

ShapeExplorer version 4

User Manual

ShapeExplorer (Fig. 1) is an interactive software tool designed to display shapes defined by *parametric* and *implicit* function scripts and colors defined by *explicit* functions. It can run on Windows, MacOS and Linux (Ubuntu) computers. Shape Explorer is just one interactive window where you can type the definitions scripts and other parameters. The definitions can be saved in a proprietary format and loaded later to the software to continue working with them. Shape Explorer supports only one shape visualization at the time. Its purpose is to work as a quick all-in-one multi-platform visualization tool.

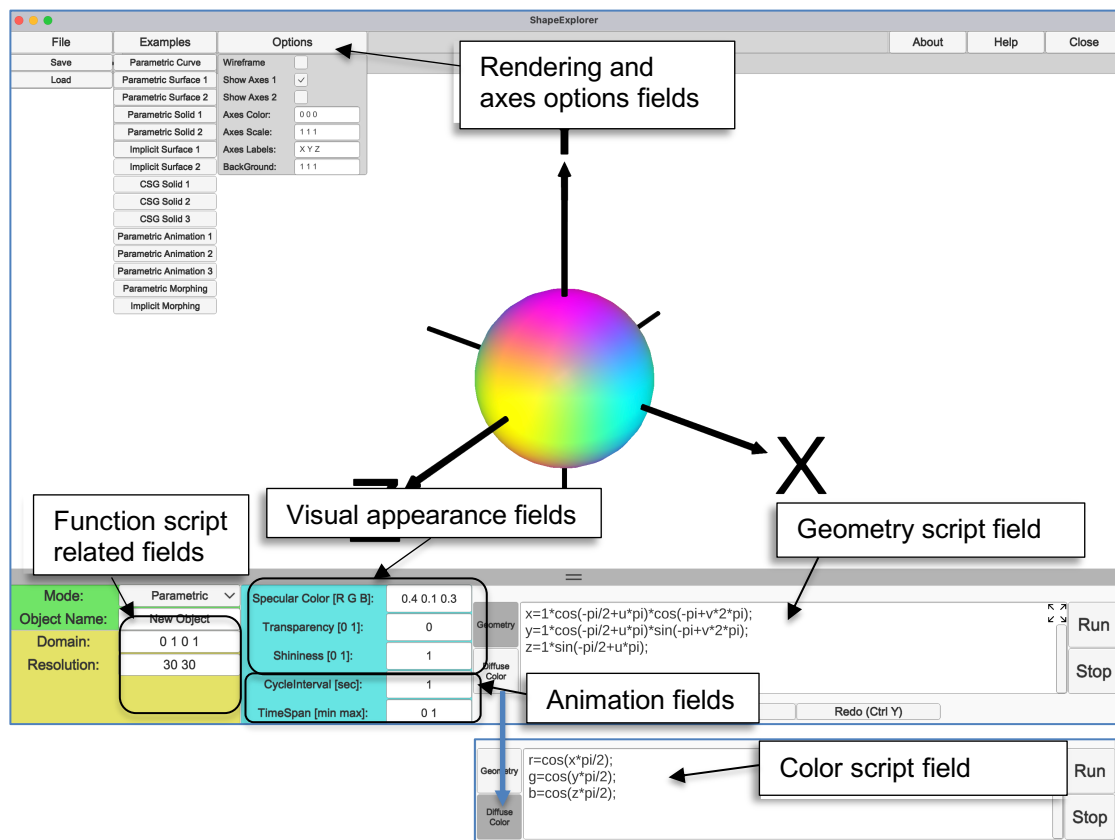


Figure 1. ShapeExplorer at a glance.

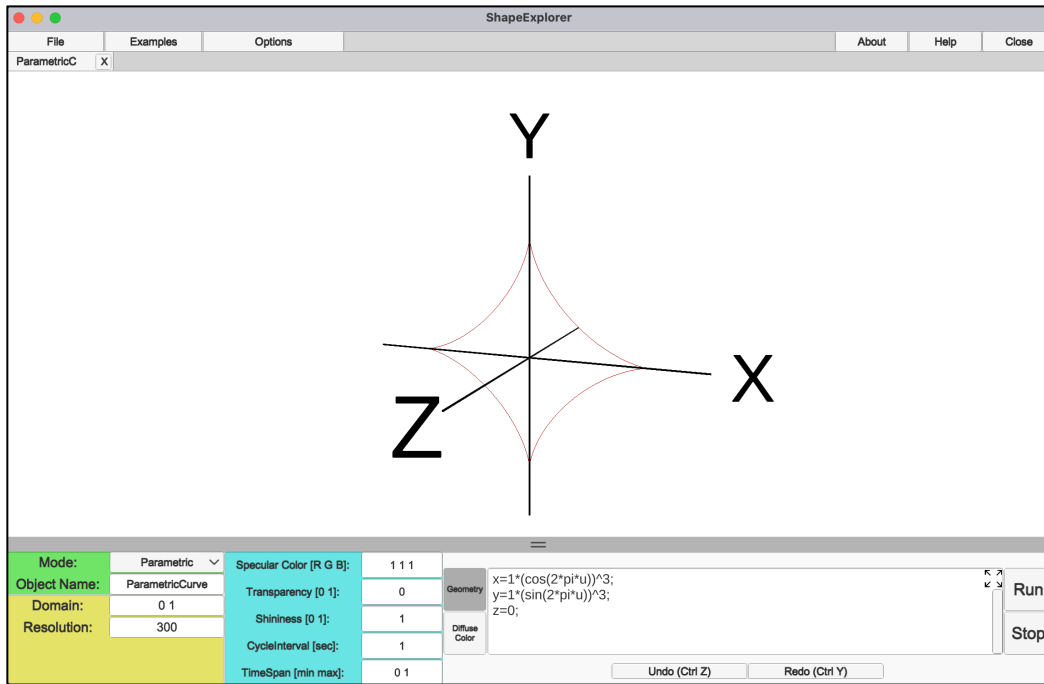


Figure 2. Displaying a parametric curve.

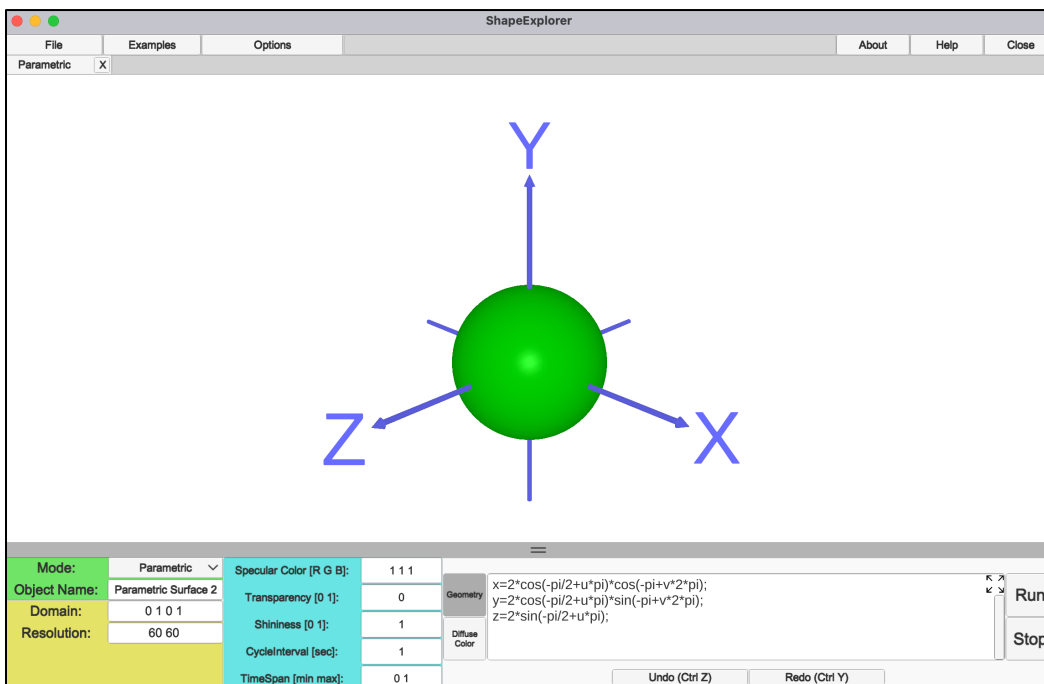


Figure 3. Displaying a parametric surface.

Parametric functions in ShapeExplorer are explicit functions of up to three variables u, v, w (which are called parameters) and time t . They can define Cartesian coordinates x, y, z of curves, surfaces and solid objects. To define a curve, only one parameter u, v or w has to be used, to define a surface—any two parameters u, v, v, w or u, w are required, for solid objects—all three parameters u, v, w have to be used. When t is added, these objects will become time-dependent. The examples of defining parametrically curves, surfaces and solid objects are given in Figs. 2, 3 and 4.

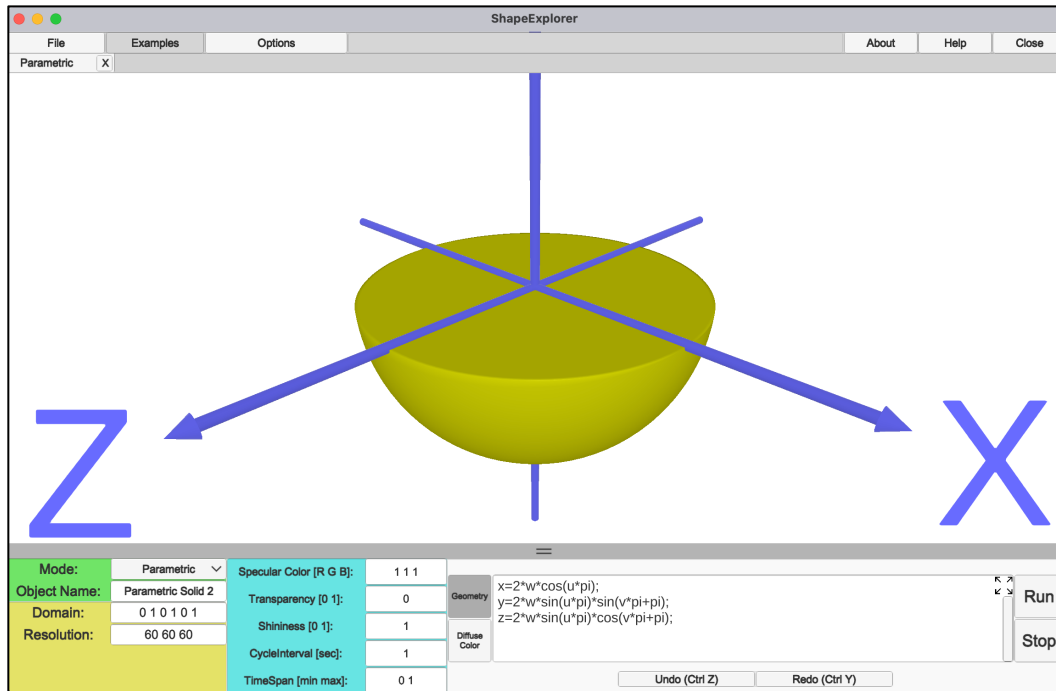


Figure 4 Displaying a parametric solid.

Implicit functions are the functions defined as $f(x, y, z, t) = 0$, where x, y, z are Cartesian coordinates and t is the time. You will only use them for defining surfaces. The implicit functions are equal to zero for the points located on the surface. By changing this equality into an inequality $g = f(x, y, z, t) \geq 0$, known in computer graphics as *FRep*, we define not only a surface but the space bounded by this surface, or a half-space. In this case, the function equals to zero for the points located on the surface, positive values of the function indicate points inside the solid object, and negative values are for the points which are outside the object. In the ShapeExplorer script field, only the left part of the implicit or *FRep* function has to be written – it is always assumed to be ≥ 0 . The example of defining a surface is given in Fig. 5 and a definition of the solid object using set-theoretic (Boolean) operations is illustrated in Fig. 6.

Defining shapes and animations may require multiple formulas and temporary variables. ShapeExplorer supports a subset of C# language for writing definition scripts. The following mathematical functions are supported:

abs, sqrt, exp, log, sin, cos, tan, acos, asin, atan, ceil, floor, atan2, mod, round, max, min, cosh, sinh, tanh, log10.

There is also *if{ } else { }* operator. Variables x, y, z are reserved for Cartesian coordinates, while variables u, v, w are parametric coordinates. Variable t is reserved for defining the time. All other variable can be used without declaration—they will be declared as *float*.

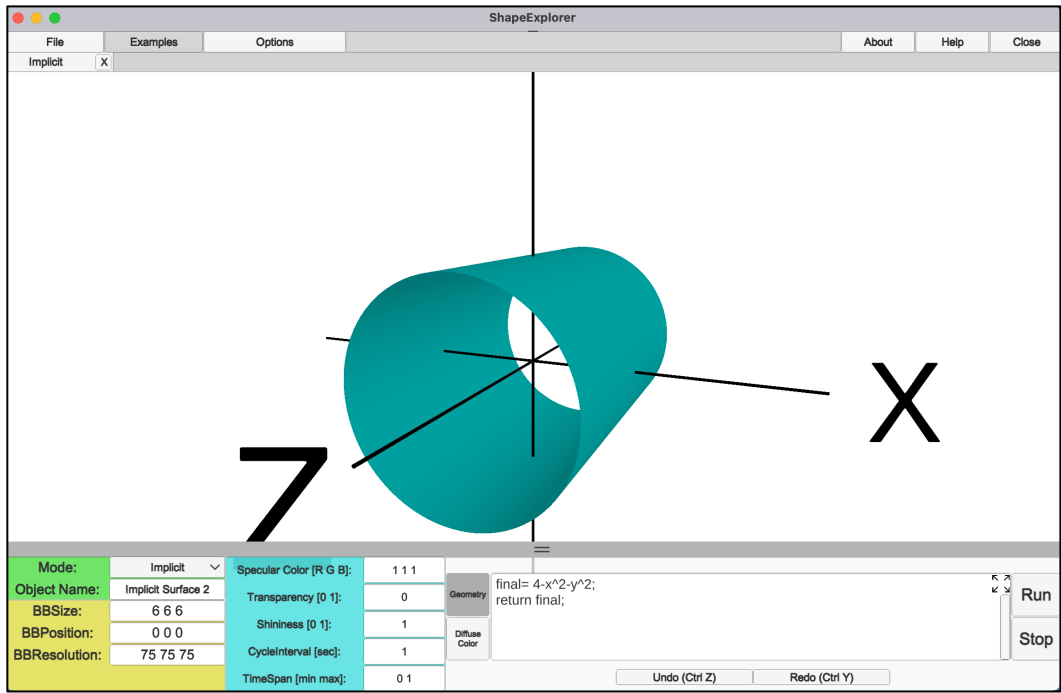


Figure 5 Displaying an implicit surface. Both sides of the surface are visible.

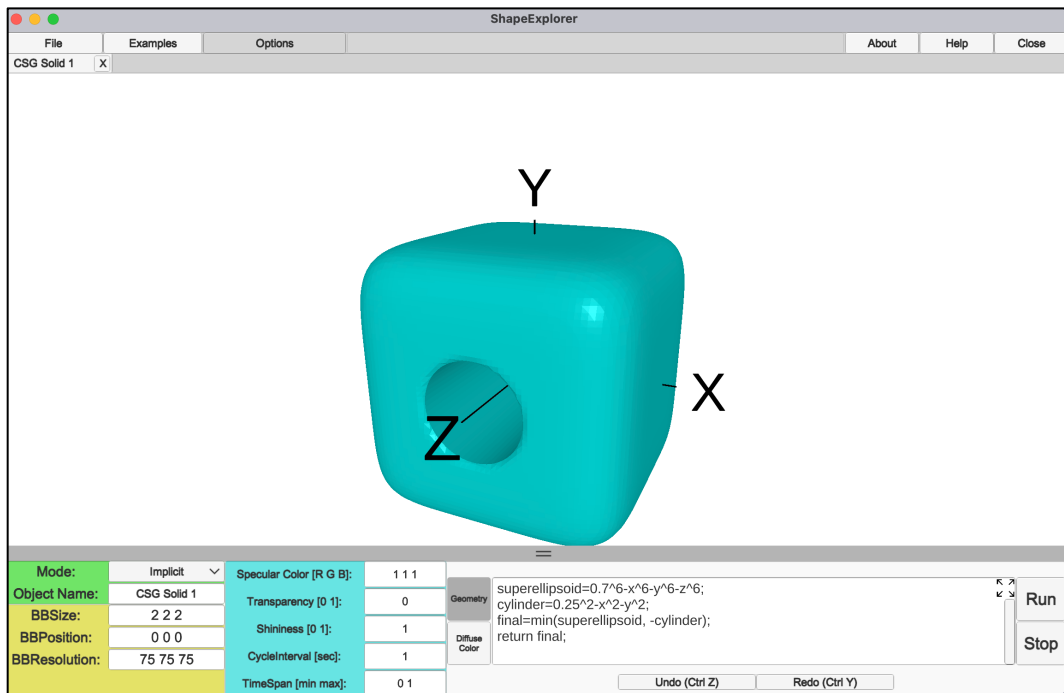


Figure 6 Displaying an implicit solid. The inner surface of the object is not visible in solids.

For defining r, g, b diffuse colors, scripts with explicit functions of coordinates x, y, z have to be used:

$r|g|b = f(x, y, z, t)$; $r, g, b \in [0,1]$. Thus the color is defined for any point of the 3D modelling space and it is then sampled by the geometry (see Fig. 7). Constant colors can also be defined as $r|g|b = value \in [0,1]$;

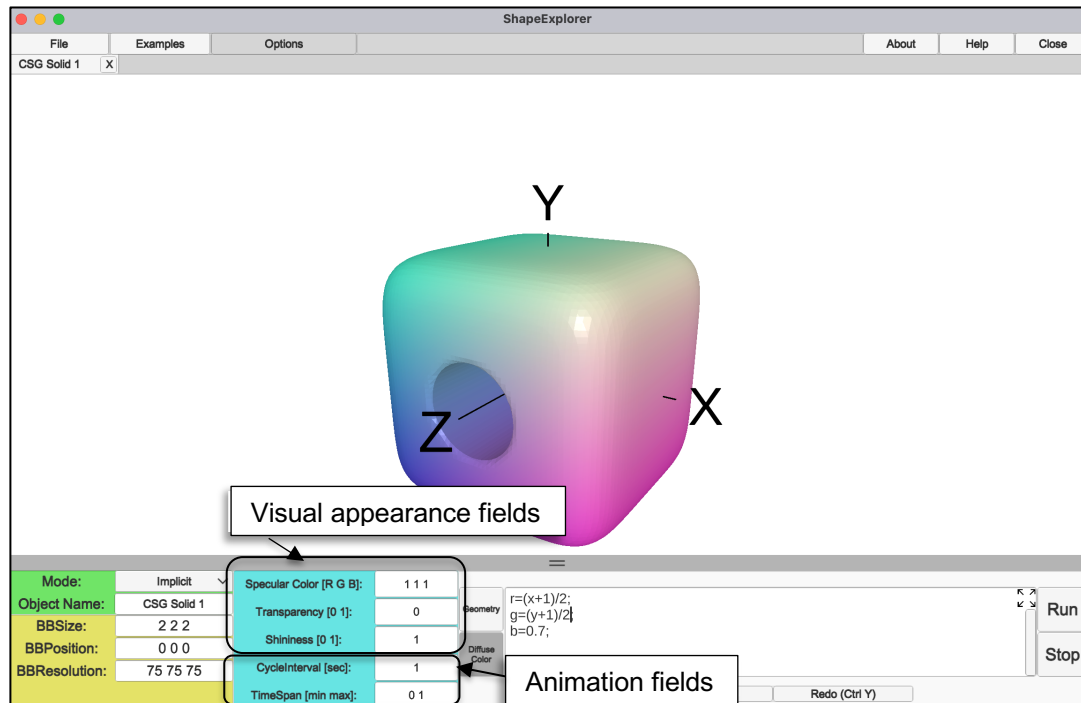


Figure 7 Defining diffuse colors as functions of coordinates.

In addition, in the Visual Appearance fields the *Specular Color* [0,1], *Shininess* [0,1] and *Transparency* [0,1] values can be defined. *Specular Color* affects overall illumination of the object. It is responsible for displaying reflections or specs on the surface of objects. Parameter *Shininess* controls the size of such specs. Colors of curves can be only defined if *Shininess* is set to 1.

The remaining Animation fields (Fig. 7) contain *TimeSpan* and *CycleInterval*. The *TimeSpan* contains minimum and maximum values of the time values between which it will be incremented in a cyclic manner. These values can be positive and negative. The *TimeSpan* interval then is mapped to the real time interval *CycleInterval* in seconds – this is how the animation will be then displayed.

Options fields (Fig. 7) contain parameters of the coordinate axes, an option to toggle to wireframe visualization mode, and background color. Coordinate axes can be scaled—shorten with the scaling parameters less than 1 or elongated with the scaling parameters greater than 1. Scaling coefficient 0 will toggle off the respective axis.