h^2 \mathbf{K}_{\mathrm{d}i,i}}{\mathbf{M}_{i,i}} \leq 4\alpha \tag{5}$$

is the stability criterion, where $\alpha \in [0 \dots 1]$ is a positive scalar used to determine how close the system must be to the boundary before damping is applied. If the stability criterion in Eq. 5 is violated for any degree of freedom i, a damping coefficient $\mathbf{B}_{i,i}$ is computed as:

$$\mathbf{B}_{i,i} = \frac{h^2 \mathbf{K}_{\mathrm{d}i,i} - 4\alpha \mathbf{M}_{i,i}}{h} \,. \tag{6}$$

The augmented mass matrix in Eq. 3 is then assembled as

$$\tilde{\mathbf{M}} = \mathbf{M} + h\mathbf{B} \,,$$

where $\mathbf{B}$ is the global damping matrix with non-zero coefficient only for indices i where Eq. 5 indicates that damping is required.

3 Adaptive Sub-stepping

In the section, we explain how the stability criterion in Eq. 5 may be repurposed for performing adaptive sub-stepping, thus leading to our proposed algorithm. The stability criterion is derived from a stability analysis of a damped oscillator with stiffness k, damping b, and mass m. The criterion states that the eigenvalues of the update matrix of the integrator satisfy the inequality

$$\left| \frac{h^2 k + hb - 2m \pm \sqrt{h^4 k^2 + 2h^3 bk + h^2 b^2 - 4h^2 km}}{2m} \right| \leq 1 \,. \tag{7}$$

Observe that multiple parameters affect the simulation stability: time step size, mass, stiffness, and damping. Strategies to satisfy the criterion in Eq. 7 include lowering the stiffness, increasing the mass, adding damping, or reducing the time step. Servin et al. [2011] derived a similar stability threshold that allowed them to adaptively refine the resolution of a cable simulation.

While Andrews et al. [2017] noted that their adaptive damping scheme produced very stable simulations, it also sometimes produced artifacts in the dynamic behavior. Particularly when applied to translational degrees of freedom, objects appeared to move through a viscous fluid. Therefore, they suggested only applying damping to the rotational degrees of freedom. However, damping ultimately changes the physical behavior of the simulation and introduces additional energy dissipation in the simulation that may be undesired. We provide further analysis on this point in the results section. Increasing the mass and stiffness similarly changes the dynamical behavior.

Alternatively, stability may be achieved by reducing the time step. Macklin et al. [2019b] explored this strategy and proposed an integration scheme that uses more sub-steps and fewer solver iterations to achieve improved stability and reduced numerical dissipation. Notably, they demonstrate that a sub-stepping strategy can better preserve the kinetic energy present in the system.

Our approach is inspired by both the work by Andrews et al. [2017] and Macklin et al. [2019b], where an adaptive sub-stepping scheme is guided by a stability analysis of the numerical integrator. To this end, we use the criterion in Eq. 5 to determine if additional sub-steps are required for stability over the next time step h:

Algorithm 1 Geometric Stiffness Guided Adaptive Time-stepping

```
compute K_d using q, u                                   ▷ Eq. 4
r ← 0
for each degree of freedom i do
    r ← max(r, K_di,i / M_i,i)                           ▷ Eq. 9
end for
h̄ = 2√(α/r)                                              ▷ Eq. 10
N_steps = ⌈h/h̄⌉                                          ▷ Eq. 11
Δt = h/N_steps

while n < N_steps do
    perform collision detection using q, u
    q⁺, u⁺ ← solve(q, u, Δt)
    n ← n + 1
end while
```

$$h \leq 2\sqrt{\alpha \frac{\mathbf{M}_{i,i}}{\mathbf{K}_{\mathrm{d}i,i}}} \,. \tag{8}$$

Solving Eq. 5 for h gives the maximum time step for the ith degree of freedom that meets the stability threshold. It suffices then to compute the largest inverse ratio across all n constrained degrees of freedom:

$$r = \max_{i \in n} \frac{\mathbf{K}_{\mathrm{d}i,i}}{\mathbf{M}_{i,i}} \,. \tag{9}$$

The stability analysis then gives us the maximum stable time step $\bar{h}$ for some stability threshold α:

$$\bar{h} = 2\sqrt{\frac{\alpha}{r}} \,. \tag{10}$$

The number of sub-steps needed for a stable simulation is then the ceiling of the ratio of the target time step h and the maximum stable time step $\bar{h}$:

$$N_{\text{steps}} = \left\lceil \frac{h}{\bar{h}} \right\rceil \tag{11}$$

We summarize the method in Algorithm 1. Note that while Eq. 2 is solved for the constraint impulses and velocities at the next time step, our method is compatible with a variety of integration and constraint solving methods as long as they accept a variable time step and can produce constraint forces λ used for computing $\tilde{\mathbf{K}}$.

4 Results

We evaluate our adaptive sub-stepping approach using the example scenes shown in the teaser image (see Figure 1), which are known to cause instability. A time step of $t = \frac{1}{60}$ s is used for each simulation frame. Furthermore, a custom rigid body framework featuring collision detection, contact handling, articulated joints, and various solver algorithms is used to simulate the example scenes.

The baseline comparison in our experiments is to execute the solver algorithm using a single sub-step and without additional damping. Two solvers are used in our experiments: (i) projected Gauss-Seidel (PGS) and (ii) block principal pivoting (BPP) algorithm [Júdice and Pires 1994] with Cholesky factorization (direct). We compare our adaptive approach against a sub-stepping scheme

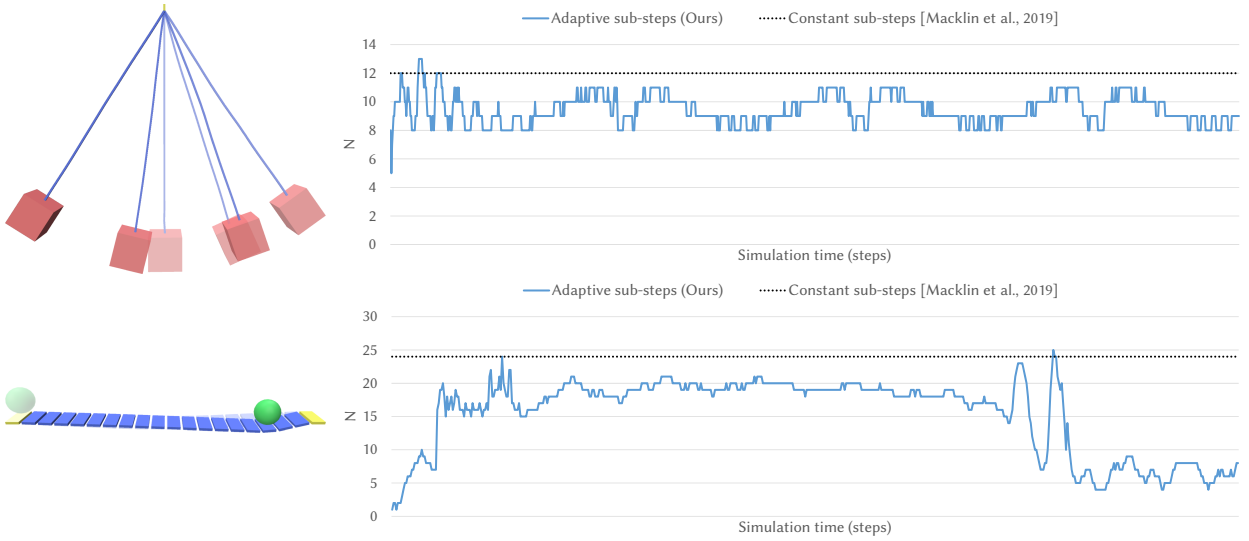

Figure 2: Comparison of the number of sub-steps using our adaptive approach (blue) versus the constant N (dashed) proposed in previous work [Macklin et al. 2019b]. In these experiments, N (constant sub-steps) is tuned such that the total constraint error is below a fixed threshold for each simulation frame. Stability threshold $\alpha = 0.1$ is used in all of our experiments.

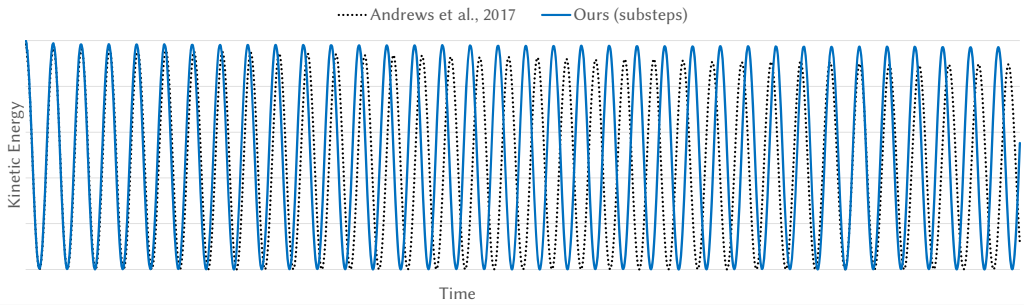

Figure 3: Kinetic energy computed for the Swinging Box example over 200 s of simulation time. The adaptive damping technique proposed by [Andrews et al. 2017] causes kinetic energy to gradually dissipate, whereas adaptive sub-stepping better preserves kinetic energy of the system.

where the number of sub-steps is determined by manually adjusting N to give good behavior. This is effectively equivalent to the technique proposed by Macklin et al. [2019b].

The supplementary video contains complete animations obtained from simulations of each example scene. A PGS solver with a maximum of 25 iterations is used for the Swinging Box and Bridge examples, whereas a BPP solver with 20 pivoting steps is used for the Heavy Stack example.

4.1 Examples

4.1.1 Swinging Box. A 25-link chain is modeled using hinge constraints and supports a large, heavy box. The mass ratio here is 1000:1 and both the PGS and the BPP solvers work for this scene, but explode without additional sub-steps or damping added. Running with a single sub-step and no damping applied, the chain is immediately unstable and falls through the floor, even with a direct solver. The chain swings stably using the method of Andrews et al. [2017], albeit with slightly damped behavior. This is evident from the kinetic energy of the system, which slowly decreases with the adaptive damping stabilization (see Figure 3). Using our proposed approach, the chain swings stably and conserves energy very well. The number of sub-steps our method uses is greatest at the bottom of the swing, since the the forces are greatest at that point, and reduce as the chain swings upward, which can be seen in the upper row of Figure 2

4.1.2 Bridge. A series of thin planks connected by pairs of ball-and-socket joints is used to model a swinging bridge. The large ball rolling through the scene has mass of 1000 kg, giving a mass ratio of 1000:1. A Coulomb coefficient of $\mu = 0.8$ is used to simulate friction between the ball and planks. This scene demonstrates the stabilizing abilities of our approach with a combination of joints and contact when subjected to high mass ratios. Simulation with a single sub-step and no damping, the bridge quickly jitters apart

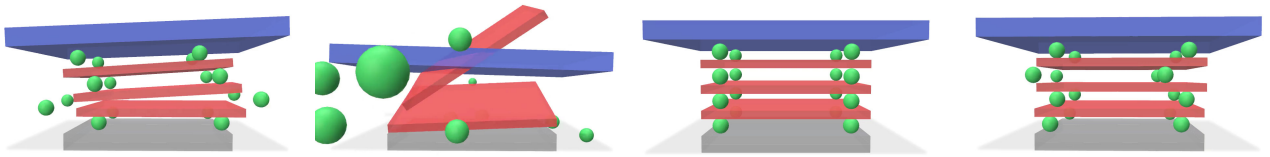

Figure 4: Selected frames from the Heavy Stack example simulated using various solvers and stabilization methods. Only the BPP solver with Cholesky factorization is able to maintain a stable configuration for this high mass ratio example, and only using our adaptive sub-stepping approach. Even with adaptive damping, the balls in between the plates begin to slide, indicating that numerical errors and instabilities begin to affect the behavior.

and cannot support the weight of the ball. The damping approach of Andrews et al. [2017] works well and is able to maintain stability as the ball rolls across, although there is some notable jitter in the planks as it does so. The proposed adaptive time-stepping method is stable, maintains energy well and does not suffer from increased jitter. The number of sub-steps used by our adaptive technique automatically reduces once the ball falls off the bridge, which can be seen in the lower row of Figure 2.

4.1.3 Heavy Stack. A stack of balls and thin plates, where all bodies have mass of 1 kg except for the top most box that has mass of 10,000 kg. There is no friction in this example, since friction is a dissipative element in the simulation that could contribute to stabilization. Only the BPP solver is able to simulate this example. This is likely due to the fact that warm-starting iterative solvers with contact is not implemented by our framework. The PGS solver is thus not able to converge enough to maintain stability, even with hundreds of iterations. However, even the direct solver struggles here. The tower quickly collapses without additional stabilization or smaller time-steps. The damping method of Andrews et al. [2017] fails to maintain stability for very long. Our proposed approach gives enough stability such that the stack remains standing, even after 10 s of simulation time. The supplementary video also shows a taller version of this example, with 20,000:1 mass ratio. However, setting $\alpha = 0.05$ was required to obtain this more challenging result.

4.2 Implementation Details

One important detail about our implementation of Macklin et al. [2019b] is that collision detection is performed at each sub-step. This is because our simulation engine does not feature predictive collision handling, which allowed the previous work to perform collision detection only once per frame. Another important difference is that we keep the number of PGS iterations fixed per sub-step, where Macklin et al. [2019b] proposed to keep constant $N_{\text{steps}}\times$ the number of PGS iterations. Rather, for our comparisons, we tune the number of PGS iterations such that the total constraint error, i.e. $\sum_i \|\phi_i(\mathbf{q})\|_1$, remains below a small threshold in our experiments.

Mathematical details about the geometric stiffness matrices used for our experiments can be found in the supplementary appendix. Andrews et al. [2017] provide similar mathematical details. We additionally propose a novel geometric stiffness for contacts, specifically non-interpenetration constraints, which is a modified version of the geometric stiffness for ball-and-socket constraints.

5 Conclusion

In this paper, we have proposed a new take on sub-stepping for constrained rigid body simulation. Rather than manually tuning the number of sub-steps in order to achieve stability and accuracy for each different simulation, we instead analyze the problem based on the stability criterion of the numerical integrator. This simplifies the process of tuning the sub-step parameter, and we demonstrate that the quality of results achieved with our method is comparable, or better, to prior work on this topic. Additionally, since the number of sub-steps is computed on a per frame basis, simulations can be executed more efficiently. Our method is also agnostic to the specific numerical method used to solve the dynamical equations, and thus we believe many physics engines could benefit from our approach.

One limitation of our work is that, depending on the value of α, a large number of sub-steps may be invoked, which is computationally costly. We found that $\alpha = 0.1$ worked well for all the examples presented in this paper. However, selecting the number of sub-steps will increase proportional to decreases in α, e.g. halving the parameter will double the number of time steps, although we did not find this to be problematic. A possible extension of our algorithm is to combine the adaptive sub-stepping and damping approaches, whereby sub-steps are increased until a maximum threshold N_{max}, at which point adaptive damping is for a time step of $\frac{h}{N_{\text{max}}}$. We leave this for future work.

Acknowledgments

We thank Eric Dow, Nick Burgess, Peter Cheng and Kenny Erleben for many illuminating discussions on numerical integration and rigid body simulations.

References

Sheldon Andrews, Kenny Erleben, and Zachary Ferguson. 2022. Contact and friction simulation for computer graphics. In *ACM SIGGRAPH 2022 Courses* (Vancouver, British Columbia, Canada) *(SIGGRAPH '22)*. Association for Computing Machinery, New York, NY, USA, Article 3, 172 pages. https://doi.org/10.1145/3532720.3535640

Sheldon Andrews, Marek Teichmann, and Paul G. Kry. 2017. Geometric Stiffness for Real-time Constrained Multibody Dynamics. *Computer Graphics Forum* 36, 2 (2017), 235–246. https://doi.org/10.1111/cgf.13122

J. Baumgarte. 1972. Stabilization of constraints and integrals of motion in dynamical systems. *Computer Methods in Applied Mechanics and Engineering* 1, 1 (1972), 1–16. https://doi.org/10.1016/0045-7825(72)90018-7

Jan Bender, Kenny Erleben, and Jeff Trinkle. 2014. Interactive Simulation of Rigid Body Dynamics in Computer Graphics. *Computer Graphics Forum* 33, 1 (2014), 246–270. https://doi.org/10.1111/cgf.12272

Michael Bradley Cline. 2002. *Rigid body simulation with contact and constraints.* Ph. D. Dissertation. University of British Columbia.

Joaquim J. Júdice and Fernanda M. Pires. 1994. A block principal pivoting algorithm for large-scale strictly monotone linear complementarity problems. *Computers & Operations Research* 21, 5 (1994), 587–596. https://doi.org/10.1016/0305-0548(94)90106-6

Miles Macklin, Kenny Erleben, Matthias Müller, Nuttapong Chentanez, Stefan Jeschke, and Viktor Makoviychuk. 2019a. Non-smooth Newton Methods for Deformable Multi-body Dynamics. *ACM Trans. Graph.* 38, 5, Article 140 (oct 2019), 20 pages. https://doi.org/10.1145/3338695

Miles Macklin, Kier Storey, Michelle Lu, Pierre Terdiman, Nuttapong Chentanez, Stefan Jeschke, and Matthias Müller. 2019b. Small steps in physics simulation. In *Proceedings of the 18th Annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Los Angeles, California) *(SCA '19)*. Association for Computing Machinery, New York, NY, USA, Article 2, 7 pages. https://doi.org/10.1145/3309486.3340247

Matthias Müller, Miles Macklin, Nuttapong Chentanez, Stefan Jeschke, and Tae-Yong Kim. 2020. Detailed Rigid Body Simulation with Extended Position Based Dynamics. *Computer Graphics Forum* 39, 8 (2020), 101–112. https://doi.org/10.1111/cgf.14105

Martin Servin, Claude Lacoursière, Fredrik Nordfelth, and Kenneth Bodin. 2011. Hybrid, Multiresolution Wires with Massless Frictional Contacts. *IEEE Transactions on Visualization and Computer Graphics* 17, 7 (2011), 970–982. https://doi.org/10.1109/TVCG.2010.122

Maxime Tournier, Matthieu Nesme, Benjamin Gilles, and François Faure. 2015. Stable constrained dynamics. *ACM Trans. Graph.* 34, 4, Article 132 (jul 2015), 10 pages. https://doi.org/10.1145/2766969