**4.5 橢體**

var Conversions = Core.Conversions;

var Point2D = Core.Point2D;

var Debug = Core.Debug;

var Path2D = Core.Path2D;

var Point3D = Core.Point3D;

var Matrix2D = Core.Matrix2D;

var Matrix3D = Core.Matrix3D;

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;

params = [

{ "id": "numberOfPoints", "displayName": "# of cone sides", "type": "int", "rangeMin": 4, "rangeMax":330, "default": 20 },

{ "id": "radius\_a", "displayName": "Radius a of ecllipse", "type": "length", "rangeMin": 1, "rangeMax": 150, "default": 10 },

{ "id": "radius\_b", "displayName": "Radius b of ecllipse", "type": "length", "rangeMin": 1, "rangeMax": 150, "default": 10 },

{ "id": "radius\_c", "displayName": "Radius c of ecllipse", "type": "length", "rangeMin": 1, "rangeMax": 150, "default": 10 },

{ "id": "x\_pos",

"displayName": "x position",

"type": "float",

"rangeMin": -200,

"rangeMax": 200,

"default": 0

//預設100是因為將圖形放到桌面中央時，調整x、y position為零時，可將圖中心線形移往桌面左下角。

},

{ "id": "y\_pos",

"displayName": "y position",

"type": "float",

"rangeMin": -200,

"rangeMax": 200,

"default": 0

},

{ "id": "z\_pos",

"displayName": "z position",

"type": "float",

"rangeMin": -200,

"rangeMax": 200,

"default": 0

}

];

function ran(maxNum,minNum)

{

var re = Math.random() \* (maxNum - minNum ) + minNum;

return re;

}

function process(params)

{

var numPoints = params['numberOfPoints'];

var ndivs = Math.max(4, numPoints);

var h = params["Height"] ;

var ra = params["radius\_a"] ;

var rb = params["radius\_b"] ;

var rc = params["radius\_c"] ;

var x\_pos = params.x\_pos;

var y\_pos = params.y\_pos;

var z\_pos = params.z\_pos;

ra=Math.max(1, ra) ;

rb=Math.max(1, rb) ;

rc=Math.max(1, rc) ;

var h\_down = params["Height\_down"] ;

var numb = params['number'];

var down=[];

var star = [];

var mesh = new Mesh3D();

// var bot = [0,0,0];

// upper half ellipsoid

var top = [x\_pos+0,y\_pos+0,z\_pos+rc];

var net=[];

var zr;

net[0]=[];

zr=1;

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

net[0][i]=[];

var a = i/ndivs \* Math.PI\*2;

var x = Math.cos(a) \* ra\*zr;

var y = Math.sin(a) \* rb\*zr;

net[0][i].push(x\_pos+x,y\_pos+ y,z\_pos+ 0);

}

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

// mesh.triangle(bot, net[0][i], net[0][i+1]);

// mesh.triangle(top, net[0][i], net[0][i+1]);

// }

for (var j = 1; j < ndivs; j++)

{

net[j]=[];

zr=Math.sqrt(1-j\*j/ndivs/ndivs);

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

net[j][i]=[];

var a = i/ndivs \* Math.PI\*2;

var x = Math.cos(a) \* ra\*zr;

var y = Math.sin(a) \* rb\*zr;

net[j][i].push(x\_pos+x, y\_pos+y,z\_pos+ j/ndivs\*rc);

}

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

mesh.triangle(net[j-1][i], net[j-1][i+1], net[j][i]);

mesh.triangle(net[j-1][i+1], net[j][i], net[j][i+1]);

if(j==ndivs-1) mesh.triangle(top, net[j][i], net[j][i+1]);

}

}

// lower half ellipsoid

var bottom = [x\_pos+0,y\_pos+0,z\_pos-rc];

// var top = [0,0,rc];

var bnet=[];

var bzr;

bnet[0]=[];

bzr=1;

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

bnet[0][i]=[];

var a = i/ndivs \* Math.PI\*2;

var x = Math.cos(a) \* ra\*bzr;

var y = Math.sin(a) \* rb\*bzr;

bnet[0][i].push(x\_pos+x,y\_pos+ y, z\_pos+0);

}

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

// mesh.triangle(bot, net[0][i], net[0][i+1]);

// mesh.triangle(top, net[0][i], net[0][i+1]);

// }

for (var j = 1; j < ndivs; j++)

{

bnet[j]=[];

bzr=Math.sqrt(1-j\*j/ndivs/ndivs);

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

bnet[j][i]=[];

var a = i/ndivs \* Math.PI\*2;

var xc = Math.cos(a) \* ra\*bzr;

var yc = Math.sin(a) \* rb\*bzr;

bnet[j][i].push(x\_pos+xc, y\_pos+yc,z\_pos -1\*j/ndivs\*rc);

}

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

mesh.triangle(bnet[j-1][i], bnet[j-1][i+1], bnet[j][i]);

mesh.triangle(bnet[j-1][i+1], bnet[j][i], bnet[j][i+1]);

if(j==ndivs-1) mesh.triangle(bottom, bnet[j][i], bnet[j][i+1]);

}

}

return Solid.make(mesh);

}