Introduction to the research paper and how the algorithm is going to work with the system

1. Introduction

Rigid body simulations with contact is widely used in different areas. For example, in Robotics, simulations are used in processes such as virtual prototyping, robot testing, model predictive control, and planning. These simulations are mostly implemented using a time-stepping scheme. In this scheme, the simulation finds out the changes that will happen to the bodies in the system at the current time frame and then integrates the changes to the current configuration to proceed to the next time step. One of the problems of this approach is that numerical error cumulates during the process, and bodies start to interpenetrate when the error is large enough.

Despite the fact that such interpenetration causes simulations to behave unrealistically, the time stepping approaches for rigid body simulations do not even attempt to prevent bodies from interpenetrating due to the high difficulty of preventing them. Instead, people are using algorithm to correct interpenetrations after they occur.

Currently, People mainly correct interpenetrations by adding extra to push interpenetrating bodies apart. The strength of the force is dependent on how much the interpenetration is. The following diagrams shows an example of this approach. In *Diagram 1.1*, one of the bodies is lying on top of the other. In this case, a force F_x is

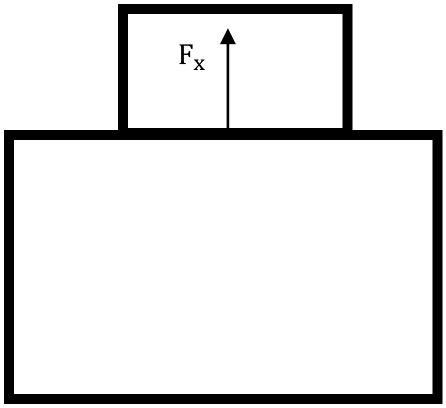


Diagram 1.1

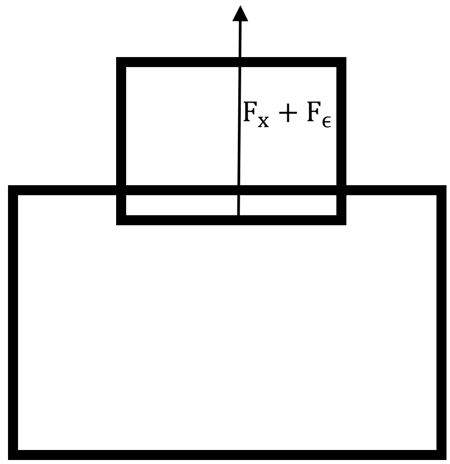
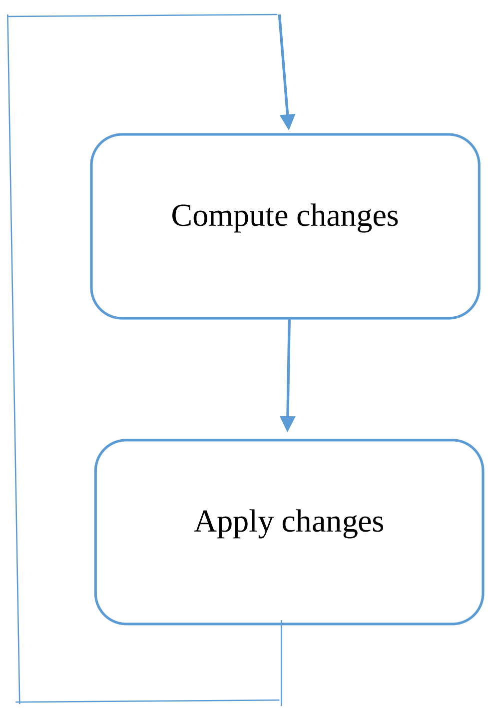


Diagram 1.2

applied to keep the two bodies separated. In Diagram *1.2*, the two bodies are interpenetrating, therefore an extra force F_ϵ is also applied to push away the two bodies. The main advantages of this algorithm are its simplicity, time efficiency, and its ability to yield the proper result as the integration step size tends to zero; however, this algorithm also has several downsides. First, this algorithm has preset constants and therefore requires tuning while running, and there is no automatic way to do this tuning. Second, since we are just adding forcing on the bodies when they interpenetrate, we are allowing the simulation to be in an inadmissible configuration (bodies violating the assumption of rigidity) at some times, and due to the fact that bodies always then to contact with each other in these simulations, the simulation will often be in an inadmissible configuration. Third, this algorithm will decrease the stability and accuracy of the system due to adding extra forces to the system.

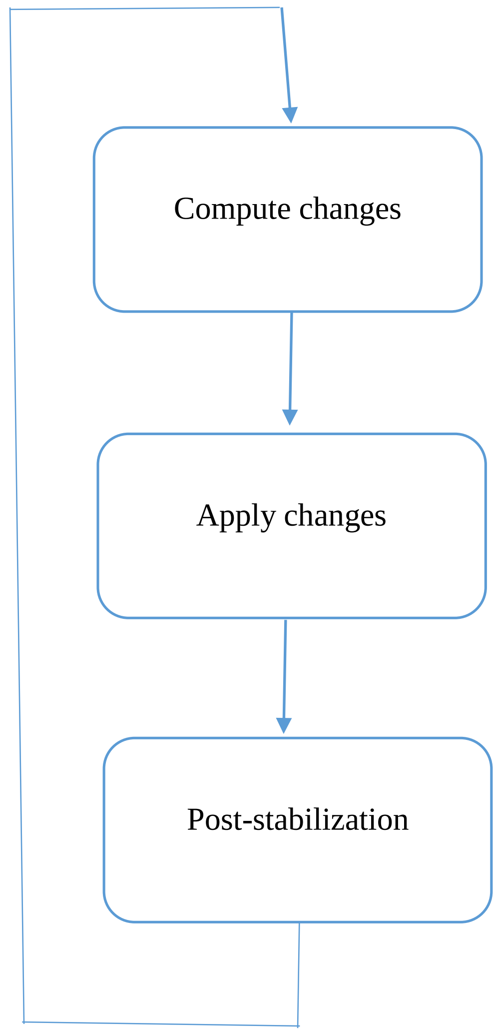
Since these downsides can be greatly problematic to simulations, another interpenetration correction method called post-stabilization has been developed. In this method bodies are moved apart directly after every step. Although this method is slower than the one mentioned in the last paragraph, it addresses all the three disadvantages mentioned in the previous paragraph. No tuning constant is involved in the process. The bodies are always in admissible configurations at the end of a step. And the stability of the system is also less affected because less energy is added to the system. This paper will discuss post-stabilization for convex bodies represented by polyhedra -- including the effectiveness, limitations and performance – using both pathological and realistic examples.

The post-stabilization algorithm is implemented under Moby, a multi-rigid body dynamics simulator in time-stepping scheme. The following two diagram shows how this algorithm is going to be implemented into the system.



Original Flow of Moby's Simulation

Originally, Moby separates the entire simulation process into small steps. During every step, it calculates the changes of the current configuration, applies the change to the current configuration, and goes to the next step and repeats the process again, as shown in the diagram on the left.



Modified Flow of Moby's Simulation

To implement post-stabilization, an extra function is called at the end of every step of the simulation. Now before the simulation moves on to the next step, the configurations of the bodies are fed into the post-stabilization function. The function then checks if the configuration is inadmissible. If it is, it will execute the post-stabilization algorithm and move the bodies apart. After the execution, all the bodies configurations should be admissible, i.e. there exist no interpenetration inside the system. The simulation will then proceed to the next step and repeats the process mentioned in this and the previous paragraph.