KAIST CS492 - Homework 3

XU YIN - 20205445

November 8, 2021

Question 2

0.1 Evaluating a point on the surface

- Fill the control points (see Fig. 1).

```
//Positive y and positive z quarter of torus
{
    {
        {R + r, 0, 0, 1},
        {R - r, 0, 0, 1},
        {-(R + r), 0, 0, 1}
    },
    {
        {-(R - r), 0, 0, 1},
        {0, 0, r, 0},
        {0, 0, R + r, 0}
    },
    {
        {0, 0, 0, 0},
        {0, 0, R - r, 0},
        {0, 0, r, 0}
    }
},
```

Figure 1: fill the control points of one quarter.

- Fill in the function Calculate(u,v,k)(show in Fig. 2).
- Demonstration (see Fig. 3).

0.2 Construct a mesh

- list the vertexes and insert the face (see Fig. 4).
- Demonstration (see Fig. 5).

0.3 A whole torus

- Insert all control points and use the symmetry calculation (see Fig. 6).

```
Point Calculate(float u, float v, int idx) {
    Point surfacePoint;
    surfacePoint.x = surfacePoint.y = surfacePoint.z = 0;
    surfacePoint.w = 1;

    // STUDENT CODE SECTION 2
    // WRITE CODE HERE TO EVALUATE THE VERTEX POSITION OF

    double sum = R + r + u * u*(R - r);
    surfacePoint.x = (1 - v * v)*sum;
    surfacePoint.y = sum * 2 * v;
    surfacePoint.z = 2 * u*r*(1 + v * v);
    surfacePoint.w = (1 + u * u)*(1 + v * v);



    // ------------------------------------------------

    return surfacePoint;
}
```

Figure 2: fill the calculation function.

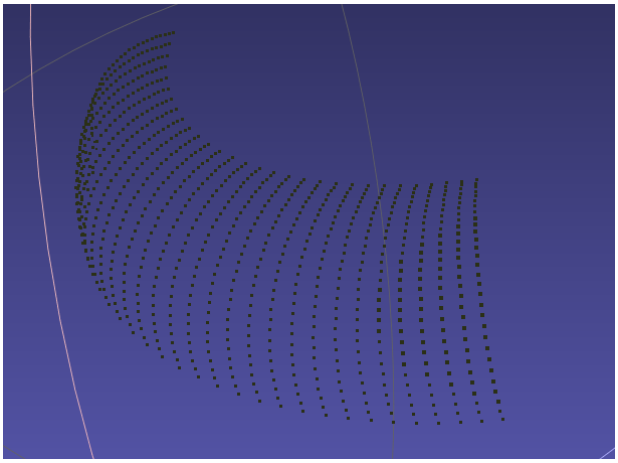

Figure 3: fill the control points of one quarter.

- Demonstration of the whole torus.

```
std::vector<Mesh::VertexHandle>  face_vhandles;
for (int k = posYposZ; k <= negYnegZ; ++k) {
for (int i = 0; i < LOD-1 ; ++i) {
    for (int j = 0; j < LOD-1 ; ++j) {

            // STUDENT CODE SECTION 3
            // NEED TO ADD FACES TO THE MESH USI
            // ----------------------------------
            face_vhandles.push_back(vhandle[k][i
        }
    }
mesh->add_face(face_vhandles);
face_vhandles.clear();

}
```

Figure 4: Demonstration of a quarter.

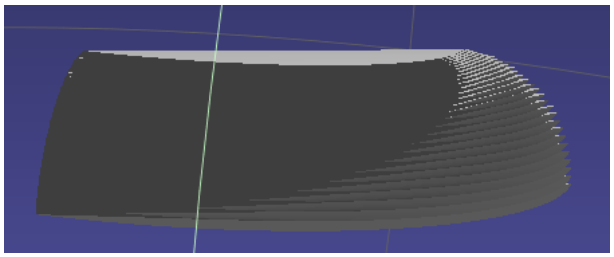

Figure 5: A quarter of torus.

```
if (idx % 2)
{
    surfacePoint.z /= -1;

}
if (idx >= 2)
{
    surfacePoint.y /= -1;
}
```

Figure 6: Using the symmetry calculation.

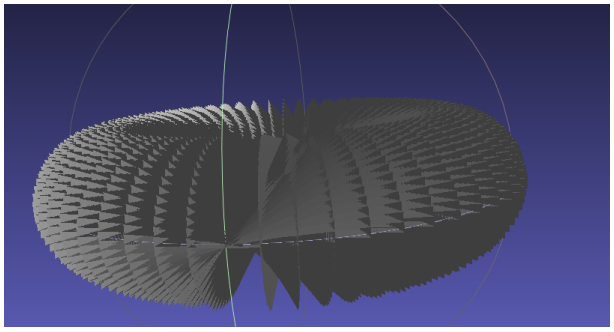

Figure 7: Show the whole torus.