**4.24 圓周多斜圓柱**

var Mesh3D = Core.Mesh3D;

var Plugin = Core.Plugin;

var Tess = Core.Tess;

var Sketch2D = Core.Sketch2D;

var Solid = Core.Solid;

var Vector2D = Core.Vector2D;

var Vector3D = Core.Vector3D;

// Template Code:

params = [

{ "id": "bigTopDia", "displayName": "Big Top Diameter", "type": "length", "rangeMin": 0, "rangeMax": 50, "default": 10.0 },

{ "id": "bigBottomDia", "displayName": "Big Bottom Diameter", "type": "length", "rangeMin": 0, "rangeMax": 50, "default": 20.0 },

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

{ "id": "diaTop", "displayName": "Top Diameter", "type": "length", "rangeMin": 0, "rangeMax": 50, "default": 10.0 },

{ "id": "dia", "displayName": "Bottom Diameter", "type": "length", "rangeMin": 0, "rangeMax": 50, "default": 20.0 },

{ "id": "height", "displayName": "Height", "type": "length", "rangeMin": 0, "rangeMax": 50, "default": 10.0 },

{ "id": "res", "displayName": "Resolution", "type": "int", "rangeMin": 3, "rangeMax": 50, "default": 10 },

{ "id": "x\_pos", "displayName": "Bottom x position","type": "float","rangeMin": -50,"rangeMax": 50,"default": 0 },

{ "id": "y\_pos", "displayName": "Bottom y position", "type": "float","rangeMin": -50,"rangeMax": 50, "default": 0},

{ "id": "z\_pos", "displayName": "Bottom z position", "type": "float","rangeMin": -50, "rangeMax": 50,"default": 0 }

];

function process(params) {

var bigTopDia= params["bigTopDia"];

var bigBottomDia = params["bigBottomDia"];

var number= params["number"];

var diaTop = params["diaTop"];

var dia = params["dia"];

var height = params["height"];

var res = params["res"]; //取解析度

var r2Top=diaTop/2.0;

var r2 = dia/2.0;

//設定解析度\*外圈半徑當作分割值，

//之後就是三角形數量

var x\_pos = params["x\_pos"];

var y\_pos = params["y\_pos"];

var z\_pos = params["z\_pos"];

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 inL = [x\_pos+bigBottomX,y\_pos+bigBottomY,z\_pos+0];  
 //設定0度內圈底部點為"目前內圈底部點"

var inH = [x\_pos+bigTopX,y\_pos+bigTopY,z\_pos+height];  
 //0度內圈頂部點為"目前頂圈底部點"

var outL = [x\_pos+bigBottomX+r2,y\_pos+bigBottomY,z\_pos+0];   
 //0度外圈底部點"目前外圈底部點"

var outH = [x\_pos+bigTopX+r2Top,y\_pos+bigTopY,z\_pos+height];  
 //0度外圈頂部點"目前外圈頂部點"

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

var a = (i+1)/res\* Math.PI\*2.0;   
 //由０開始環繞360度取ndivs個角度

var s = Math.sin(a);

var c = Math.cos(a);

//準備順時鐘繞取點

var nextinL = [x\_pos+bigBottomX, y\_pos+bigBottomY,z\_pos+0];   
 //-a度內圈底部點

var nextinH = [x\_pos+bigTopX, y\_pos+bigTopY, z\_pos+height];  
 //-a度內圈頂部點

var nextoutL = [x\_pos+bigBottomX+r2\*c, y\_pos+bigBottomY-r2\*s,z\_pos+ 0];   
 //-a度外圈底部點

var nextoutH = [x\_pos+bigTopX+r2Top\*c, y\_pos+bigTopY-r2Top\*s, z\_pos+height];

//-a度外圈頂部點

mesh.triangle(inH, outH, nextoutH); //頂部三角形

mesh.quad(outL, nextoutL,nextoutH,outH); //外部四角形

mesh.triangle(inL,nextoutL,outL); //底部三角形

//更新所有目前四點

inL = nextinL;

inH = nextinH;

outL = nextoutL;

outH = nextoutH;

}

}

var solid = Solid.make(mesh);

return solid;

}