Non-penetration modeling error in physical simulation time-steppers

A more accurate and robust method

Binh Nguyen, Jeff Trinkle

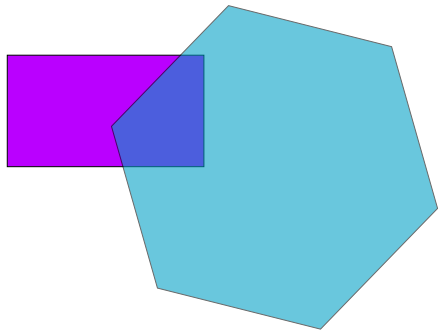
Computer Science Department,RPI

May 29, 2008

Building block of physical simulation

Common physical simulations follow the same set of rules (models)

- $force = mass \times acceleration$.
- Non-penetration and rigid body laws
- Friction law



Current physical simulation methods

There are many physical simulation libraries available

- Havok (Intel), Ageia (NVIDIA), ODE (open source), Bullet (open source),CHRONO (free for noncommercial), TrueAxis (free for noncommercial),...

Current physical simulation methods

There are many physical simulation libraries available

- Havok (Intel), Ageia (NVIDIA), ODE (open source), Bullet (open source),CHRONO (free for noncommercial), TrueAxis (free for noncommercial),...
- Most of them are created for games, virtual reality.

Current physical simulation methods

There are many physical simulation libraries available

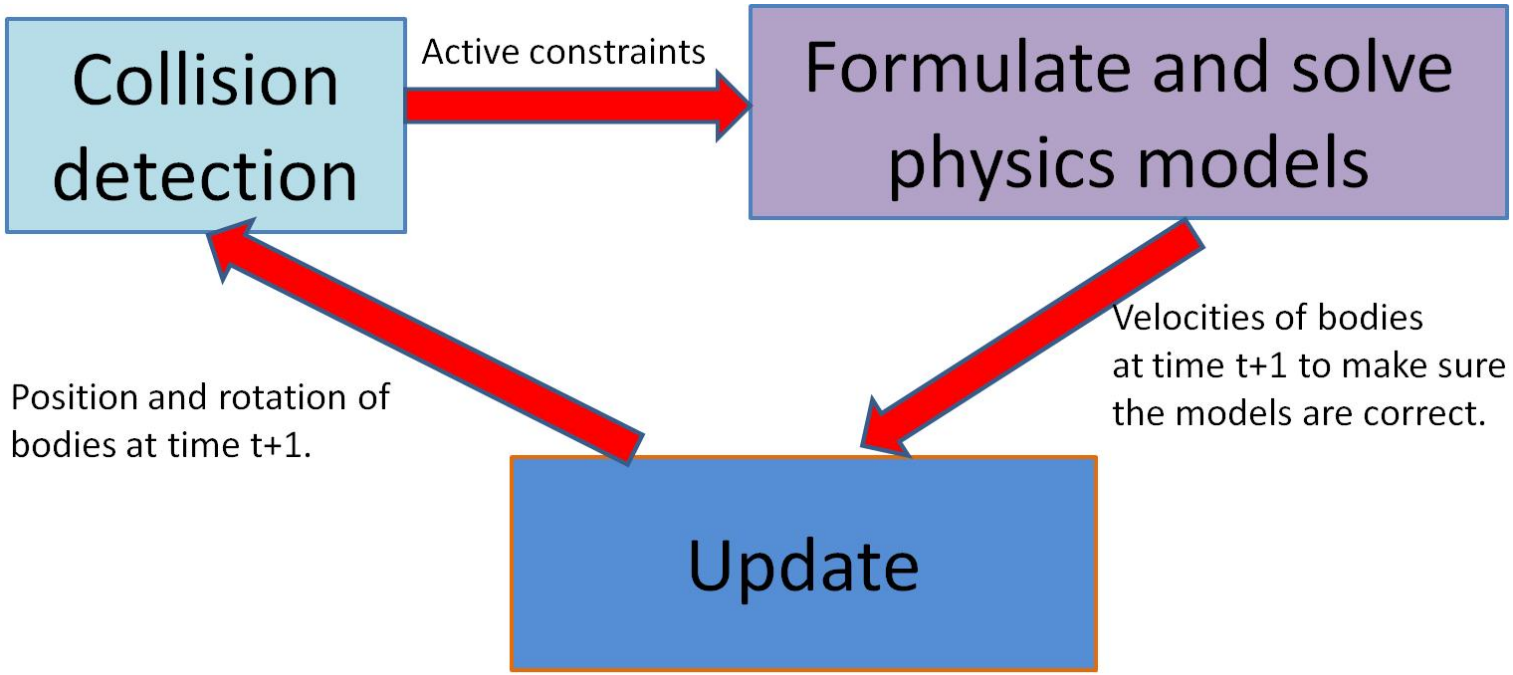
- Havok (Intel), Ageia (NVIDIA), ODE (open source), Bullet (open source),CHRONO (free for noncommercial), TrueAxis (free for noncommercial),...
- Most of them are created for games, virtual reality.
- Many robotics researchers are using them and having great difficulty to get satisfactory accuracy.

Current physical simulation methods

There are many physical simulation libraries available

- Havok (Intel), Ageia (NVIDIA), ODE (open source), Bullet (open source),CHRONO (free for noncommercial), TrueAxis (free for noncommercial),...
- Most of them are created for games, virtual reality.
- Many robotics researchers are using them and having great difficulty to get satisfactory accuracy.
- Simulation quality depends heavily on collision detection.

Normal simulation steps



Constraint in 2D consists of a pair of edge and vertex.
In step two, we model the system as Linear Complimentarily Problem.

Non-penetration constraint

There are two main types of method to model non-penetration constraint:

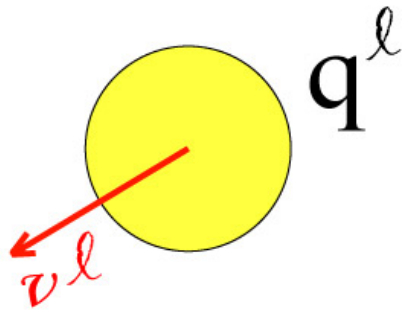
- Type 1: Correcting penetration (most current physics engines)

Non-penetration constraint

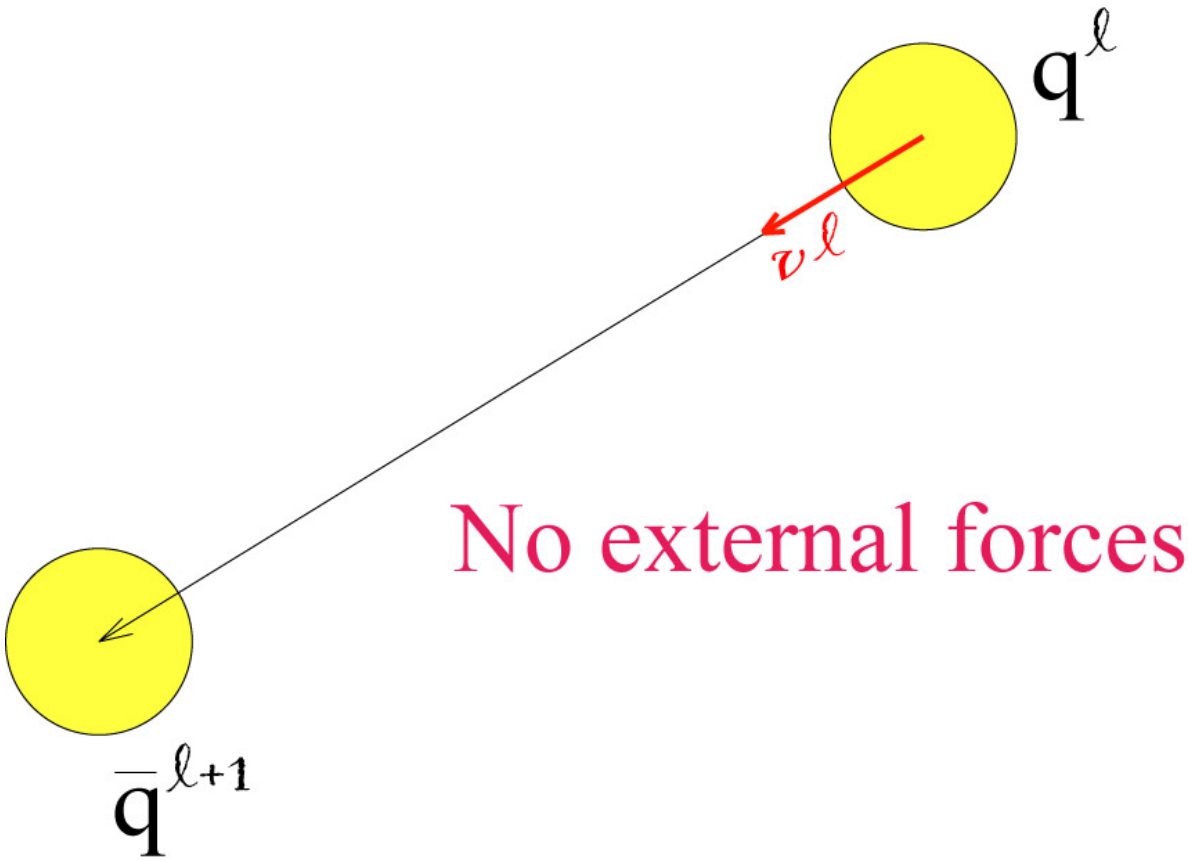
There are two main types of method to model non-penetration constraint:

- Type 1: Correcting penetration (most current physics engines)
- Type 2: Preventing penetration (dvc2D, a 2D physics engine developed by RPI and UPenn)

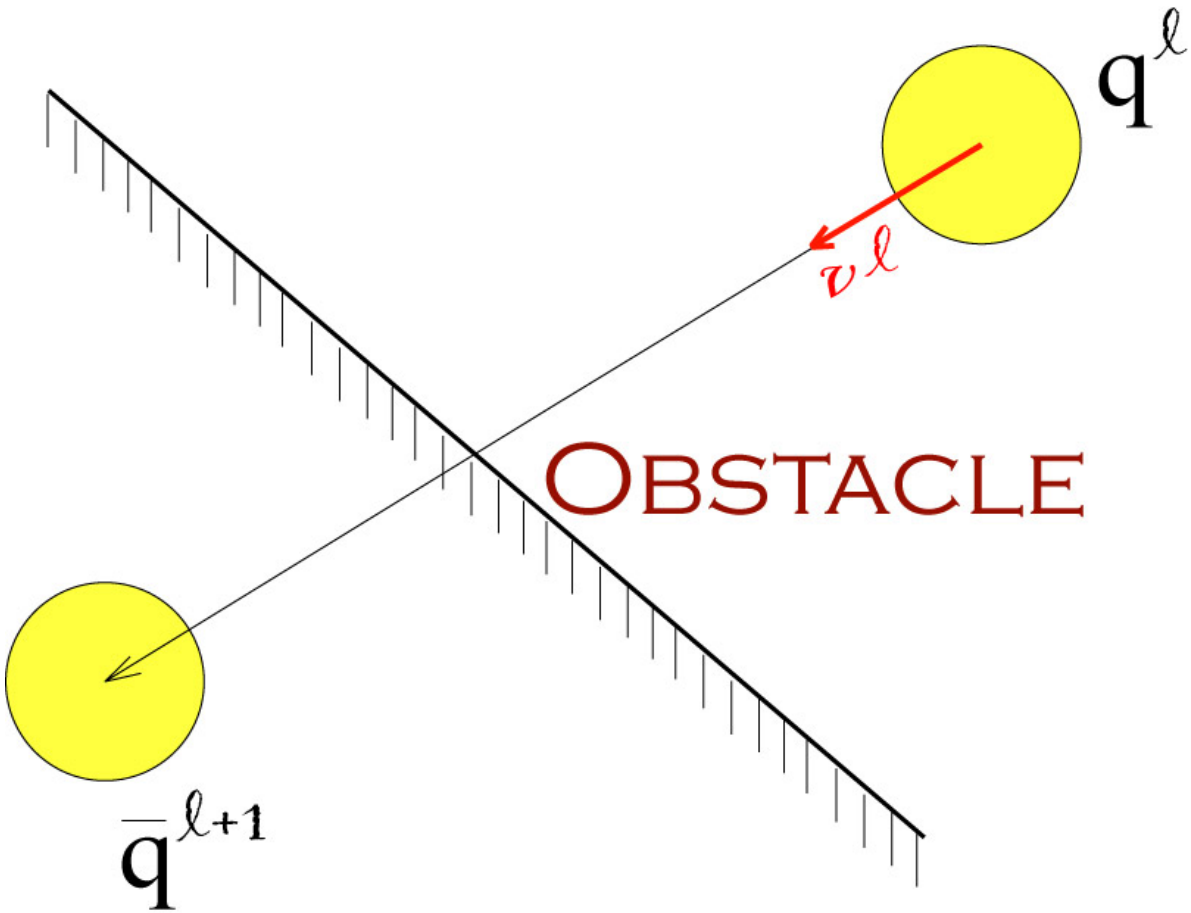
A case study: Disk - Line



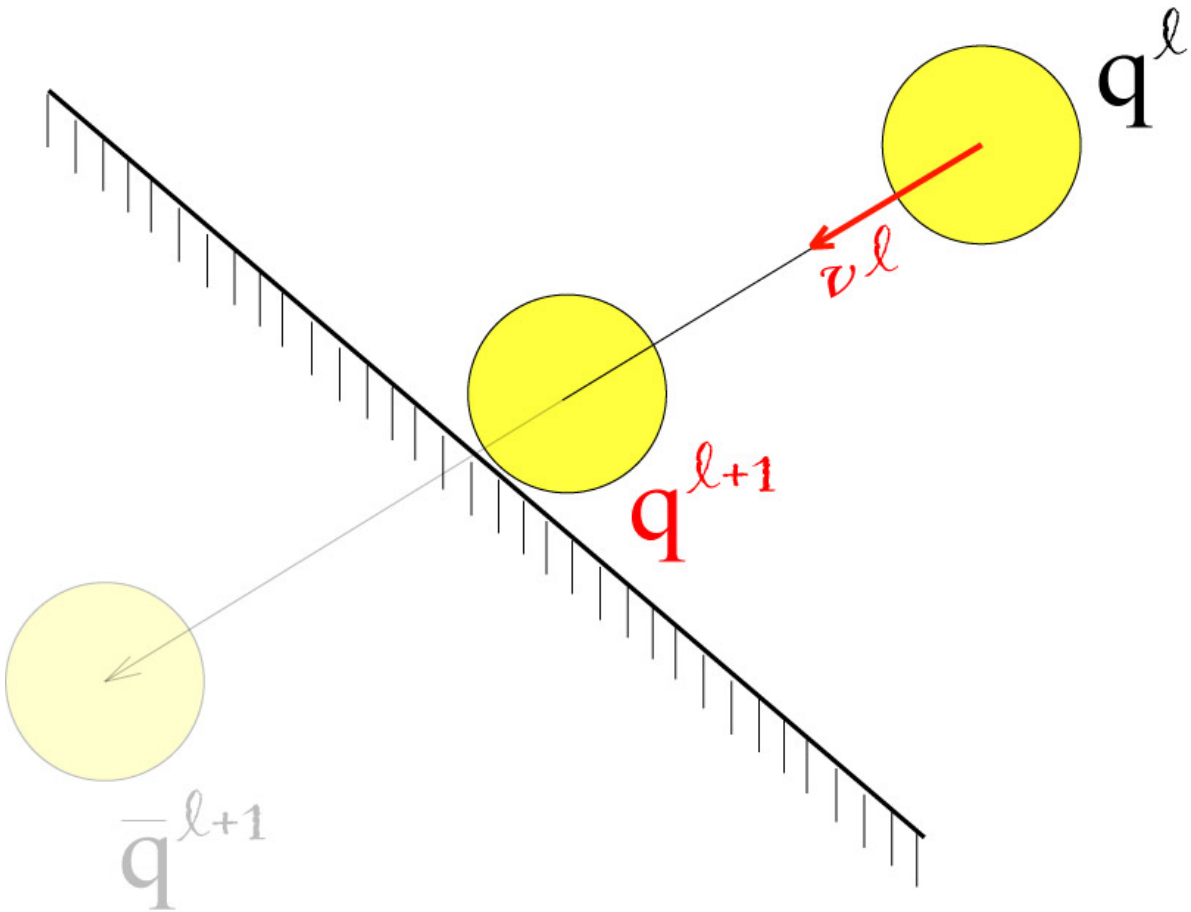
Disk - Line



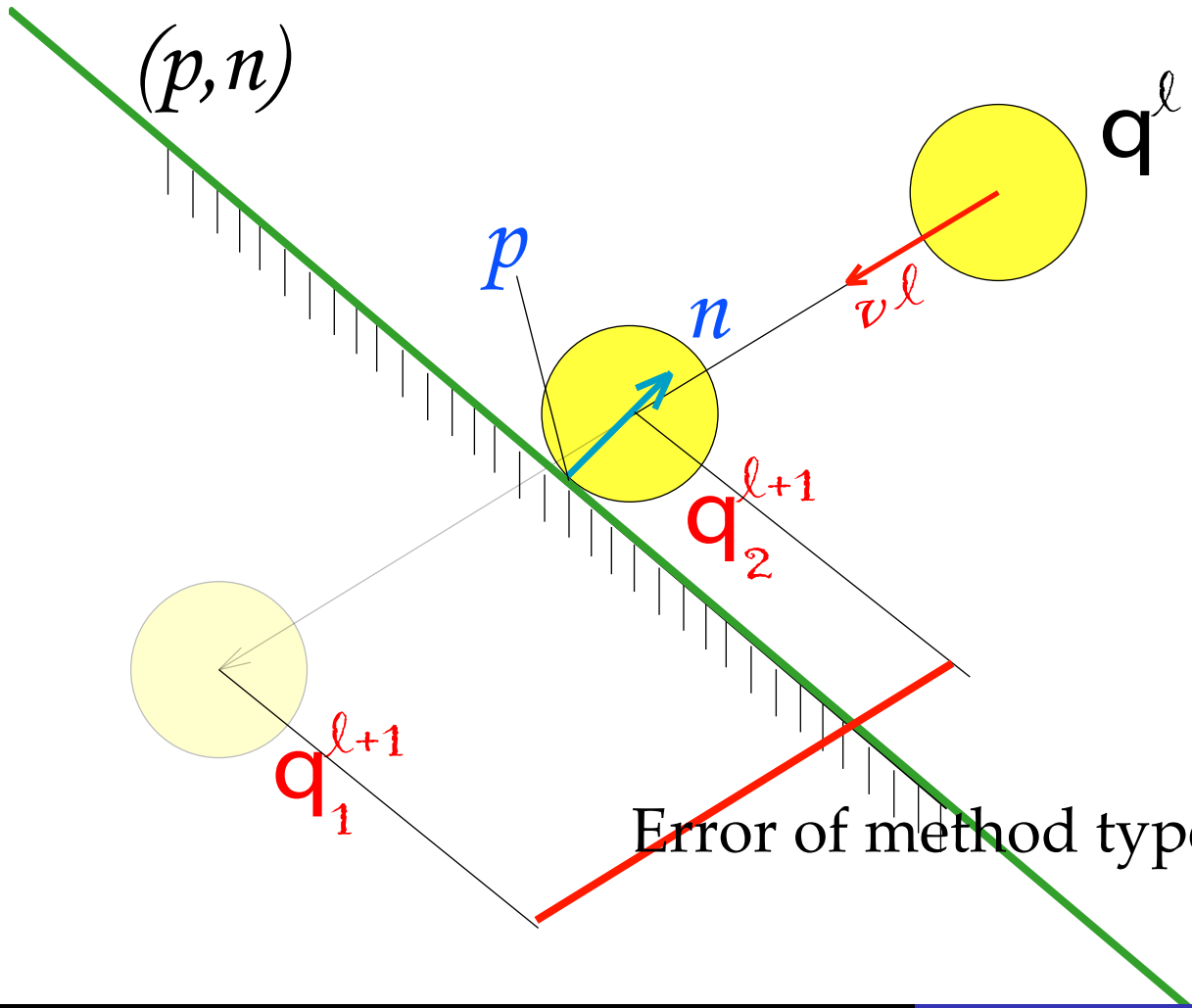
Disk - Line

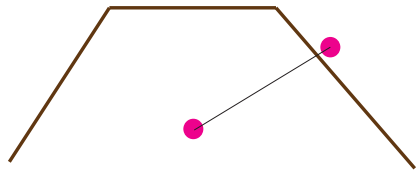


Disk - Line



Disk - Line





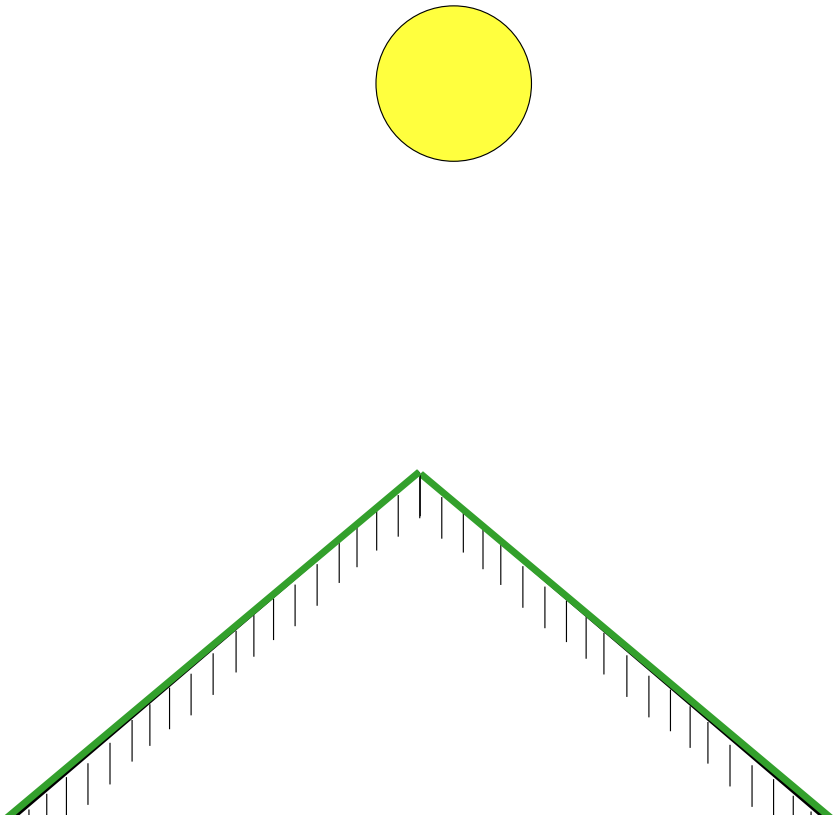
Constraints activation scheme

- Type 1 : Only activate when there is penetration. So the active set A reported from collision detection in this method at time I is empty and at time $I + 1$ is $\{(circle, plane)\}$.
- Type 2 : Only activate if there could be a penetration in the next time step. So the active set A reported from collision detection in this method is $\{(circle, plane)\}$ at both time I and $I + 1$. A simple (and actual used in practice) heuristic to decide whether to activate a constraint is if their distance is less than a threshold ϵ then activate.

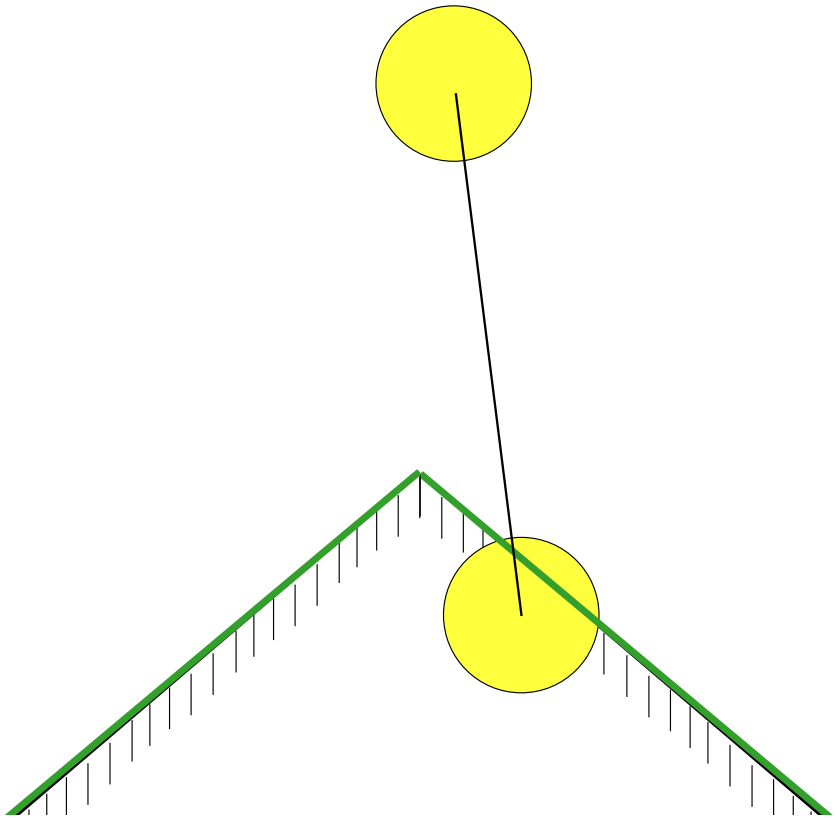
Lesson learnt

- Methods of type one always have a error proportion to relative velocity and time step even in this simplest example.
- Methods of type two give expected result.

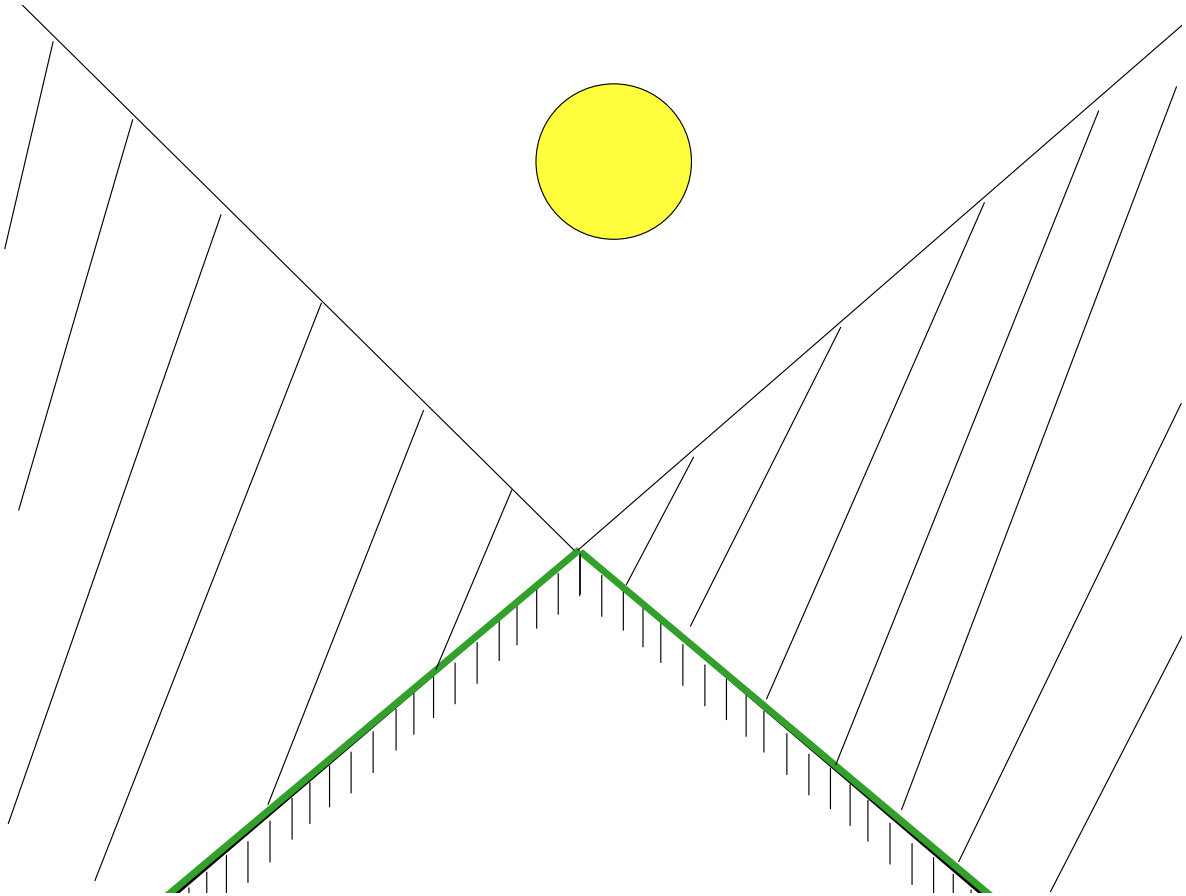
Disk - Two lines



Disk - Two lines



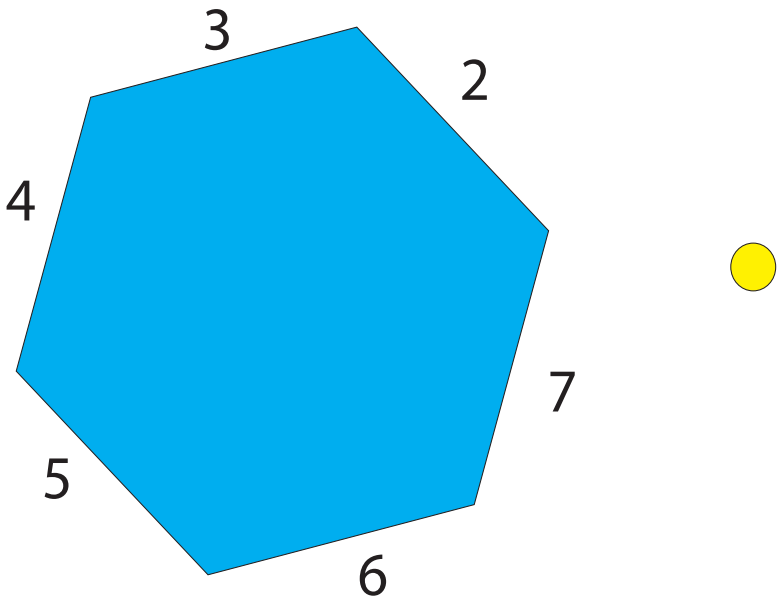
Disk - Two lines



The algorithm

Definition

A **group of constraint** corresponding to a pair of objects contains all possible constraints between them to make sure that there is no penetration.



Depend on the situation, we can choose the group of constraint differently.

The algorithm

- Step 1 (Exploratory step) : Formulate a complete system with all possible constraints. Solve for it and determine which constraint is actually active in a group. (Active constraint is the one with distance value equals zero)
- Step 2 : If there is no active constraint then go to step 4.
- Step 3 : Now reformulate with the correct active constraint determined in step 1 then solve for it.
- Step 4 : Update then go to step 1.

Conclusion

- The three types of methods to model non-penetration constraints have different errors and activation schemes.

Conclusion

- The three types of methods to model non-penetration constraints have different errors and activation schemes.
- The new method is more flexible and robust.

Conclusion

- The three types of methods to model non-penetration constraints have different errors and activation schemes.
- The new method is more flexible and robust.
- It can also remove the need for a sophisticated collision detection (with the price of having to solve larger LCP system).