

Assignment 5:

NAME: XIAOHAN WU

STUDENT NUMBER: 2019533093

EMAIL: WUXH@SHANGHAITECH.EDU.CN

1 INTRODUCTION

In HW5, we are required to implement a very basic rigid body simulation involving translation of spheres with collision detection and contact handling between spheres and parallelograms. The program contains 4 basic parts and 1 optional part. The four parts are: 1. Synchronize the simulation and the rendering in OpenGL to show the result in real time. 2. Implement the collision detection between the parallelograms and the sphere. 3. Implement the collision adjustment so that the inter-penetration is within the given tolerance (a small threshold). 4. Implement the colliding contact handling between the the parallelogram and the sphere without consideration of rotation. The optional part is: Simulate with multiple spheres simultaneously.

2 IMPLEMENTATION DETAILS

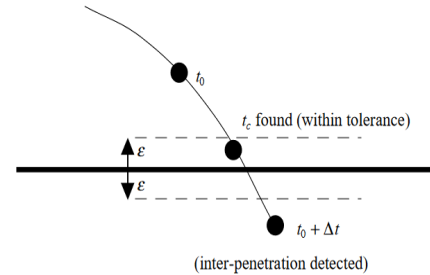
In this section, I will introduce how I implement my program for each part respectively.

1. The realization of moving the scene without considering collisions bases on the concept of momentum. First, we add up with all the force, multiplying it with time step dt to get the impulse, then divide it with the mass of the object to get its velocity at that time, also we can get the position of the object. After we finish Sphere::Forward part, for the Sphere::Backward part, it's nearly as same as the previous one except that it is in the inverse order.

2. After making the scene move, we start to detect the collision. To detect the collision between parallelogram and sphere is more difficult than that between sphere and sphere. We have to first calculate the distance between sphere and the parallelogram, comparing it with the radius of the sphere. Then, we have to project the sphere to the parallelogram to judge if the center of the sphere really can be projected onto the parallelogram, or it will also be judged as failure to collide. To detect the collision between two spheres, we need only to calculate the distance between two center positions of the spheres and compare it with the sum of the radius of two spheres.

3. After implementing the previous part, we have to adjust the collision point. As being taught in the class, we first forward the scene with dt . If inter-penetration is detected, then we have to backward and forward the scene back and forth such that time t_c is founded within the tolerance. However, it may occur that whatever we iterate, we can't find such time satisfying the condition. For that situation, we set up a threshold for the time to avoid unbreakable loop.

4. After the collision is detected and adjusted, the next step is to handle the collision contact. Before that, we should remember to



clear all the transient forces before (this can be easily forgotten. And also pay attention to that we have to clear them after adjusting the collision because the forward and backward still requires the calculation of those forces.) This is the most difficult part in this program. To handle the collision between wall and sphere, we have to first calculate the relative velocity along the surface normal and tangent direction respectively, and multiply them with corresponding epsilon to get the out velocity. Then we can get the delta of the relative velocity to calculate the change of the momentum so that to store that force into the sphere object to be processed in the next time step. Also, we need to pay attention to the supporting force of the surface. What I do in my code is that if the relative velocity of the sphere along the surface normal is smaller than the threshold, we calculate the joint force of the object and divide it into two directions (along surface normal and tangent). Thus the supporting force is the value of the one along surface normal with opposite direction. (This is an approximate model and not the same as the reality. But for simplicity, we program the code in such a easier way which can also simulate not bad.)

To handle the collision between two spheres, I just use the formula mentioned in the class to calculate the impulse between two collided spheres.

$$j = \frac{-(1 + \epsilon)v_{rel}}{\frac{1}{M_a} + \frac{1}{M_b} + \hat{n}(t_0) \cdot (J_a^{-1}(t_0)(r_a \times \hat{n}(t_0))) \times r_a + \hat{n}(t_0) \cdot (J_b^{-1}(t_0)(r_b \times \hat{n}(t_0))) \times r_b}.$$

3 RESULTS

Below are the results of my programming work.

1:2 • Name: Xiaohan Wu
student number: 2019533093
email: wuxh@shanghaitech.edu.cn

