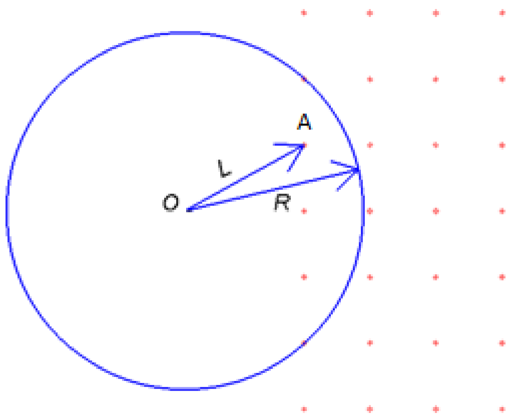
1. To check if point from given point cloud belong to sphere the next condition should be true:

$$|OA| <= R$$

where |OA| - length of vector with start at sphere center point and with end at point from given point cloud
O(a, b, c) – sphere center with a, b, c coordinates,
A(x, y, z) – intersecting point in point field with x, y, z coordinates,
R – sphere radius



Length of vector can be calculated as:

$$VLength = \sqrt{(a-x)^2 + (b-y)^2 + (c-z)^2}$$

2. A new class *ArrayPoints* can be written to store, compare, delete and print to file 3dpoint objects.
It will have private data – a *std::vector* collection to store 3dpoint objects, *std::vector<>::iterator* to iterate through vector, and *std::ofstream* object to print coordinates to specified file.
Also it will have member functions for main calculations, printing coordinates to file, comparing and deleting points.

VCalculate() – member function, that calculates length of vector by formula covered previously. It takes as arguments discrete function object, sphere radius and discrete step for sphere movement.
The main structure is consist of two nested for loops.
The first one represents sphere movement from starting parameter *t1* to end parameter *t2,* received from discrete function class object.
The second one iterates through container of 3dpoints object. Main calculations performed inside for each point from point cloud. *Sqrt()* and *pow()* functions from *math.h* should be used for vector length calculation.

Then another member function void *delArrayElem()* is calling inside nested loop to compare length of vector with radius of sphere, and in case if main condition is true – delete point.

Main condition looks like:

If (VLenght <= R) then delete point.
printArray() – function which prints coordinates to specified file in "0.0" format. *Setprecision()* from *iomanip* library is used to set format.

The *main()* function from *CreateSkin.cpp* have only new *ArrayPoints* class objects creation with passed arguments, which call then its member function. Constructor of *ArrayPoints* class object written to fill container with 3dpoint object, set their coordinates and open *ofstream* object for writing to file. It takes as arguments *reference point* object, max number of points in X, Y, Z directions, value of distance between points in the point grid and name of the file to write data.

3. If discrete step will be too large and density of point cloud to high the result will be inaccurate. Some point can be missed and remained undeleted. So the resulting surface will be unsmooth, probably will have ribs that is inappropriate. Quality of final product will be lower and additional technical operations will be needed to smooth surface.