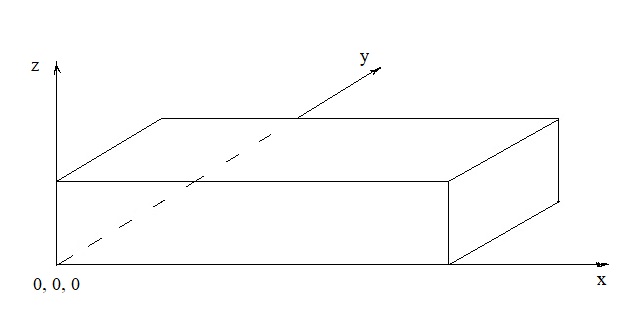
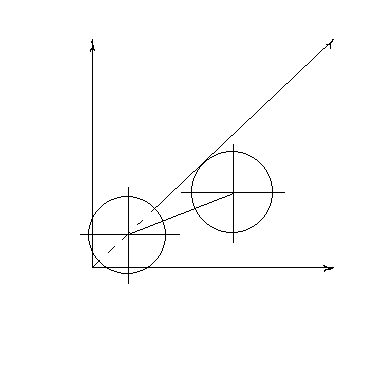
Documentation.

Given an Euclidian points cloud starting at 0,0,0 with constant distance grid (pic 1)

This cloud is intersected by a move of sphere where the path of center sphere defined as formula x=f(t). The function f(t) handled as a discrete function with Δt. Sphere formula is:





Sphere formula is:

(x-a)^2+(y-b)^2+(z-c)^2=R^2 where a, b, c

coordinates of center sphere that changes

by function evaluate with given Δt.

So enough compare (<=) left expression value

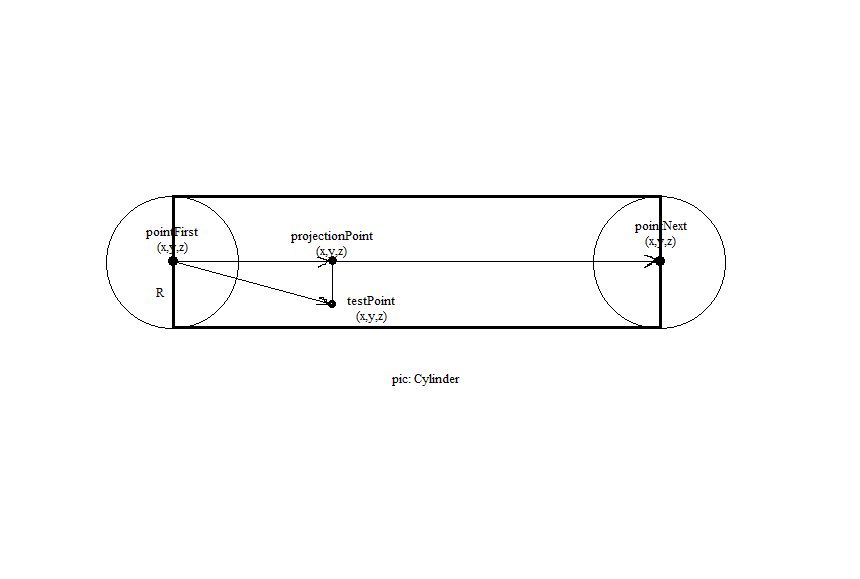
with square of radius, to check points of cloud

that intersected every sphere’s step.

To improve program performance exclude

check deleted cloud points. Also in program

used multithreading.



Having a discrete moving sphere, we need consider points of cloud between first and next step of sphere. As we see in pic – Cylinder. Sphere creates cylinders on its way every step. So having some point of Euclidian cloud (testPoint) we can check this point to belong cylinder.

Also we have vectors: vectorPointFirstpointNext = testPoint - pointFirst, VectorPointFirstTestPoint = pointNext - pointFirst. Knowing formula projection vector we can find length of vector pointFirstprojectionPoint. Then considering Pythagorean theorem we can find projectionPoint-testPoint line. Comparing length of vector pointFirstprojectionPoint with length of vectorPointFirstpointNext and length of projectionPoint-testPoint line with R of sphere we can approve about belonging testPoint to cylinder or not.