**4.25 圓周多斜圓筒**

var Debug = Core.Debug;

var Mesh3D = Core.Mesh3D;

var Path2D = Core.Path2D;

var Plugin = Core.Plugin;

var Tess = Core.Tess;

var Solid = Core.Solid;

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

// Original Shape Script:

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

params = [

{ "id": "bigTopDia", "displayName": "Big Top Diameter", "type": "length", "rangeMin": 0, "rangeMax": 50, "default": 25 },

{ "id": "bigBottomDia", "displayName": "Big Bottom Diameter", "type": "length", "rangeMin": 0, "rangeMax": 50, "default": 36 },

{ "id": "number", "displayName": "Number of Cylinder", "type": "int", "rangeMin": 1, "rangeMax": 20, "default": 5 },

{ "id": "r1", "displayName": "Base Radius", "type": "length", "rangeMin": 0, "rangeMax": 100, "default": 10.0 },

{ "id": "r1\_inner", "displayName": "Base inner Radius", "type": "length", "rangeMin": 0, "rangeMax": 100, "default": 5.0 },

{ "id": "r2", "displayName": "Top Radius", "type": "length", "rangeMin": 0, "rangeMax": 100, "default": 10.0 },

{ "id": "r2\_inner", "displayName": "Top inner Radius", "type": "length", "rangeMin": 0, "rangeMax": 100, "default": 5.0 },

{ "id": "height", "displayName": "Height", "type": "length", "rangeMin": 1.0, "rangeMax": 100.0, "default": 20.0 },

{ "id": "arc", "displayName": "Arc", "type": "angle", "rangeMin": 0, "rangeMax": 360.0, "default": 270.0 },

//{ "id": "x\_pos\_Top", "displayName": "Top x position","type": "float","rangeMin": -100,"rangeMax": 100,"default": 10 },

//{ "id": "y\_pos\_Top", "displayName": "Top y position", "type": "float","rangeMin": -100,"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},

{ "id": "times", "displayName": "precision", "type": "int", "rangeMin": 3, "rangeMax": 200, "default": 54}

];

function process(params) {

var bigTopDia= params["bigTopDia"];

var bigBottomDia = params["bigBottomDia"];

var number= params["number"];

var x\_pos = params.x\_pos;

var y\_pos = params.y\_pos;

var z\_pos = params.z\_pos;

var times = params["times"];

var height = params["height"];

var arc = params ["arc"];

var r1 = params["r1"];

var r1\_inner = params["r1\_inner"];

r1\_inner =Math.min(r1,r1\_inner);

var r2 = params["r2"];

var r2\_inner = params["r2\_inner"];

r2\_inner =Math.min(r2,r2\_inner);

var mesh = new Mesh3D();

for (var k = 0; k < number; k++) { //掃描大圈

var bigAngle= Math.PI\*2.0\*k/number; //大圈角度

var bigBottomX=bigBottomDia\* Math.cos( bigAngle);   
 //底部中心x座標

var bigBottomY= bigBottomDia\*Math.sin( bigAngle);   
 //底部中心y座標

var bigTopX=bigTopDia\* Math.cos( bigAngle);   
 //頂部中心x座標

var bigTopY= bigTopDia\*Math.sin( bigAngle);   
 //頂部中心y座標

var cl = [x\_pos+bigBottomX+r1\_inner,y\_pos+bigBottomY+0,z\_pos+0];   
 //內圈低點(起始點)

var ch = [x\_pos+bigTopX+r2\_inner,y\_pos+bigTopY+0,z\_pos+height];   
 //內圈高點(起始點)

var pl = [x\_pos+bigBottomX+r1,y\_pos+bigBottomY+0,z\_pos+0]; //外圈低點(起始點)

var ph = [x\_pos+bigTopX+r2,y\_pos+bigTopY+0,z\_pos+height];   
 //外圈高點(起始點)

if (arc < 360) {

mesh.triangle(cl, pl, ch); //起始側邊三角形1

mesh.triangle(cl, ph, ch); //起始側邊三角形2

}

var ndivs=times;

for (var i = 0; i < ndivs; i++) { //開始掃描

var a = (i+1)/ndivs \* arc / 180 \* Math.PI;

var c = Math.cos(a);

var s = Math.sin(a);

var ncl = [x\_pos+bigBottomX+r1\_inner\*c, y\_pos+bigBottomY-r1\_inner\*s,z\_pos+ 0];

//內圈低點(掃描點)

var nch = [x\_pos+bigTopX+r2\_inner\*c, y\_pos+bigTopY-r2\_inner\*s, z\_pos+height]; //內圈高點(掃描點)

var npl = [x\_pos+bigBottomX+r1\*c, y\_pos+bigBottomY-r1\*s,z\_pos+ 0]; //外圈低點(掃描點)

var nph = [x\_pos+bigTopX+r2\*c, y\_pos+bigTopY-r2\*s, z\_pos+height];

//外圈高點(掃描點)

mesh.triangle(pl, ph, npl); //外部三角形1

mesh.triangle(npl, ph, nph); //外部三角形2

mesh.triangle(cl, ch, ncl); //內部三角形1

mesh.triangle(ncl, ch, nch); //內部三角形2

mesh.triangle(cl, pl, npl); //底部三角形1

mesh.triangle(npl, ncl, cl); //底部三角形2

mesh.triangle(ch, nph, ph); //頂部三角形1

mesh.triangle(ch, nch, nph); //頂部三角形2

cl=ncl;

ch=nch;

pl = npl;

ph = nph;

}

// open cap

if (arc < 360) {

mesh.triangle(cl, ph, ch); //結束側邊三角形1

mesh.triangle(cl, ph, pl); //結束側邊三角形2

}

}

var solid = Solid.make(mesh);

return solid;

}

presets = [

{ 'color': [33, 129, 45] }

];