**4.14 齒輪設計**

var Conversions = Core.Conversions;

var Debug = Core.Debug;

var Path2D = Core.Path2D;

var Point2D = Core.Point2D;

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": "height",

"displayName": "Height",

"type": "length",

"rangeMin": 1,

"rangeMax": 128,

"default": 4

},

{ "id": "teeth",

"displayName": "Teeth",

"type": "int",

"rangeMin": 3,

"rangeMax": 128,

"default": 8

},

{ "id": "toothbasewidth",

"displayName": "Tooth Base Width",

"type": "length",

"rangeMin": 1,

"rangeMax": 12,

"default": 3

},

{ "id": "toothbaseheight",

"displayName": "Tooth Base Height",

"type": "length",

"rangeMin": 1,

"rangeMax": 12,

"default": 2

},

{ "id": "toothcorewidth",

"displayName": "Tooth Core Width",

"type": "length",

"rangeMin": 1,

"rangeMax": 12,

"default": 2

},

{ "id": "toothcoreheight",

"displayName": "Tooth Core Height",

"type": "length",

"rangeMin": 1,

"rangeMax": 12,

"default": 1

},

{ "id": "toothtipwidth",

"displayName": "Tooth Tip Width",

"type": "length",

"rangeMin": 1,

"rangeMax": 12,

"default": 1

},

{ "id": "toothgapwidth",

"displayName": "Tooth Gap Width",

"type": "length",

"rangeMin": 1,

"rangeMax": 12,

"default": 3

},

{ "id": "bevelangle",

"displayName": "Bevel Angle",

"type": "int",

"rangeMin": 0,

"rangeMax": 85,

"default": 0

},

{ "id": "helicalangle",

"displayName": "Helical Angle",

"type": "int",

"rangeMin": 0,

"rangeMax": 720,

"default": 0

},

{ "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 jellyQuad( mesh, p1, p2, p3, p4 )

{

mesh.quad( p1, p2, p3, p4 );

}

function jellyTriangle( mesh, p1, p2, p3 )

{

mesh.triangle( p1, p2, p3 );

}

function process(params)

{

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

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

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

var tooth;

var h;

var i;

var height = params.height;

var teeth = params.teeth;

var bevelangle = params.bevelangle \* Math.PI / 180.0;

var helicalangle = params.helicalangle \* Math.PI / 180.0;

var toothbaseheight = params.toothbaseheight;

var toothcoreheight = params.toothcoreheight;

var toothbasewidth = params.toothbasewidth;

var toothcorewidth = params.toothcorewidth;

var toothtipwidth = params.toothtipwidth;

var toothgapwidth = params.toothgapwidth;

var hsteps = (height \* 2);

var dh = height / hsteps;

var h0 = 0;

var h1 = 0;

var circ0 = 0;

var circ1 = (toothbasewidth + toothgapwidth) \* teeth;

var radius0 = 0;

var radius1 = circ1 / (2.0 \* Math.PI);

var helicaloffset = height \* Math.tan(helicalangle) / radius1;

var dhelix = helicaloffset / hsteps;

var toothbaseradial = toothbasewidth / radius1;

var toothcoreradial = toothcorewidth / (radius1 + toothbaseheight);

var toothtipradial = toothtipwidth / (radius1 + toothbaseheight + toothcoreheight);

var toothgapradial = toothgapwidth / radius1;

var mesh = new Mesh3D();

var radial = [0,0,0,0,0,0,0,0,0,0,0,0,0,0];

var radialcenter = (toothbaseradial / 2);

radial[7] = 0;

radial[8] = radialcenter - (toothcoreradial / 2);

radial[9] = radialcenter - (toothtipradial / 2);

radial[10] = radial[9] + toothtipradial;

radial[11] = radial[8] + toothcoreradial;

radial[12] = toothbaseradial;

radial[13] = radial[12] + toothgapradial;

// horizontal segments

for ( h=0; h<hsteps; h++ )

{

radius0 = radius1;

radius1 = radius0 + Math.tan(bevelangle) \* dh;

circ0 = circ1;

circ1 = 2.0 \* Math.PI \* radius1;

h0 = h1;

h1 = h0 + dh;

for ( i=0; i<7; i++ )

{

radial[i] = radial[i+7];

radial[i+7] = radial[i] + dhelix;

}

for ( tooth=0; tooth<teeth; tooth++ )

{

var x,y,z;

var point = [];

var origin = [];

var radius = [];

radius[0] = radius0;

radius[1] = radius0 + toothbaseheight;

radius[2] = radius0 + toothbaseheight + toothcoreheight;

radius[3] = radius[2];

radius[4] = radius[1];

radius[5] = radius0;

radius[6] = radius0;

radius[7] = radius1;

radius[8] = radius1 + toothbaseheight;

radius[9] = radius1 + toothbaseheight + toothcoreheight;

radius[10] = radius[9];

radius[11] = radius[8];

radius[12] = radius1;

radius[13] = radius1;

for ( i=0; i<7; i++ )

{

x = radius[i]\*Math.sin(radial[i]);

y = radius[i]\*Math.cos(radial[i]);

z = h0;

point[i] = [x\_pos+x,y\_pos+y,z\_pos+z];

x = radius[i+7]\*Math.sin(radial[i+7]);

y = radius[i+7]\*Math.cos(radial[i+7]);

z = h1;

point[i+7] = [x\_pos+x,y\_pos+y,z\_pos+z];

}

// Tooth

jellyQuad( mesh, point[0], point[1], point[8], point[7] );

jellyQuad( mesh, point[1], point[2], point[9], point[8] );

jellyQuad( mesh, point[2], point[3], point[10], point[9] );

jellyQuad( mesh, point[3], point[4], point[11], point[10] );

jellyQuad( mesh, point[4], point[5], point[12], point[11] );

// Gap

jellyQuad( mesh, point[5], point[6], point[13], point[12] );

if ( h===0 )

{

origin = [x\_pos+0,y\_pos+0,z\_pos+0];

// Tooth

jellyTriangle( mesh, origin, point[5], point[0] );

jellyQuad( mesh, point[0], point[5], point[4], point[1] );

jellyQuad( mesh, point[1], point[4], point[3], point[2] );

// Gap

jellyTriangle( mesh, origin, point[6], point[5] );

}

if ( h==(hsteps-1) )

{

origin = [x\_pos+0,y\_pos+0,z\_pos+height];

// Tooth

jellyTriangle( mesh, origin, point[7], point[12] );

jellyQuad( mesh, point[8], point[11], point[12], point[7] );

jellyQuad( mesh, point[8], point[9], point[10], point[11] );

// Gap

jellyTriangle( mesh, origin, point[12], point[13] );

}

for ( i=0; i<14; i++ )

radial[i] += (toothbaseradial + toothgapradial);

}

}

var solid = Solid.make(mesh);

return solid;

**}**