Generator:

Since the sphere (generator)’s radius is known, we can use the following transformation to get the random sampling of the surface.



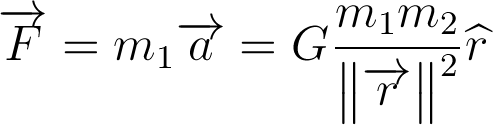
By getting random number for α and β (from 0 to 2π), we can convert it and produce a random sampling of the surface of the sphere.

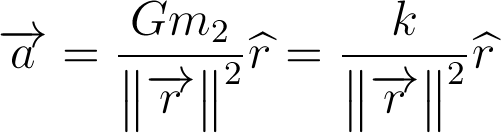
Gravity:

Discuss how to implement proper gravitational forces

Learnt from high school physics, the gravitational forces between two massed objects are

 = G \frac{m_1 m_2}{r^2}\ , where G is the gravitational constant, *m*1 is the first mass, *m*2 is the second mass, *r* is the distance between the centers of the masses. Assume first object is the particle, second object is the obstacle and obstacle does not move at all, then the force acting on the particle:

http://latex.codecogs.com/png.latex?%5Cdpi%7B300%7D%20F%20=%20m_1a%20=%20G%20%5Cfrac%7Bm_1m_2%7D%7Br%5E2%7Dhttp://latex.codecogs.com/png.latex?%5Cdpi%7B300%7D%20F%20=%20m_1a%20=%20G%20%5Cfrac%7Bm_1m_2%7D%7Br%5E2%7D

, where k is a constant.

By altering the k, one can simulate the strength of force acting on each particle. In each frame rate interval T, the velocity of the particle is in addition affected by the gravitational force, which will pull it towards the obstacle. There is no perfect number that makes the best simulation of the gravity field, but by giving the degree of freedom, users can change it to suit their needs.

Collision detection

We first update all particles’ velocities, and then we split into two computations. First one is to check the collision between particle pairs. For each pairs of particles, we assume they will collide after time s. Then we solve the equation d(A+sU,B+sV)=a+b, where A, B are the centers of the particles, U, V are their corresponding velocities and a, b are the radius of them. For all of the particles pairs, we find the pair with the smallest positive collision time ct1 and store it.

Second one is to check the collision between particles and obstacle. Like checking particles collision, we solve the same equation with B is the obstacle. And then we find the smallest positive collision time ct2 and store it. Then we compare ct1 and ct2 to get the smaller one and use it to do the collision action.