三维建模算法(三)

项道德（微信：daode1212） 2021-02-13

编程语言：C# + OpenTK

人们从使用负数、无理数，拓展到虚数、复数，看似抽象，其实作用不小。然而，现在可以用编程来研究一下，球的乘方与开方，球的幂运算，球的指数与对数运算、以及三角与反三角，双曲与反双曲等，它们究竟是什么图形？ 现在一一展现给你。

0.标准的球面方程是： x2+y2+z2=r2 ，但是，在三维编程中还是用极坐标来呈现方便。

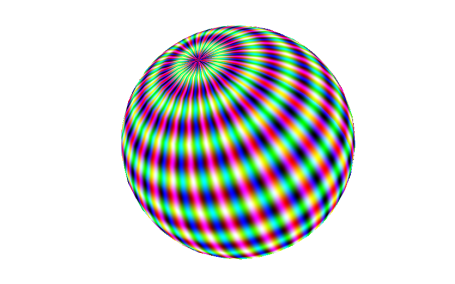
本文大多使用极坐标方程来表示及变换。

//三维建模体---球:[经度,a范围:0...+360, 纬度,b范围:-90...+90]

float x = (float)(Math.Cos(a) \* Math.Cos(b));//空间坐标x

float y = (float)(Math.Sin(a) \* Math.Cos(b));//空间坐标y

float z = (float)(Math.Sin(b));//空间坐标z



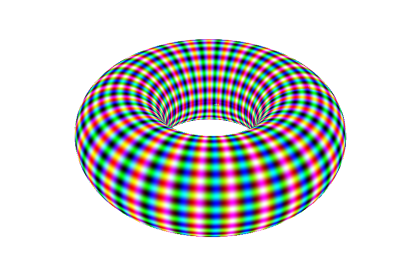
1.仅在X-Y平面上加入轨迹为圆的平移，可以生成环了：

//三维建模体---环:[经度, a范围:-180...+180, 纬度, b范围:-180...+180]

float x = (float)(Math.Cos(a) \*Math.Cos(b) + 2\*Math.Cos(a) );//空间坐标x

float y = (float)(Math.Sin(a) \*Math.Cos(b) + 2\*Math.Sin(a) );//空间坐标y

float z = (float)(Math.Sin(b));//空间坐标z



2.进一步，再在Z方向，加上a/6偏移，可以生成螺旋管了：

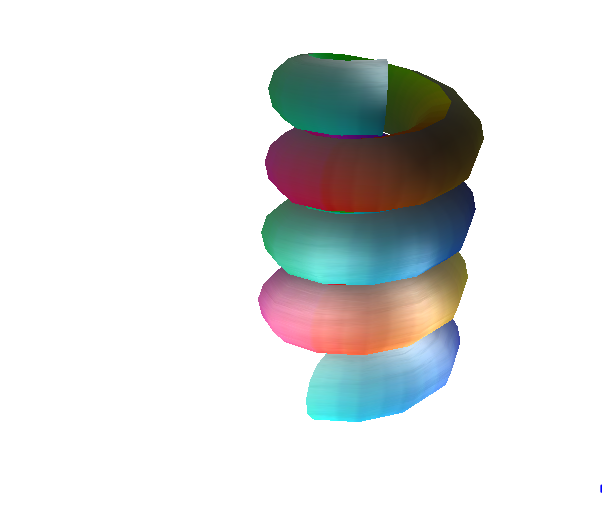
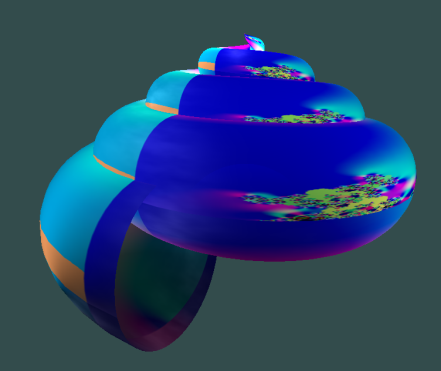
//三维建模体---螺旋管:[经度,a范围:-720...+720,纬度, b范围:-180...+180]

float r = 0.5f;

float x = (float)(r\*Math.Cos(a) \* (Math.Cos(b)+2));//空间坐标x

float y = (float)(r\*Math.Sin(a) \* (Math.Cos(b)+2));//空间坐标y

float z = (float)(a/6+r\*Math.Sin(b));//空间坐标z

拓展之一：（螺蛳）

3.开N次方，大多会变方方的，四个角却是圆角的。负数不能开方？你一定想到了虚数与复数，但是，我却偏偏不用它，我采用符号分离的方式处理：

//三维建模体--- 方环[环的开方]:[经度范围a:-180...+180,纬度范围b:-180...+180]

float x0 = (float)(Math.Cos(a) \* (Math.Cos(b) + 2));//空间坐标x

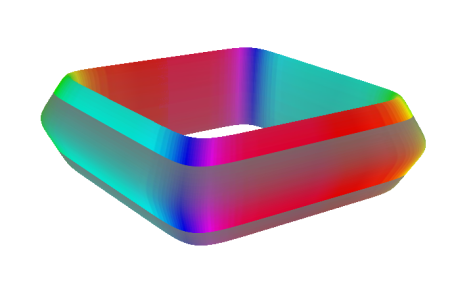
float y0 = (float)(Math.Sin(a) \* (Math.Cos(b) + 2));//空间坐标y

float z0 = (float)(Math.Sin(b));//空间坐标z

float x = (float)(Math.Sign(x0) \* Math.Pow(x0\*x0, 0.1f));//

float y = (float)(Math.Sign(y0) \* Math.Pow(y0\*y0, 0.1f));//

float z = z0/(.1f+x0\*x0+y0\*y0);



4.有意思的是：球的开N次方是圆角立方体，球的乘方是六锐角体：

//三维建模体--- 圆角立方体:[经度, a范围:0...+360,纬度, b范围:-90...+90]

float x0 = (float)(Math.Cos(a) \*Math.Cos(b));//空间坐标x

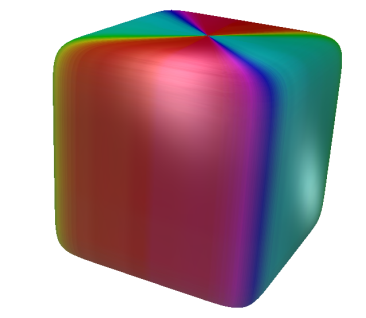
float y0 = (float)(Math.Sin(a) \* Math.Cos(b));//空间坐标y

float z0 = (float)(Math.Sin(b));//空间坐标z

float x = (float)(Math.Sign(x0) \* Math.Pow(x0 \* x0, .1));

float y = (float)(Math.Sign(y0) \* Math.Pow(y0 \* y0, .1));

float z = (float)(Math.Sign(z0) \* Math.Pow(z0 \* z0, .1));

这是开5次方的效果。

5.乘方运算，锐角化了：

//三维建模体--- 六锐角体:[经度,a范围:0...+360,纬度,b范围:-90...+90]

float x0 = (float)(Math.Cos(a) \* Math.Cos(b));//空间坐标x

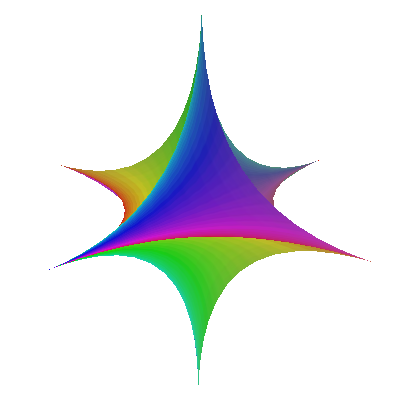
float y0 = (float)(Math.Sin(a) \* Math.Cos(b));//空间坐标y

float z0 = (float)(Math.Sin(b));//空间坐标z

float x = (float)(Math.Sign(x0) \* Math.Pow(x0 \* x0, 2));

float y = (float)(Math.Sign(y0) \* Math.Pow(y0 \* y0, 2));

float z = (float)(Math.Sign(z0) \* Math.Pow(z0 \* z0, 2));



6.空间有另一类曲线，封闭而无交点的，如微观粒子运动的轨迹，叫它们为“空间纽结”：

空间纽结有着独特的算法，需单独研究与描述，这里仅展示一例。

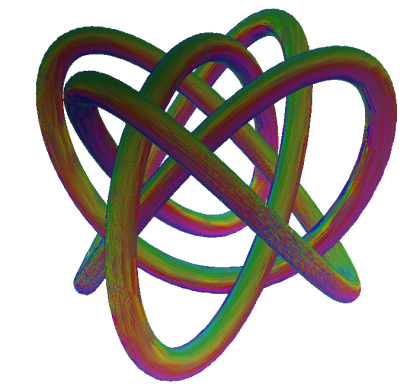
d += 0.05f;// d范围:0...12032

//三维建模体---空间纽结:[经度a范围:-180...+180,纬度b范围:-180...+180]

float x = (float)(Math.Sin(d \* 3) + .1 \* Math.Cos(a) \* Math.Cos(b));//空间坐标x

float y = (float)(Math.Cos(d \* 5) + .1 \* Math.Sin(a) \* Math.Cos(b));//空间坐标y

float z = (float)(Math.Sin(d \* 4) + .1 \* Math.Sin(b));//空间坐标z



7.现在，看看球的倒数，可以变换成一张桌子：

//三维建模体--- 桌子:[经度范围a:-180...+180,纬度范围b:0...+180]

float x0 = (float)(Math.Cos(a) \* Math.Cos(b));//空间坐标x

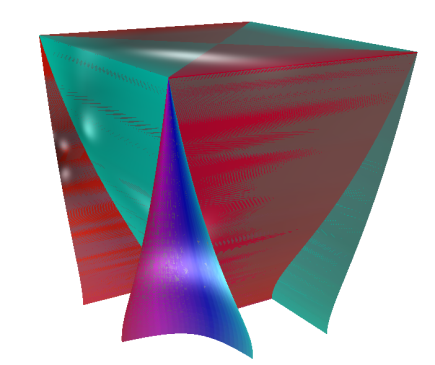
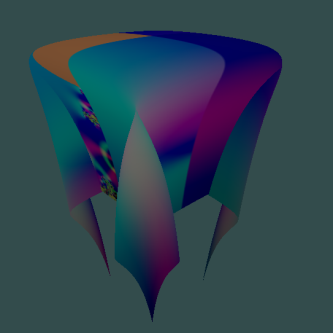
float y0 = (float)(Math.Sin(a) \* Math.Cos(b));//空间坐标y

float z0 = (float)(Math.Sin(b));//空间坐标z

float x = (float)(Math.Sign(x0) \* Math.Pow(.5+x0 \* x0, -1f));//

float y = (float)(Math.Sign(y0) \* Math.Pow(.5+y0 \* y0, -1f));//

float z = z0 / (.25f + x0 \* x0 + y0 \* y0);

拓展之一：

8.再将球进行指数运算，是魔方格箱子

//三维建模体--- 魔方格箱子:[经度范围a:-180...+180,纬度范围b:-180...+180]

float x0 = (float)(Math.Cos(a) \* Math.Cos(b));//空间坐标x

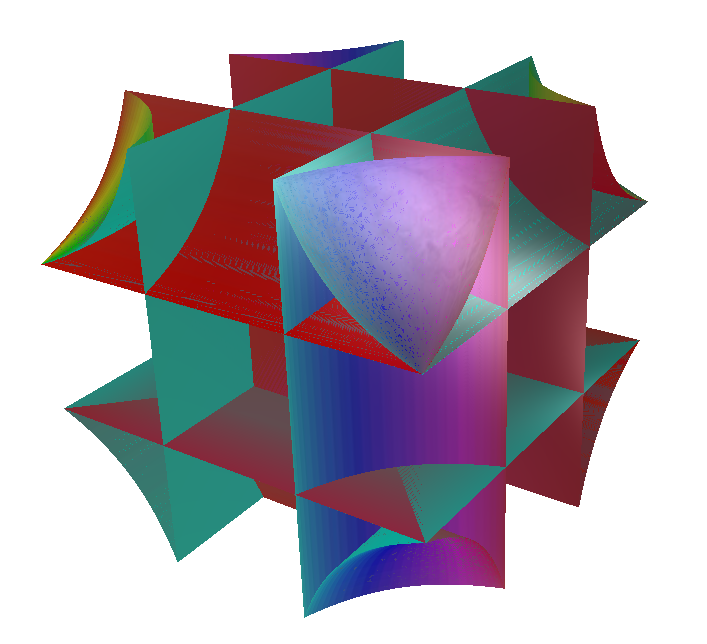
float y0 = (float)(Math.Sin(a) \* Math.Cos(b));//空间坐标y

float z0 = (float)(Math.Sin(b));//空间坐标z

float x = (float)(Math.Sign(x0) \* Math.Exp(-x0 \* x0));//

float y = (float)(Math.Sign(y0) \* Math.Exp(-y0 \* y0));//

float z = (float)(Math.Sign(z0) \* Math.Exp(-z0 \* z0));//



9.球的反正切变换+对数变换，形成如炮弹或导弹的三维对象：

//三维建模体--- 飞弹:[经度范围a:-180...+180,纬度范围b:-180...+180]

float x0 = (float)(Math.Cos(a) \* Math.Cos(b));//空间坐标x

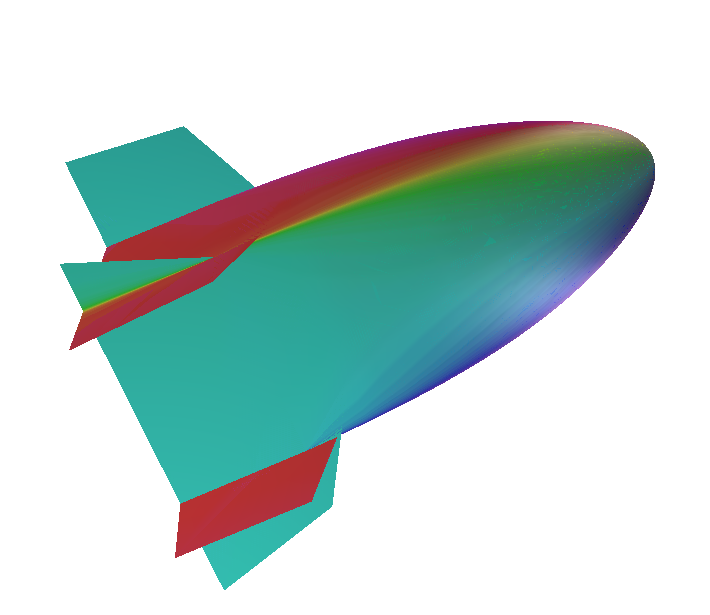
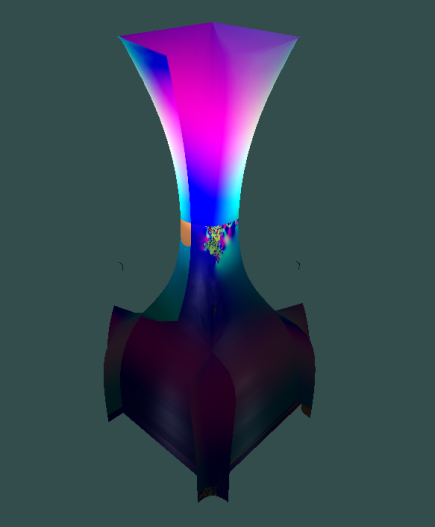
float y0 = (float)(Math.Sin(a) \* Math.Cos(b));//空间坐标y

float z0 = (float)(Math.Sin(b));//空间坐标z

float x = (float)( Math.Atan2(x0, z0 ));//

float y = (float)( Math.Atan2(y0, z0 ));//

float z = (float)( 2\*Math.Log(.05+z0));//

拓展之一：

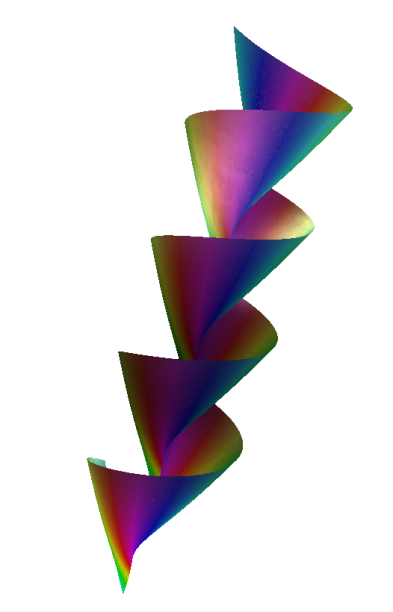
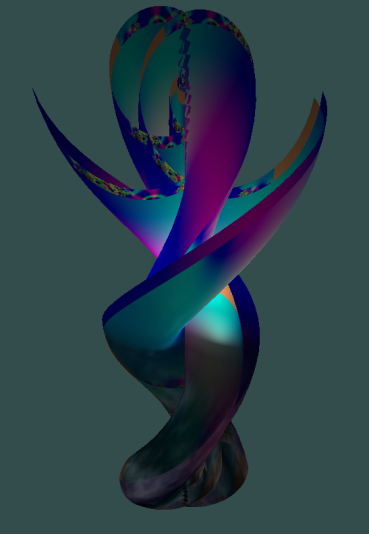
10.麻花绳，或麻花钻，也是在球的公式中变换而来的：

//三维建模体---麻花绳，或麻花钻:[经度范围a:-720...+720,纬度范围b:-90...+90]

float x = (float)(Math.Cos(a) \* Math.Cos(b));

float y = (float)(Math.Sin(a) \* Math.Cos(b));

float z = (float)(a/4+b\*Math.Sin(b));

拓展之一：

11.经过适当的迭代，会生成小木船：

//三维建模体--- 小木船:[经度范围a:-180...+180,纬度范围b:-180...+180]

float xx = (float)(Math.Cos(a) \* Math.Cos(b));

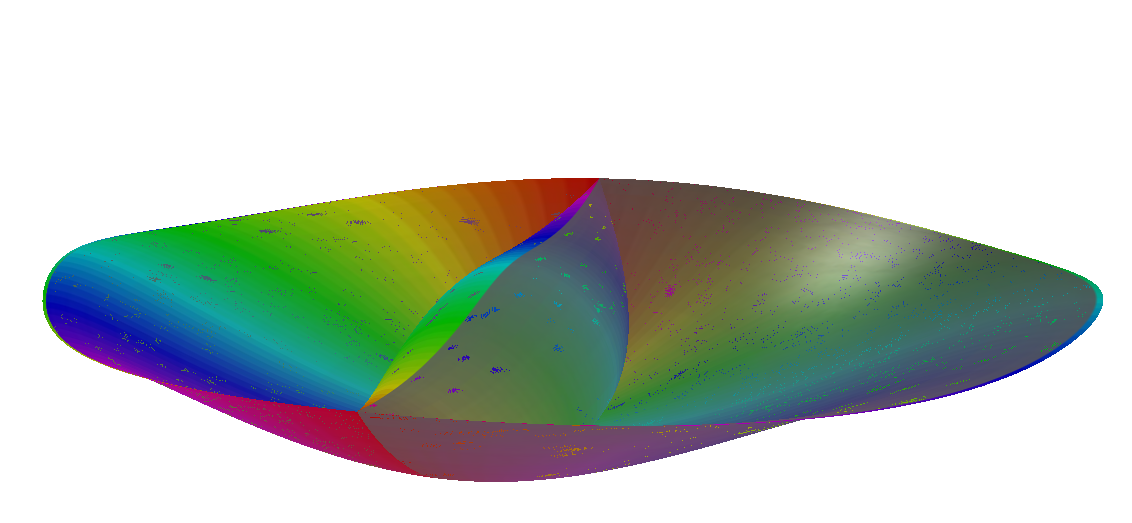
float yy = (float)(Math.Sin(a) \* Math.Cos(b));

float zz = (float)(Math.Sin(b));

float x = (float)(Math.Cos(xx) \* Math.Cos(yy));

float y = (float)(Math.Sin(xx) \* Math.Cos(yy));

float z = (float)(b\*Math.Sin(yy)/2);



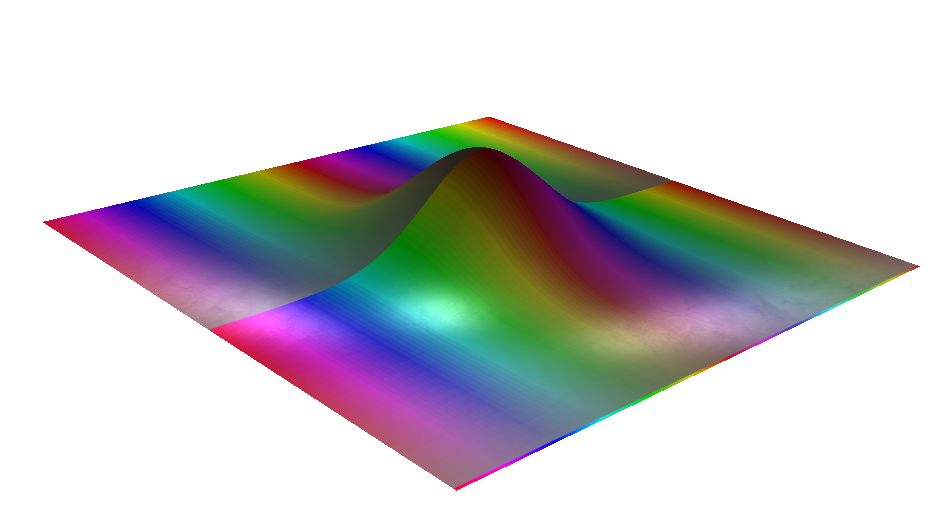
12.在学习概率统计时，常常提及正态分布、高斯分布，二维时就形如这里的指数凹凸面：

//三维建模体--- 指数凹凸面:[x范围:-PI...+PI,y范围:-PI...+PI]

float x = (float)(a);

float y = (float)(b);

float z = (float)(Math.Exp(-a\*a-b\*b));



13.运用高次幂，也可作一些方形的建模，与冲压件相像：

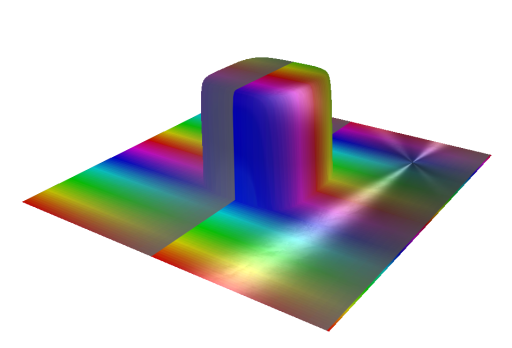
//三维建模体--- 高次幂凹凸面:[x范围:-PI...+PI,y范围:-PI...+PI]

float x = (float)(a);

float y = (float)(b);

float z = (float)Math.Pow(a\*a\*a\*a+b\*b\*b\*b,10);

z = z > 2 ? 2 : z;



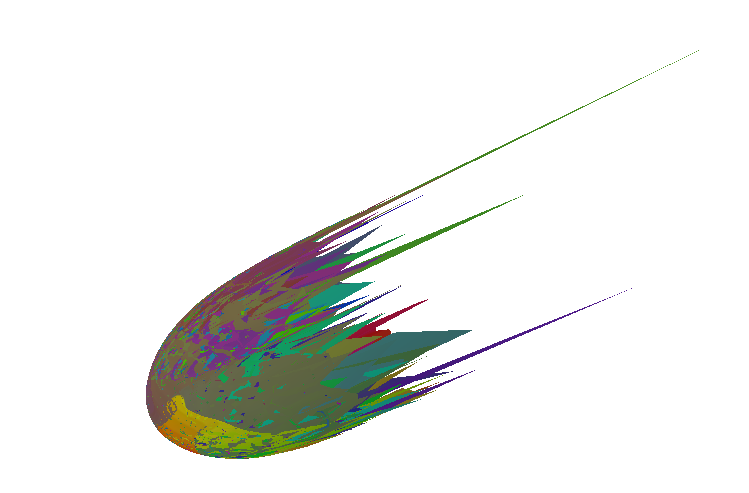
14.彗星，有条不规则的尾巴，引入了一个反双曲函数：

//三维建模体--- 彗星:[x范围:-PI...+PI,y范围:-PI...+PI]

float x = (float)(b+a\*Math.Cos(a\*7));

float y = (float)(a+b\*Math.Sin(b\*7));

float z = (float)Math.Atanh(-x\*x-y\*y);



15.反正切与高次幂的引入，可以生成凳子：

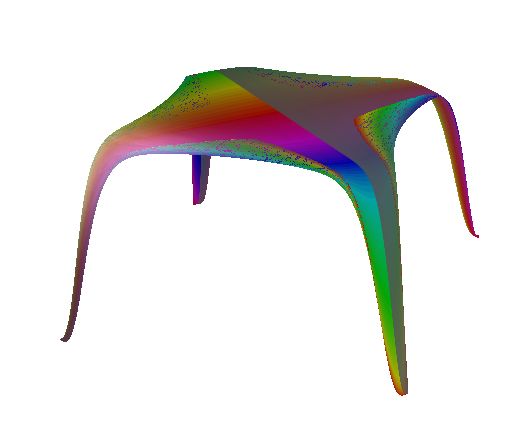
//三维建模体--- 塑料凳子:[x范围:-PI...+PI,y范围:-PI...+PI]

float x = (float)Math.Atan2(b, .98+a\*a);

float y = (float)Math.Atan2(a, .98+b\*b);

float z = (float)Math.Sin(x \* x + y \* y);

z = (float)Math.Pow(z,12);



16.稍稍改动下参数，聚宝盆生成了：

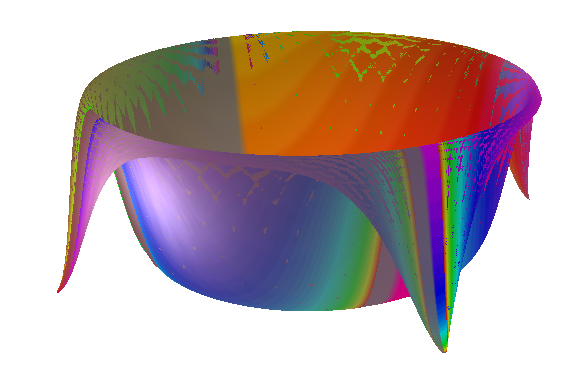
//三维建模体--- 聚宝盆:[x范围:-PI...+PI,y范围:-PI...+PI]

float x = (float)Math.Atan2(b, .15 + a \* a);

float y = (float)Math.Atan2(a, .15 + b \* b);

float z = (float)Math.Sin(x \* x+y \* y);

z = (float)Math.Pow(z, 12);



我们在工程三维仿真的时候，通常需要做很多的立体件，我们平时都用的其他三维建模软件，如Maya、Rhino、UG、pro-E、AutoCAD、SolidWorks、LabVIEW、3d-max等，但有了这些算法，以后我们的基础三维件就可以通过这些算法直接生成，不需要在别的软件中逐一绘制了。