**4.13 錐狀舌片體**

var Debug = Core.Debug;

var Mesh3D = Core.Mesh3D;

var Tess = Core.Tess;

var Solid = Core.Solid;

// Template Code:

params = [

{ "id": "innerRadius", "displayName": "Inner radius", "type": "length", "rangeMin": 1, "rangeMax": 100, "default": 5.0 },

{ "id": "outerRadius", "displayName": "Outer radius", "type": "length", "rangeMin": 1, "rangeMax": 100, "default": 10.0 },

{ "id": "innerRadius\_top", "displayName": "Inner radius top", "type": "length", "rangeMin": 1, "rangeMax": 100, "default": 5.0 },

{ "id": "outerRadius\_top", "displayName": "Outer radius top", "type": "length", "rangeMin": 1, "rangeMax": 100, "default": 10.0 },

{ "id": "tongsNb", "displayName": "Tongs", "type": "int", "rangeMin": 1, "rangeMax": 60, "default": 20 },

{ "id": "tongThickness", "displayName": "Tong thickness", "type": "float", "rangeMin": 1, "rangeMax": 100, "default": 30 },

{ "id": "height", "displayName": "Height", "type": "length", "rangeMin": 1, "rangeMax": 100, "default": 10 },

{ "id": "x\_pos","displayName": "x position","type": "float","rangeMin": -100,"rangeMax": 100,"default": 0},

{ "id": "y\_pos","displayName": "y position","type": "float","rangeMin": -100,"rangeMax": 100,"default": 0 },

{ "id": "z\_pos","displayName": "z position","type": "float","rangeMin": -100,"rangeMax": 100,"default": 0}

];

function xyAt(xy,height){

return [xy[0], xy[1], height];

}

function process(params) {

var x\_pos = params.x\_pos;

var y\_pos = params.y\_pos;

var z\_pos = params.z\_pos;

var innerRadius = Math.min(params.innerRadius, params.outerRadius);

var outerRadius = Math.max(params.innerRadius, params.outerRadius);

var innerRadius\_top = Math.min(params.innerRadius\_top, params.outerRadius\_top);

var outerRadius\_top = Math.max(params.innerRadius\_top, params.outerRadius\_top);

var tongsNb = params.tongsNb;

var tongAngle = 0;

var tongThickness = params.tongThickness;

var height = params.height;

var angle = 2\*Math.PI / tongsNb;

var angleThickness = angle \* tongThickness/100;

var outerAngleDelta = 2\*Math.PI\*tongAngle/360;

var tongs = [];

var tongs\_top = [];

Debug.color(255, 0, 255);

for (var i = 0; i < tongsNb; i++) {

var x0 = x\_pos+innerRadius \* Math.cos(i \* angle);

var y0 = y\_pos+innerRadius \* Math.sin(i \* angle);

var x1 = x\_pos+outerRadius \* Math.cos(i \* angle+outerAngleDelta);

var y1 = y\_pos+outerRadius \* Math.sin(i \* angle+outerAngleDelta);

var x2 = x\_pos+outerRadius \* Math.cos(i \* angle+outerAngleDelta+angleThickness);

var y2 = y\_pos+outerRadius \* Math.sin(i \* angle+outerAngleDelta+angleThickness);

var x3 = x\_pos+innerRadius \* Math.cos(i \* angle+angleThickness);

var y3 = y\_pos+innerRadius \* Math.sin(i \* angle+angleThickness);

tongs.push([[x0, y0], [x1, y1], [x2, y2], [x3, y3]]);

//舌片底部四點

var x0\_top = x\_pos+innerRadius\_top \* Math.cos(i \* angle);

var y0\_top = y\_pos+innerRadius\_top \* Math.sin(i \* angle);

var x1\_top = x\_pos+outerRadius\_top \* Math.cos(i \* angle+outerAngleDelta);

var y1\_top = y\_pos+outerRadius\_top \* Math.sin(i \* angle+outerAngleDelta);

var x2\_top = x\_pos+outerRadius\_top \* Math.cos(i \* angle+outerAngleDelta+angleThickness);

var y2\_top = y\_pos+outerRadius\_top \* Math.sin(i \* angle+outerAngleDelta+angleThickness);

var x3\_top = x\_pos+innerRadius\_top\* Math.cos(i \* angle+angleThickness);

var y3\_top = y\_pos+innerRadius\_top \* Math.sin(i \* angle+angleThickness);

tongs\_top.push([[x0\_top, y0\_top], [x1\_top, y1\_top], [x2\_top, y2\_top], [x3\_top, y3\_top]]); //舌片頂部四點

}

var mesh = new Mesh3D();

for (var i = 0; i < tongs.length; i++) {

var tong = tongs[i]; //掃描舌片底部四點

var tong\_top = tongs\_top[i]; //掃描舌片頂部四點

//舌片六個面

mesh.quad(xyAt(tong[0], 0), xyAt(tong[1], 0), xyAt(tong[2], 0), xyAt(tong[3], 0));

mesh.quad(xyAt(tong[0], z\_pos+0), xyAt(tong[1], z\_pos+0), xyAt(tong\_top[1], z\_pos+height), xyAt(tong\_top[0], z\_pos+height));

mesh.quad(xyAt(tong[1], z\_pos+0), xyAt(tong[2], z\_pos+0), xyAt(tong\_top[2], z\_pos+height), xyAt(tong\_top[1], z\_pos+height));

mesh.quad(xyAt(tong[2], z\_pos+0), xyAt(tong[3], z\_pos+0), xyAt(tong\_top[3], z\_pos+height), xyAt(tong\_top[2], z\_pos+height));

mesh.quad(xyAt(tong[3], z\_pos+0), xyAt(tong[0], z\_pos+0), xyAt(tong\_top[0], z\_pos+height), xyAt(tong\_top[3], z\_pos+height));

mesh.quad(xyAt(tong[0], height), xyAt(tong[1], height), xyAt(tong[2], height), xyAt(tong[3], height));

}

var solid = Solid.make(mesh);

return solid;

**}**