

# SingSurf

## A program to generate Symmetry Sets and other curves.

SingSurf is a java program to calculate various curves and surface in 2D and 3D defined by mathematical equations. It can calculate

- parametrised curves
- algebraic curves defined by polynomial equations
- parametrise surfaces
- algebraic surfaces
- the intersection of a curve or surface with a set defined by an equation
- the result of applying a mapping to a curve or surface
- intersections and mappings where the definitions depend on the definitions of other curves or surfaces.

This last class allows the calculations of objects studied in singularity theory. For instance it can find the pre-Symmetry Set of a curve by using an intersection of the parameter space with a definition which depends on a particular curve. A mapping can then be applied to construct the Symmetry Set. A similar approach can be used to calculate parabolic lines and ridges of surfaces.

## Installation and Requirements

The package is supplied in a zip file. SingSurf.zip. Unpack in a directory.

It requires Java to run. Java can be obtained from <https://java.com/en/download/>

The program uses the Javaview 3D geometry viewer to operate. The required Javaview files are included in the zip file but you should obtain a licence file from the <http://www.javaview.de/download/registration.html>

## Running

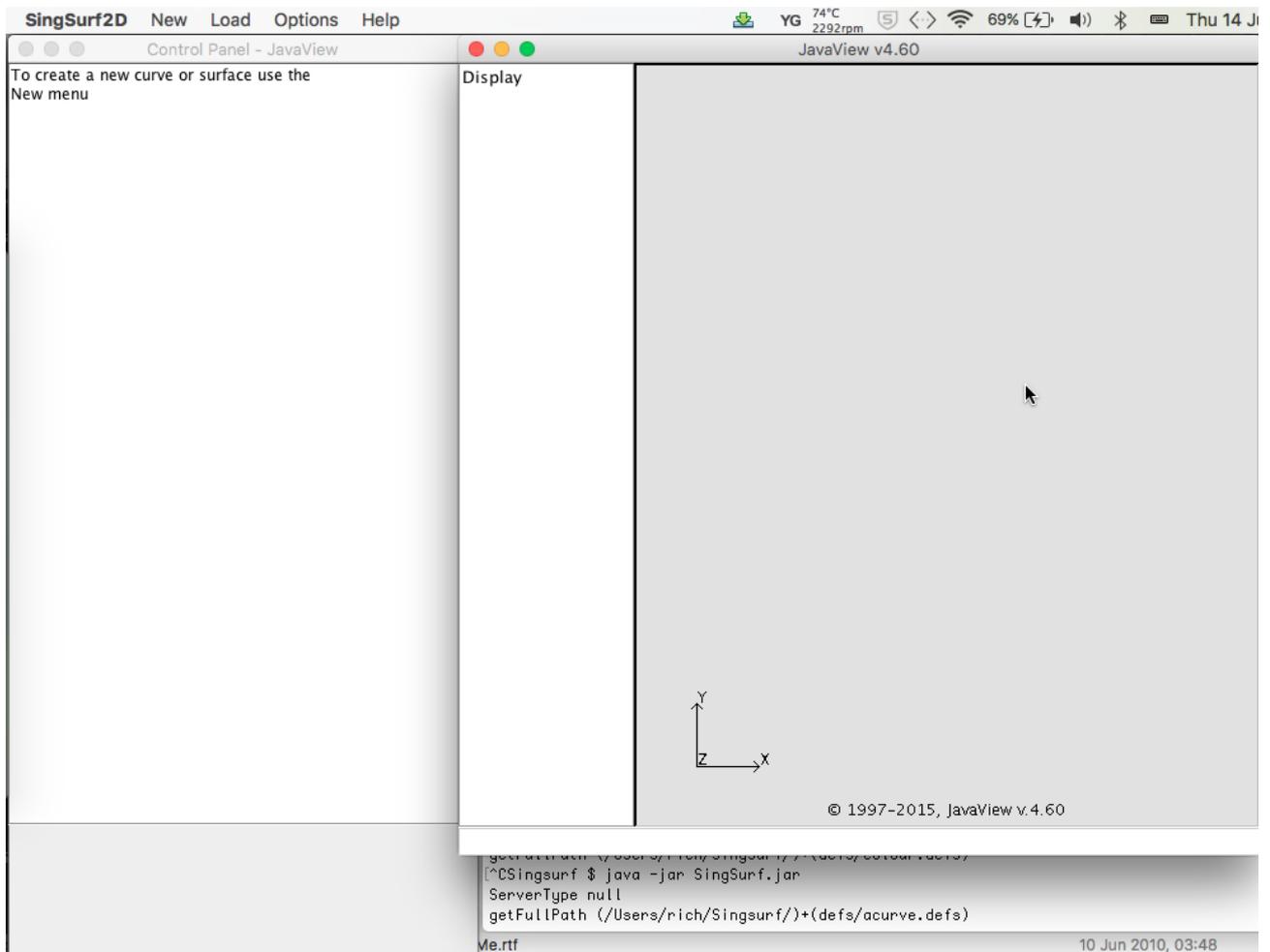
Running the program is quite simple, for the 2D version just change to the

```
java -jar SingSurf.jar
```

The program can also be used in 3D mode, to start this use

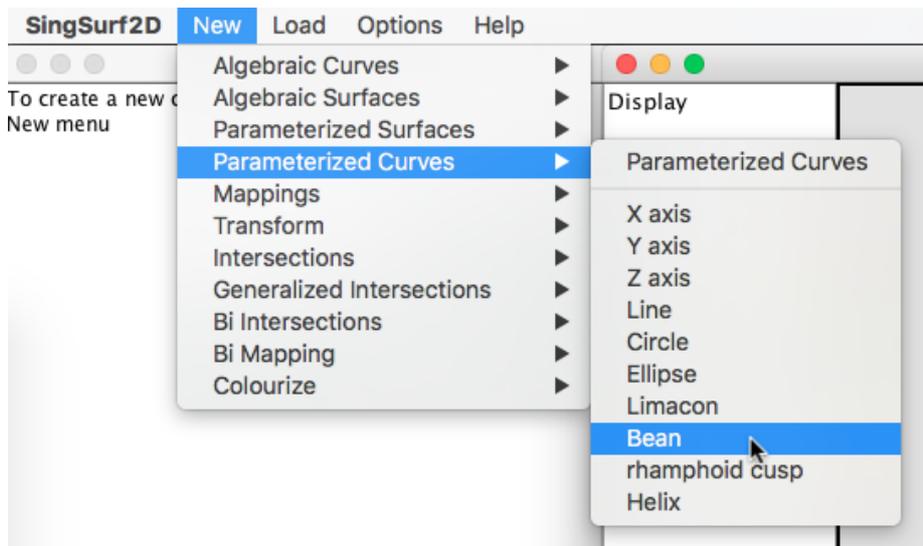
```
java -cp SingSurf.jar org.singsurf.singsurf.SingSurfPro
```

The opening screen is

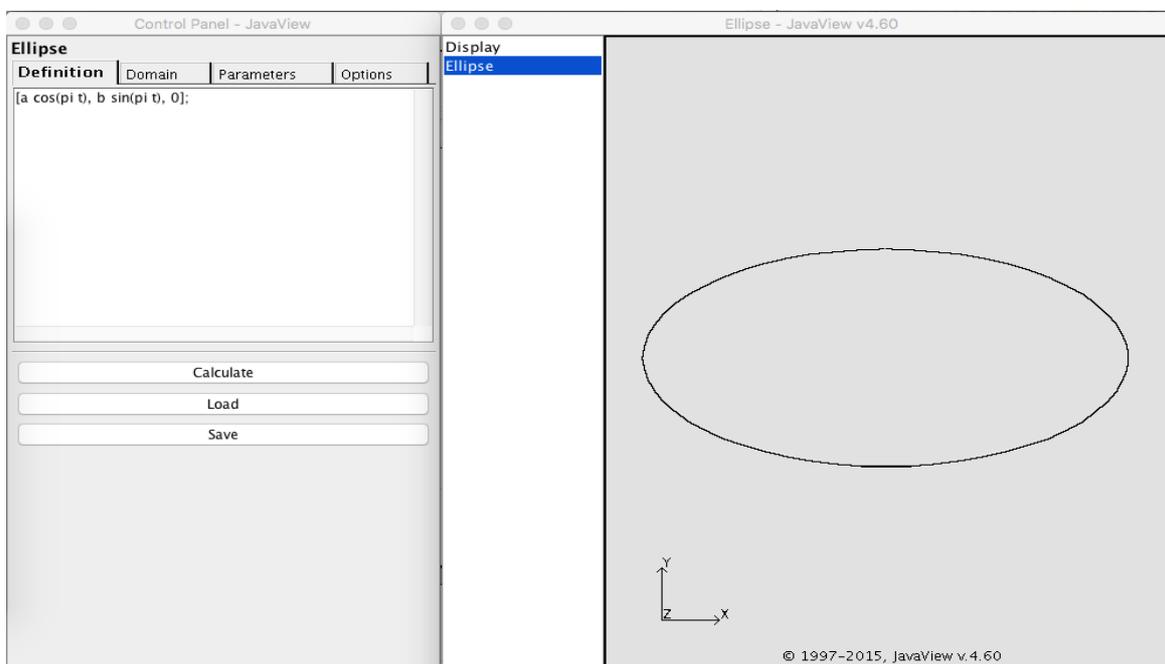


The left hand panel is where various controls are displayed and the right hand panel displays the actual objects. In the display panel the sidebar allows different curves and surfaces to be selected.

Two different set of menu are displayed. If the right hand displayed panel is selected then the above menu is displayed. The main item here is the New menu which allows new curves and surfaces to be created. For each type of object a set of predefined definitions is given.



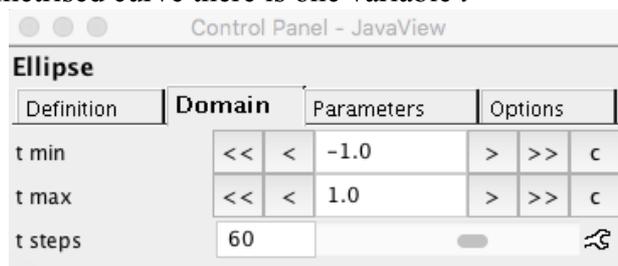
When a new definition is loaded the Control Panel will show the definition of the curve



There are several tabs here.

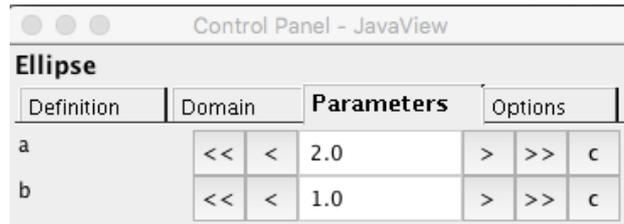
The first tab "Definition" gives the equations which defines the curve. The format of these equations is discussed below.

The second tab "Domain" define the domain of the variables. These depend on the type of curve/surface. For a parametrised curve there is one variable  $t$

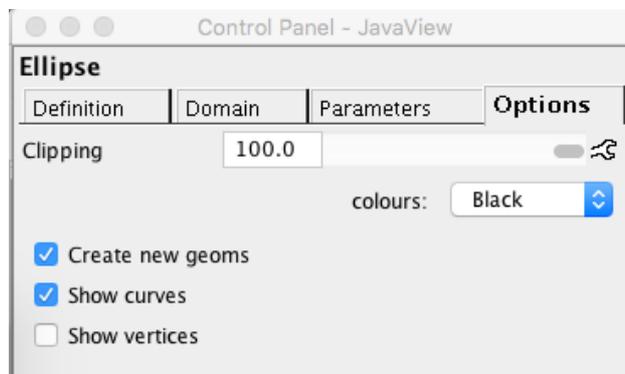


with one control for the minimum value, one for the maximum value and one for the number of steps.

The third tab "Parameters" contains values for different parameters which the definition depends on. For the ellipse example there are two parameters  $a$  and  $b$ .



The final "options" tab allows different options for the curve or surface



The clipping value is used to clip surfaces which extend to infinity.

The colour of the curve can be set, as which part of the output geometry is displayed. For curves the actual curve segments and vertices can be displayed. For surfaces the faces, edges and vertices can be displayed.

## Syntax for definition equations

The system for parsing the equations is built using the Java Expression Package <http://www.singularsys.com/jep/>. It is currently using the older Jep 2.4 version with plans to update it to the newer Jep 3.4 release. This change does not affect the syntax.

The syntax follow normal mathematical rules. Variables can be single or multiple characters with alphanumeric characters and underscores. Powers are denoted using the  $\wedge$  symbol and multiplication can be implied with a space, i.e.  $x^2 + 2 x y + y^2$ . Variable names are reserved for common constants:  $\pi$ ,  $e$  and  $i$ .

Vectors are specified like  $[x, y, x^2 - y^2]$  they typically have three components. The dot product is indicated by a  $\cdot$  and the cross product by  $\wedge$ . Vectors can be treated as complex numbers allowing multiplication and division. Here the third component of a vector is ignored so  $[a, b, 0] * [c, d, 0]$  is the vector  $[a*c - b*d, a*d + b*c, 0]$ . This is useful for defining the Symmetry set.

Most standard mathematical functions are defined:  $\sin(x)$ ,  $\cos(x)$ ,  $\tan(x)$ ,  $\text{asin}(x)$ ,  $\text{acos}(x)$ ,  $\text{atan}(x)$ ,  $\text{atan2}(y,x)$ ,  $\sinh(x)$ ,  $\cosh(x)$ ,  $\tanh(x)$ ,  $\text{asinh}(x)$ ,  $\text{acosh}(x)$ ,  $\text{atanh}(x)$ ,  $\log(x)$ ,  $\ln(x)$ ,  $\exp(x)$ ,  $\text{pow}(x,n)$ ,  $\text{sqrt}(x)$ ,  $\text{abs}(x)$ ,

mod(x,y), sum(a,b,c,d), rand(), round(x), floor(x), ceil(x). There is also an if function with a three argument form: if(cond,positive\_val,negative\_val) and four argument form if(cond,pos\_val,neg\_value,zero\_value).

Complex equation can be specified as a sequence of equations in a top down fashion, for example a Kummer Surface can be define as the algebraic surface

```
(3-v^2) ( x^2+y^2 + z^2 - v^2)^2 - (3 v^2 - 1) p q r s - 0;  
p = 1 - z - x rt2;  
q = 1 - z + x rt2;  
r = 1 + z + y rt2;  
s = 1 + z - y rt2;  
rt2 = sqrt(2);
```

The first equation is the main equation and subsequent equations are substituted into the equation above. Each equation is terminated by a semi-colon and subsequent equations must be of the form variable = .....;

It is possible to use symbolic differentiation in definition. The diff(f,x) operator calculate the derivative of f with respect to x.

## Types of curves and surfaces

Each type of curve and surface is specified by a different module which has different type of equation and different controls.

### Algebraic curves

These are specified by a single expression in two variables, x and y by default. For example

```
(x - x0)^2 + (y - y0)^2 - r^2;
```

Points on the curve are specified when this expression evaluates to zero.

The resolution of this curve is specified by the **resolution** options on the Option pane. These specify the number of divisions in the x and y directions.

### Algebraic surface

These are specified by a single expression in three variables, x, y and z by default. For example

```
x^2 + y^2 - z^2;
```

Points on the surface are specified when this expression evaluates to zero.

The resolution of this curve is specified by the **resolution** options on the Option pane. These specify the number of divisions in the x, y and z directions. These must be a power of two. Due to the way the algorithm works it is better if symmetry is broken when selecting the domain. Hence instead of the domain for x being [-1,1] results are improved if the domain is [-1,13,10,2].

## Algebraic surface

These are specified by a single expression in three variables,  $x$ ,  $y$  and  $z$  by default. For example

$$x^2 + y^2 - z^2;$$

Points on the surface are specified when this expression evaluates to zero.

The resolution of this curve is specified by the **resolution** options on the Option pane. These specify the number of divisions in the  $x$ ,  $y$  and  $z$  directions. These must be a power of two. Due to the way the algorithm works it is better if symmetry is broken when selecting the domain. Hence instead of the domain for  $x$  being  $[-1,1]$  results are improved if the domain is  $[-1,13,10,2]$ .

## Parametric curves

These are specified by a single 3D vector expression in a single variables,  $x$  by default. For example

$$[x, x^2, x^3];$$

The range of values for  $x$  is specified on the domain tab, and the number of steps on the same tab.

## Parametric surface

These are specified by a single 3D vector expression in two variables,  $x$ ,  $y$  by default. For example

$$[x, y, A x^2 + B x y + C y^2];$$

The range of values for  $x$ ,  $y$  are specified on the domain tab, and the number of steps on the same tab.

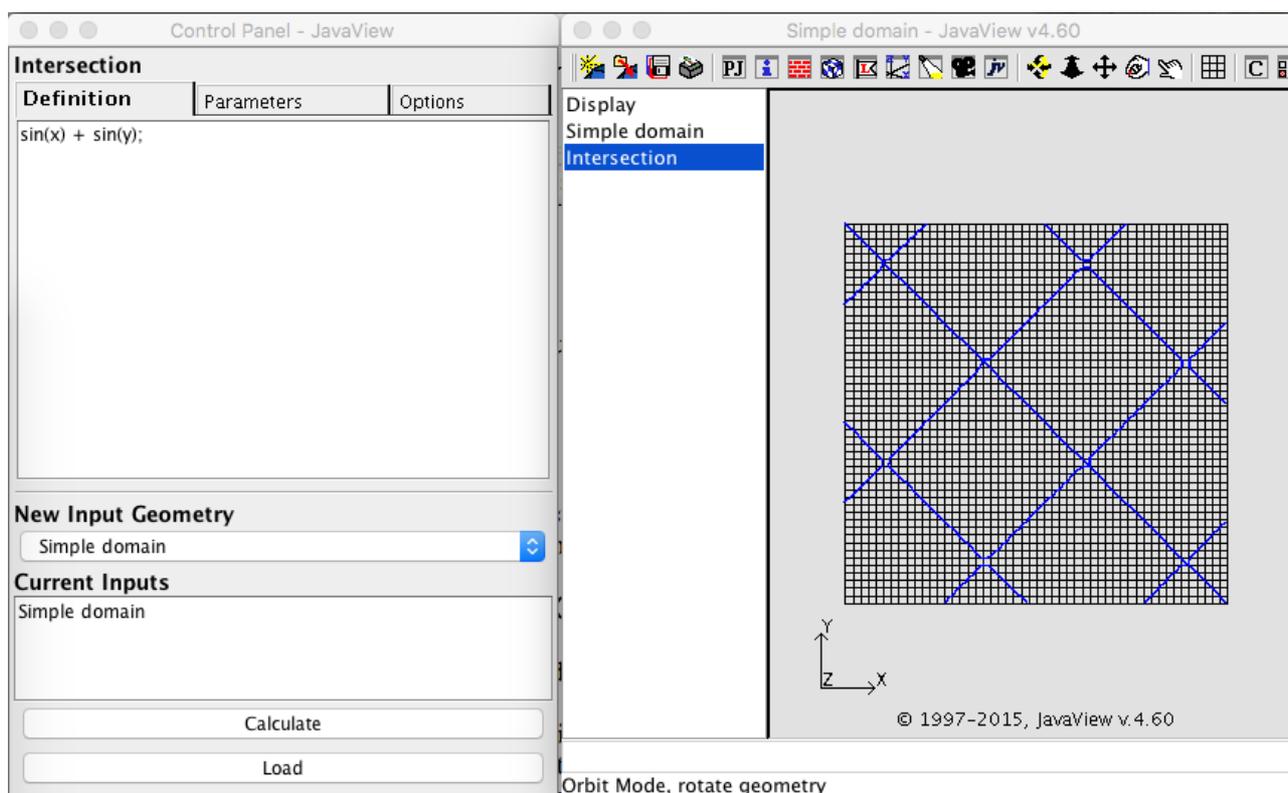
## Intersection

This module calculates the intersection of a give curve or surface with a given function. The defining equation need not be polynomial, for example

$$\sin(x) + \sin(y);$$

specifies a periodic diagonal grid.

To work an existing geometry must be specified. Often you will want a simple rectangular domain. calculated using the Parametrise Surface module. The **New Input Geometry** drop down list is used the select the input.



The picture shows the result of applying the intersection to a simple domain.

When there are multiple modules running you can select which one is displayed in the control panel by clicking the module name in sidebar. Hence to change the size or resolution of the grid click on "simple domain".

## Mapping

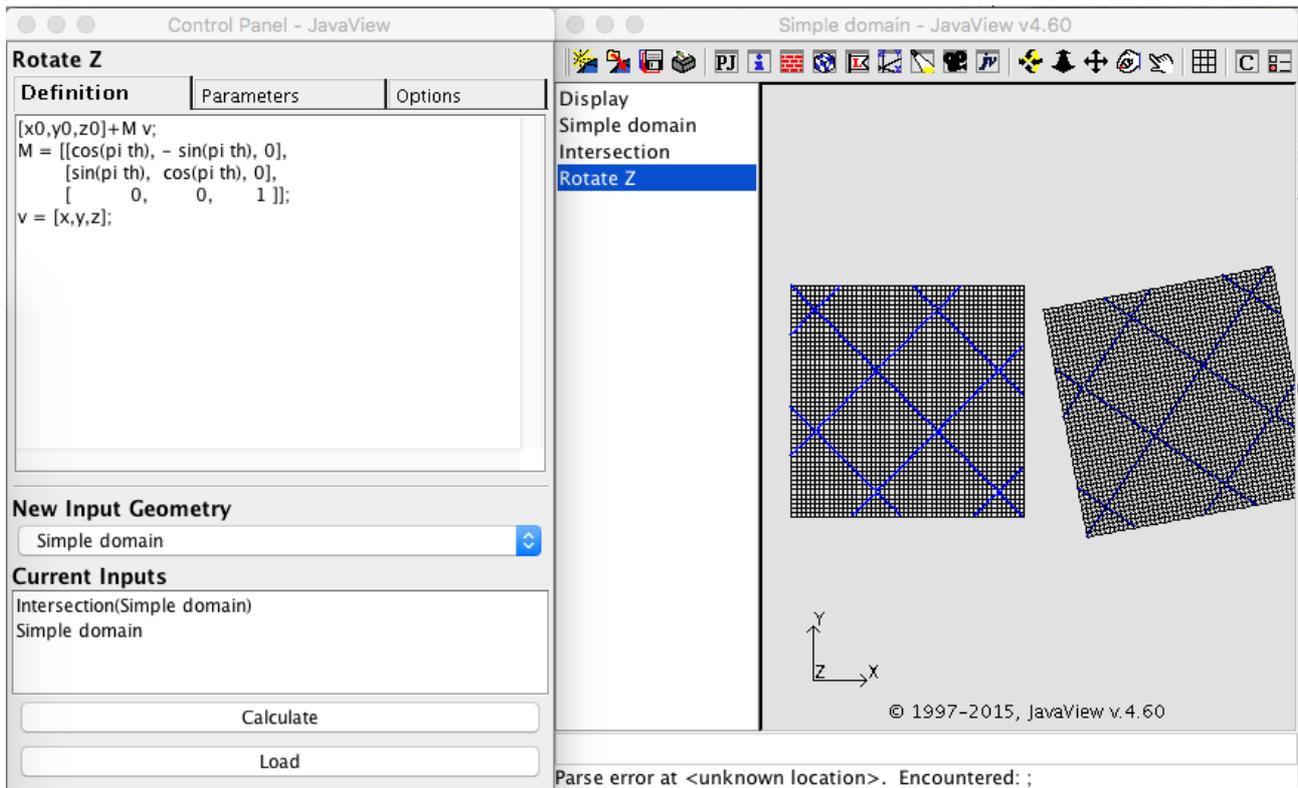
The mapping module applies a mapping from  $\mathbf{R}^3$  to  $\mathbf{R}^3$ . This is specified by a 3D vector expression in three variable.

$$[\cos(\pi \text{ th}) x - \sin(\pi \text{ th}) y, \sin(\pi \text{ th}) x + \cos(\pi \text{ th}) y, z];$$

or alternatively as a matrix equation

$$\begin{aligned} M v; \\ M &= \begin{bmatrix} \cos(\pi \text{ th}) & -\sin(\pi \text{ th}) & 0 \\ \sin(\pi \text{ th}) & \cos(\pi \text{ th}) & 0 \\ 0 & 0 & 1 \end{bmatrix}; \\ v &= [x, y, z]; \end{aligned}$$

The mapping program works in the same fashion as the intersection, with objects to be operated on chosen by selecting them from the **New Input Geometry** drop down menu. It is possible for the same mapping to apply to more than one object at the same time.



Here the rotation has been applied to the grid and also to the intersection with the grid.

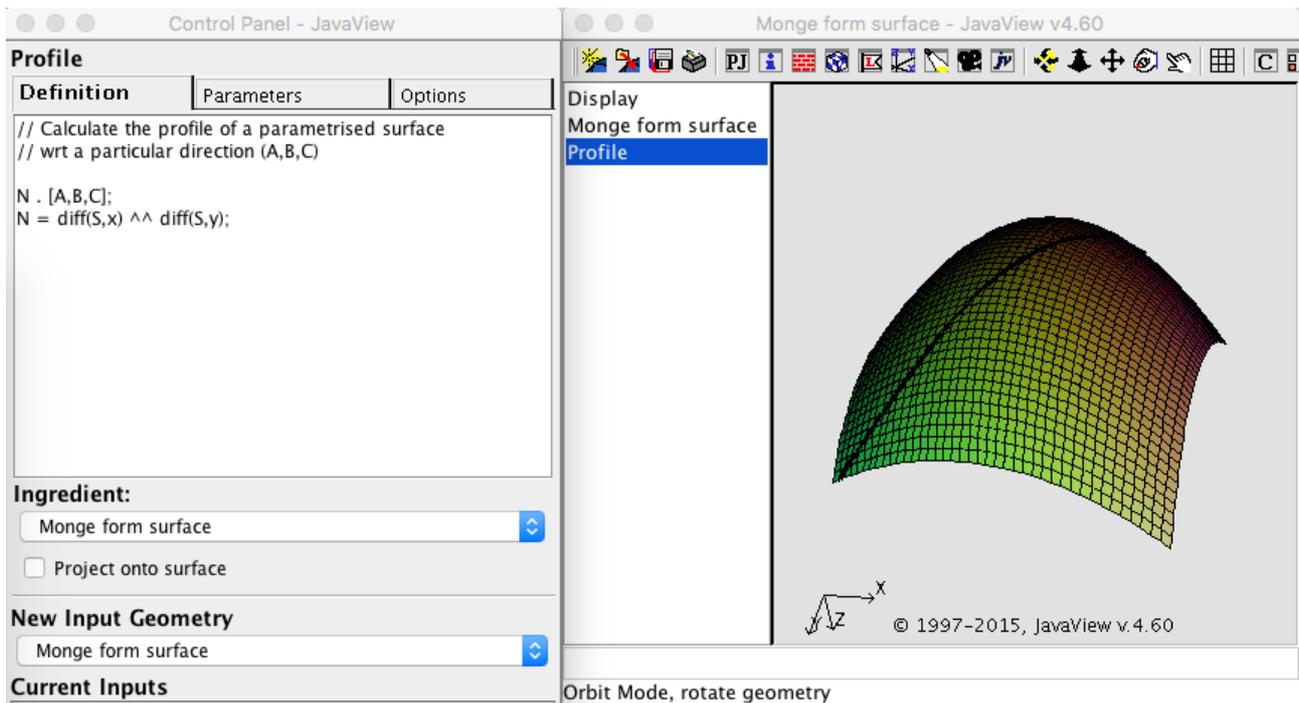
## Generalised Intersection

The generalised intersection module calculates zero sets of functions when the definition depends on the definition of another object. For example the profile of a parametric surface can be calculated with the definition

$$N = [A, B, C];$$

$$N = \text{diff}(S, x) \wedge \wedge \text{diff}(S, y);$$

Here the variable S output of the of the parametrised surface and x, y are the two input parameters.



To calculate the intersection two inputs needs to be specified. First the **Ingredient** must be specified this has the definition of the surface. Then another object with the input geometry, this need not be the same surface. There is an option which will automatically project the result of the intersection onto the surface.

## Generalised Mapping

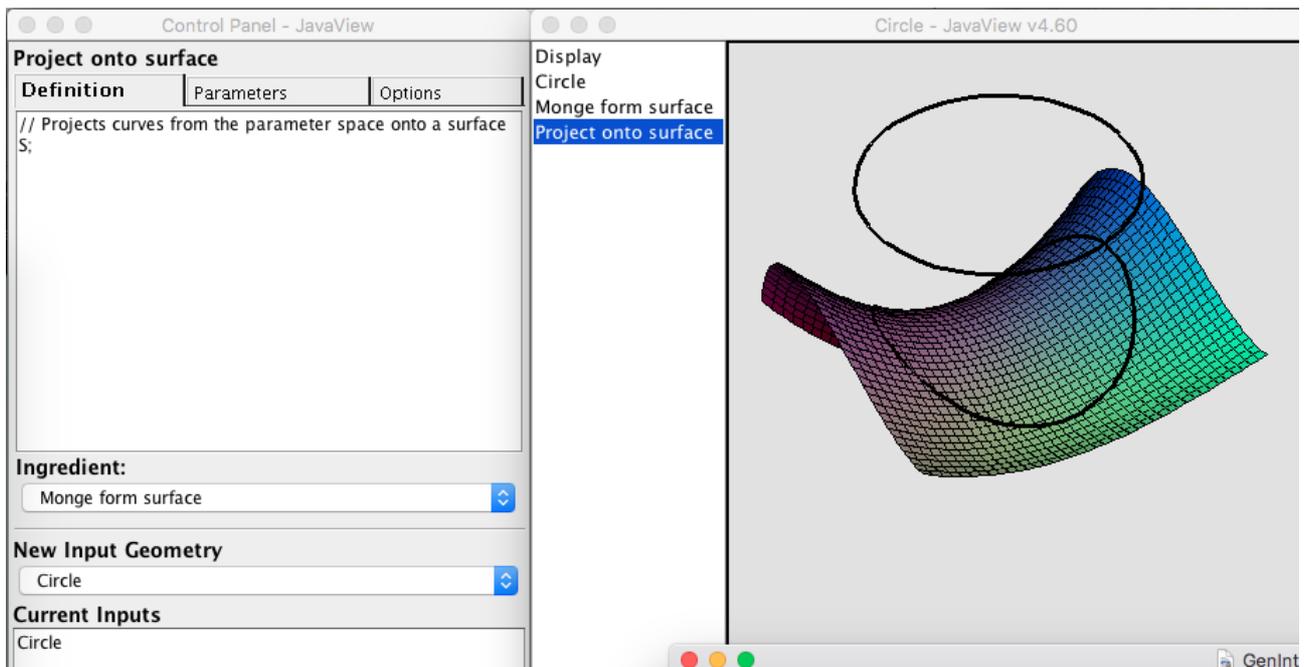
The generalised mapping module performs a mapping when the definition depends on another module. For example to project onto a surface the simple definition is

```
// Projects curves from the parameter space onto a surface
S;
```

a more complex example is the Gauss Map which uses input surface S as well as the two variables x and y.

```
N / sqrt(N.N);
N = Sx ^^ Sy;
Sx = diff(S,x);
Sy = diff(S,y);
```

here x, y are the two variables used in the definition of the surface.



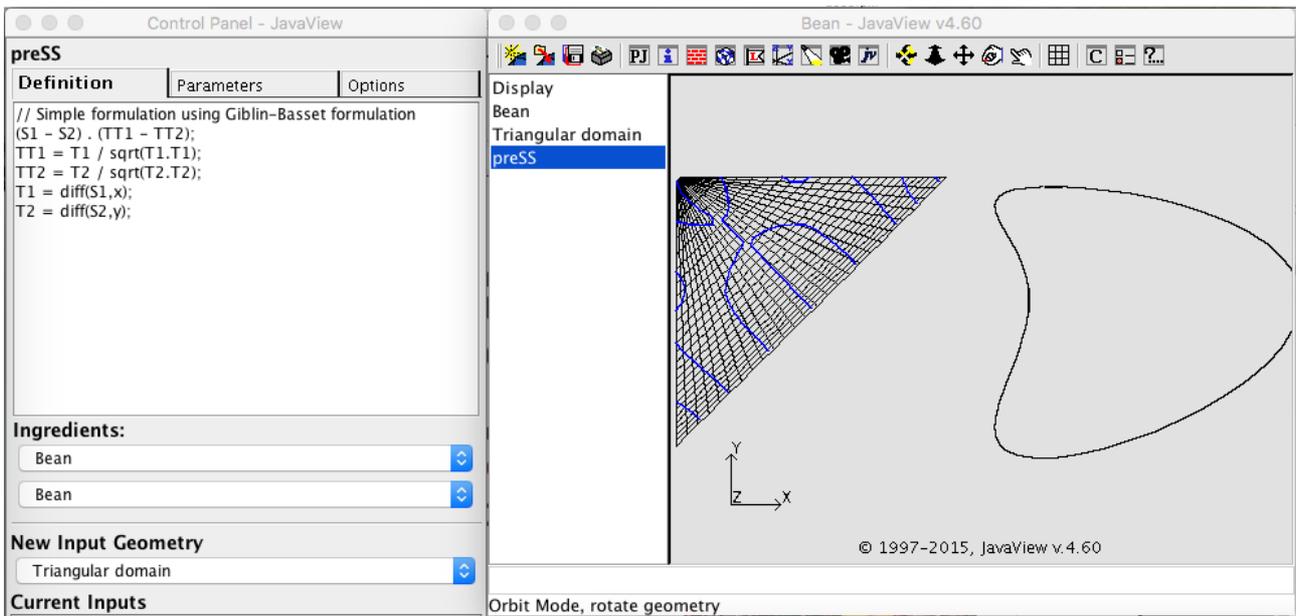
## Symmetry sets and allies

Symmetry sets and related sets depend on two curves - it is possible to calculate the symmetry set between two curve segments giving the points with a bi-tangent circle with one contact point on one segment and the other contact point in the second segment. It is frequently the case when both segments are actually the same curve.

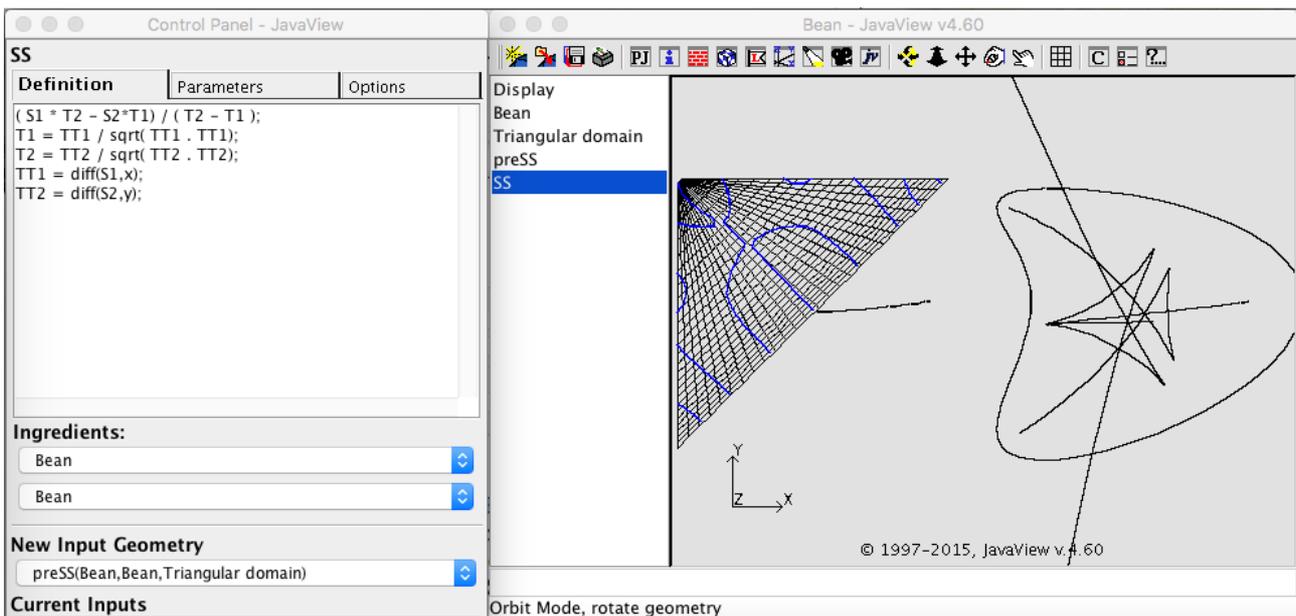
Symmetry sets are typically calculated in two steps first the pre-Symmetry set is calculated in parameter space and then mapped to give the actual Symmetry Set or midpoint locus. These two steps are created using the **Bi-Intersection** and **Bi-Mapping** modules.

Several steps are needed in calculating a symmetry sets

1. Create a parametrised curve. For example the Bean curve in the new menu.
2. Create a domain for pairs of parameters. Normally these will be parametrised surfaces with zero z-coordinates. The "triangular domain" in particular is the set  $\{ (s,t) : t > s \}$  gives unique pairs of points. You may want to turn off the show faces and show edges display properties.
3. Start the **Bi Intersection** module with the "preSS", "preADSS" or "preAESS" definitions.
  1. Set both ingredients to the curve
  2. Set the input Geometry to the domain



4. Start the Bi-Mapping module with the "midpoint", "SS", "ADSS" or "AESS" definition.
  1. Set both ingredients to the curve
  2. Set the Input Geometry to the result of the intersection. This will have a name like "preSS(Bean,Bean,Triangular Domain)".



The resolution of the final symmetry set can be improved by adjusting the domain, which will automatically update subsequent geometries. There are also clipping and continuity controls on the option pane. The continuity control sets the minimum distance between two points needed to draw a line segment between them, setting this can help cases when the SS passes through infinity.

The definitions for bi-intersections and bi-mappings both have two variables S, T for the two input curves. They also have two variables s, t, (or x, y) for the parameters of these two curves.

## JavaView and Display panels

JavaView the framework is written in has many options. In 2D mode main display windows allow translating the scene via click and drag with the mouse. To scale the scene click and hold **s** then click and drag with the mouse. In 3D mode click and drag rotates the scene. If the main display window is resized then a new menubar appears with shortcut buttons.

There are various other control panels which can be visualised. The display control panel has the main display options which can improve rendering of curves and surfaces. There are separate panels to control camera, lighting, axes. Each geometric object also has an Info and material panels. To select these other panels first click on Control panel. This changes the menus to the standard JavaView ones. A particular panel can be selected from the **Inspector** menu.

JavaView has many other options built in. Individual vertices can be selected and deleted, polygonal meshes can be cleaned up. All these options and many more can be selected the **Method** menu. The geometries can loaded and saved using the **File** menu. This only saves the geometry, not the defining equations.

More details on the JavaView options can be obtained from the JavaView website <http://www.javaview.de/development/index.html>.

## Loading and Saving definitions

Each module has options to load and save the defining equations. These are saved in an XML like format. A typical definition might be

```
<definition name="Circle" type="pcurve">
[r cos(pi t)+x0, r sin(pi t)+y0, 0];
<parameter name="r" value="1">
<parameter name="x0" value="0">
<parameter name="y0" value="0">
<variable name="t" min="-1" max="1" steps="60">
</definition>
```

The main `<definition>` tag has a `name` and a `type` attribute and an optional `optype` attribute. The type can be one of `acurve`, `asurf`, `pcurve`, `psurf`, `intersect`, or `mapping`. For generalised intersections and mappings and bi-intersections and mappings the `optype` option specifies the type of input for the ingredient, typically either `pcurve` or `psurf`.

The main equations follows the opening tag.

`<parameter>` tags define the parameters used in a definition these have a name and the default value.

`<variable>` variable tags have a name min and max values as well as the number of steps.

For generalised intersections and mapping and bi-intersections and bi-mappings there are two types of variable

```

<definition name="SS" type="mapping" opType="pcurve">
( S * T2 - T *T1) / ( T2 - T1 );
T1 = TT1 / sqrt( TT1 . TT1);
T2 = TT2 / sqrt( TT2 . TT2);
TT1 = diff(S,s);
TT2 = diff(T,t);

<variable name="S" type="pcurve">
<variable name="T" type="pcurve">
<variable name="s" type="none">
<variable name="t" type="none">
</definition>

```

A variable will either have a type of `pcurve/psurf` or `none`. If not none it should match the `opType` attribute of the definition.

A standard set of definitions can be loaded via the new menu. The actual definitions for these can be found in the `defs` directory of the distribution in the files `psurf.defs` etc.

## Bugs

The software is very much in development stage, there are many known and unknown bug. I will attempt to fix as many as I can. The standard output from the program currently writes a lot of debugging information. I hope to clean this up in the future.

Email [rich@singsurf.org](mailto:rich@singsurf.org) for any bug report/feature request.