Test task

Main problem

The main problem with a discrete step Δt in this situation is that if Δt is too large, the sphere may miss all the points that would otherwise have fallen into the sphere with a smaller Δt.

To do this, you need to use not "Discrete collision detection", but "Continuous collision detection". To do this, I will use some approximation, the movement of the sphere between f(t) and f(t + Δt) can be considered as a linear movement of the center
of the sphere. I split the collision into two parts, for the linear movement of the sphere I use a cylinder (Figure 1), and also check the collision of the spheres

Others

I also split the point cloud into sectors and I don't need a collision check for all points, I need a collision check only in a certain sector (Figure 2)

For checking collision with a sphere I use this formula:

$$(position_x - center_x)^2 + (position_y - center_y)^2 + (position_z - center_z)^2 < radius^2.$$

For checking collision with a cylinder I use these formulas:

$$(point - base) * (top - base) >= 0$$

$$(point - top) * (top - base) <= 0$$

This will confirm that the point lies between the planes of the two circular facets of the cylinder

$$\frac{|(point - base) \times (top - base)|}{|top - base|} < radius$$

This will confirm that point lies inside the cylinder.

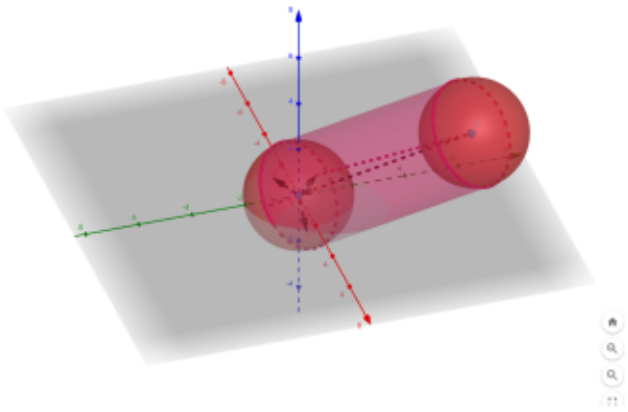

Figure 1

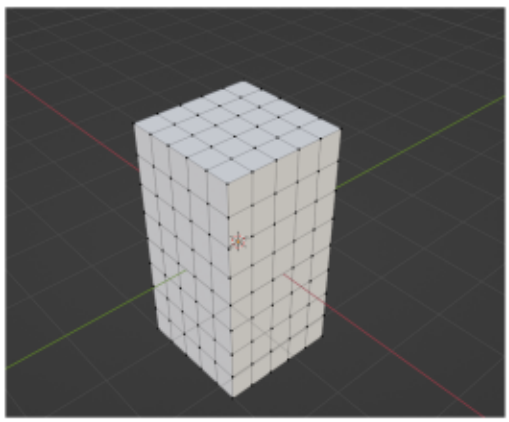

Figure 2