Real projective plane $\mathbb{RP}^2$

This Jupyter notebook demonstrates some capabilities of SageMath about differentiable manifolds on the example of real projective plane. The corresponding tools have been developed within the SageManifolds (https://sagemanifolds.obspm.fr) project.

A version of SageMath at least equal to 7.5 is required to run this notebook:

In [1]:
```
version()
```

Out[1]: 'SageMath version 9.2, Release Date: 2020-10-24'

First we set up the notebook to use LaTeX for rendering outputs:

In [2]:
```
%display latex
```

Constructing the manifold

We start by declaring the real projective plane as a 2-dimensional differentiable manifold:

In [3]:
```
RP2 = Manifold(2, 'RP^2', r'\mathbb{RP}^2')
RP2
```

Out[3]: $\mathbb{RP}^2$

Then we provide $\mathbb{RP}^2$ with some atlas. A minimal atlas on $\mathbb{RP}^2$ must have at least three charts. Such an atlas is easy to infer from the common interpretation of $\mathbb{RP}^2$ as the set of lines of $\mathbb{R}^3$ passing through the origin $(x, y, z) = (0, 0, 0)$. Let U_1 be the subset of lines that are not contained in the plane $z = 0$; this is an open set of $\mathbb{RP}^2$, so that we declare it as:

In [4]:
```
U1 = RP2.open_subset('U_1')
U1
```

Out[4]: U_1

Any line in U_1 is uniquely determined by its intersection with the plane $z = 1$. The Cartesian coordinates $(x, y, 1)$ of the intersection point lead to an obvious coordinate system (x_1, y_1) on U_1 by setting $(x_1, y_1) = (x, y)$:

In [5]:
```
X1.<x1,y1> = U1.chart()
X1
```

Out[5]: $(U_1, (x_1, y_1))$

Note that since we have not specified any coordinate range in the arguments of `chart()`, the range of (x_1, y_1) is $\mathbb{R}^2$.

Similarly, let U_2 be the set of lines through the origin of $\mathbb{R}^3$ that are not contained in the plane $x = 0$. Any line in U_2 is uniquely determined by its intersection $(1, y, z)$ with the plane $x = 1$, leading to coordinates $(x_2, y_2) = (y, z)$ on U_2:

In [6]:
```
U2 = RP2.open_subset('U_2')
X2.<x2,y2> = U2.chart()
X2
```

Out[6]: $(U_2, (x_2, y_2))$

Finally, let U_3 be the set of lines through the origin of $\mathbb{R}^3$ that are not contained in the plane $y = 0$. Any line in U_3 is uniquely determined by its intersection $(x, 1, z)$ with the plane $y = 1$, leading to coordinates $(x_3, y_3) = (z, x)$ on U_3:

In [7]:
```
U3 = RP2.open_subset('U_3')
X3.<x3,y3> = U3.chart()
X3
```

Out[7]: $(U_3, (x_3, y_3))$

We declare that the union of the three (overlapping) open domains U_1, U_2 and U_3 is $\mathbb{RP}^2$:

In [8]:
```
RP2.declare_union(U1.union(U2), U3)
U1.union(U2).union(U3)
```

Out[8]: $\mathbb{RP}^2$

At this stage, three open covers of $\mathbb{RP}^2$ have been constructed:

In [9]:
```
RP2.open_covers()
```

Out[9]: $\left[\left[\mathbb{RP}^2\right], [U_1 \cup U_2, U_3], [U_1, U_2, U_3]\right]$

Finally, to fully specify the manifold $\mathbb{RP}^2$, we give the transition maps between the various charts; the transition map between the charts X1=$(U_1, (x_1, y_1))$ and X2=$(U_2, (x_2, y_2))$ is defined on the set $U_{12} := U_1 \cap U_2$ of lines through the origin of $\mathbb{R}^3$ that are neither contained in the plane $x = 0$ ($x_1 = 0$ in U_1) nor contained in the plane $z = 0$ ($y_2 = 0$ in U_2):

In [10]:
```
X1_to_X2 = X1.transition_map(X2, (y1/x1, 1/x1), intersection_name='U_{12}',
                             restrictions1= x1!=0, restrictions2= y2!=0)
X1_to_X2.display()
```

Out[10]:
$$\left\{ \begin{array}{lcl} x_2 & = & \frac{y_1}{x_1} \\ y_2 & = & \frac{1}{x_1} \end{array} \right.$$

The inverse of this transition map is easily computed by Sage:

In [11]:
```
X2_to_X1 = X1_to_X2.inverse()
X2_to_X1.display()
```

Out[11]:
$$\left\{ \begin{array}{lcl} x_1 & = & \frac{1}{y_2} \\ y_1 & = & \frac{x_2}{y_2} \end{array} \right.$$

The transition map between the charts X1=$(U_1, (x_1, y_1))$ and X3=$(U_3, (x_3, y_3))$ is defined on the set $U_{13} := U_1 \cap U_3$ of lines through the origin of $\mathbb{R}^3$ that are neither contained in the plane $y = 0$ ($y_1 = 0$ in U_1) nor contained in the plane $z = 0$ ($x_3 = 0$ in U_3):

In [12]:
```
X1_to_X3 = X1.transition_map(X3, (1/y1, x1/y1), intersection_name='U_{13}',
                             restrictions1= y1!=0, restrictions2= x3!=0)
X1_to_X3.display()
```

Out[12]:
$$\left\{ \begin{array}{lcl} x_3 & = & \frac{1}{y_1} \\ y_3 & = & \frac{x_1}{y_1} \end{array} \right.$$

In [13]:
```
X3_to_X1 = X1_to_X3.inverse()
X3_to_X1.display()
```

Out[13]:
$$\left\{ \begin{array}{lcl} x_1 & = & \frac{y_3}{x_3} \\ y_1 & = & \frac{1}{x_3} \end{array} \right.$$

Finally, the transition map between the charts X2=$(U_2, (x_2, y_2))$ and X3=$(U_3, (x_3, y_3))$ is defined on the set $U_{23} := U_2 \cap U_3$ of lines through the origin of $\mathbb{R}^3$ that are neither contained in the plane $y = 0$ ($x_2 = 0$ in U_2) nor contained in the plane $x = 0$ ($y_3 = 0$ in U_3):

In [14]:
```
X2_to_X3 = X2.transition_map(X3, (y2/x2, 1/x2), intersection_name='U_{23}',
                             restrictions1= x2!=0, restrictions2= y3!=0)
X2_to_X3.display()
```

Out[14]:
$$\begin{cases} x_3 & = & \frac{y_2}{x_2} \\ y_3 & = & \frac{1}{x_2} \end{cases}$$

In [15]:
```
X3_to_X2 = X2_to_X3.inverse()
X3_to_X2.display()
```

Out[15]:
$$\begin{cases} x_2 & = & \frac{1}{y_3} \\ y_2 & = & \frac{x_3}{y_3} \end{cases}$$

At this stage, the manifold $\mathbb{RP}^2$ is fully constructed. It has been provided with the following atlas:

In [16]:
```
RP2.atlas()
```

Out[16]: $[(U_1, (x_1, y_1)), (U_2, (x_2, y_2)), (U_3, (x_3, y_3)), (U_{12}, (x_1, y_1)), (U_{12}, (x_2, y_2)), (U_{13}, (x_1, y_1)), (U_{13}, (x_3, y_3))$

Note that, in addition to the three chart we have defined, the atlas comprises subcharts on the intersection domains U_{12}, U_{13} and U_{23}. These charts can be obtained by the method `restrict()`:

In [17]:
```
U12 = U1.intersection(U2)
U13 = U1.intersection(U3)
U23 = U2.intersection(U3)
X1.restrict(U12)
```

Out[17]: $(U_{12}, (x_1, y_1))$

In [18]:
```
X1.restrict(U12) is RP2.atlas()[3]
```

Out[18]: True

Non-orientability of $\mathbb{RP}^2$

It is well known that $\mathbb{RP}^2$ is not an orientable manifold. To illustrate this, let us make an attempt to construct a global non-vanishing 2-form ϵ on $\mathbb{RP}^2$. If we succeed, this would provide a volume form and $\mathbb{RP}^2$ would be orientable. We start by declaring ϵ as a 2-form on $\mathbb{RP}^2$:

In [19]:
```
eps = RP2.diff_form(2, name='eps', latex_name=r'\epsilon')
print(eps)
```

```
2-form eps on the 2-dimensional differentiable manifold RP^2
```

We set the value of ϵ on domain U_1 to be $\mathrm{d}x_1 \wedge \mathrm{d}y_1$ by demanding that the component ϵ_{01} of ϵ with respect to coordinates (x_1, y_1) is one, as follows:

In [20]:
```
e1 = X1.frame()
e1
```

Out[20]:
$$\left(U_1, \left(\frac{\partial}{\partial x_1}, \frac{\partial}{\partial y_1}\right)\right)$$

In [21]:
```
eps[e1,0,1] = 1
eps.display(e1)
```

Out[21]: $\epsilon = \mathrm{d}x_1 \wedge \mathrm{d}y_1$

If we ask for the expression of ϵ in terms of the coframe $(\mathrm{d}x_2, \mathrm{d}y_2)$ associated with the chart X2 on $U_{12} = U_1 \cap U_2$, we get

In [22]:
```
eps.display(X2.frame().restrict(U12), chart=X2.restrict(U12))
```

Out[22]:
$$\epsilon = \frac{1}{y_2^3}\mathrm{d}x_2 \wedge \mathrm{d}y_2$$

Now, the complement of U_{12} in U_2 is defined by $y_2 = 0$. The above expression shows that it is not possible to extend smoothly ϵ to the whole domain U_2. We conclude that starting from $\mathrm{d}x_1 \wedge \mathrm{d}y_1$ on U_1, it is not possible to get a regular non-vanishing 2-form on $\mathbb{RP}^2$. This of course follows from the fact that $\mathbb{RP}^2$ is not orientable.

Steiner map (Roman surface)

Let us first define $\mathbb{R}^3$ as a 3-dimensional manifold, with a single-chart atlas (Cartesian coordinates Y):

In [23]:
```
R3 = Manifold(3, 'R^3', r'\mathbb{R}^3')
Y.<x,y,z> = R3.chart()
```

The Steiner map is a map $\mathbb{RP}^2 \to \mathbb{R}^3$ defined as follows:

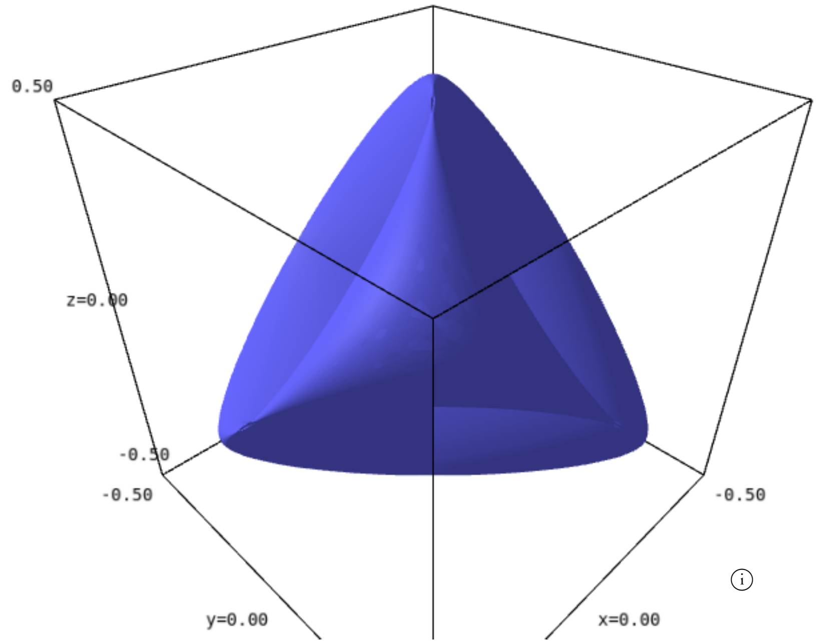
In [24]:
```
Phi = RP2.diff_map(R3, {(X1,Y): [y1/(1+x1^2+y1^2), x1/(1+x1^2+y1^2), x1*y1/(1+x1^2+y1
^2)],
                        (X2,Y): [x2*y2/(1+x2^2+y2^2), y2/(1+x2^2+y2^2), x2/(1+x2^2+y2
^2)],
                        (X3,Y): [x3/(1+x3^2+y3^2), x3*y3/(1+x3^2+y3^2), y3/(1+x3^2+y3
^2)]},
                   name='Phi', latex_name=r'\Phi')
Phi.display()
```

Out[24]:
$$\begin{array}{llcl} \Phi: & \mathbb{RP}^2 & \longrightarrow & \mathbb{R}^3 \\ \text{on } U_1: & (x_1, y_1) & \longmapsto & (x, y, z) = \left(\frac{y_1}{x_1^2+y_1^2+1}, \frac{x_1}{x_1^2+y_1^2+1}, \frac{x_1 y_1}{x_1^2+y_1^2+1}\right) \\ \text{on } U_2: & (x_2, y_2) & \longmapsto & (x, y, z) = \left(\frac{x_2 y_2}{x_2^2+y_2^2+1}, \frac{y_2}{x_2^2+y_2^2+1}, \frac{x_2}{x_2^2+y_2^2+1}\right) \\ \text{on } U_3: & (x_3, y_3) & \longmapsto & (x, y, z) = \left(\frac{x_3}{x_3^2+y_3^2+1}, \frac{x_3 y_3}{x_3^2+y_3^2+1}, \frac{y_3}{x_3^2+y_3^2+1}\right) \end{array}$$

Φ is a topological immersion of $\mathbb{RP}^2$ into $\mathbb{R}^3$, but it is not a smooth immersion (contrary to the Apéry map below): its differential is not injective at $(x_1, y_1) = (0, 1)$ and $(x_1, y_1) = (1, 0)$. The image of Φ is a self-intersecting surface of $\mathbb{R}^3$, called the **Roman surface**:

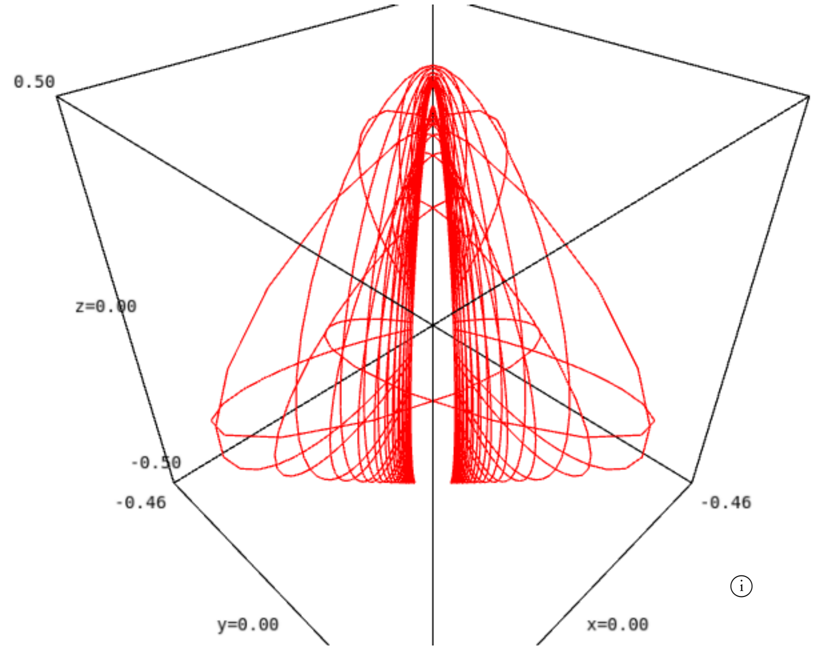
In [25]:
```
g1 = parametric_plot3d(Phi.expr(X1,Y), (x1,-10,10), (y1,-10,10), plot_points=[100,10
0])
g2 = parametric_plot3d(Phi.expr(X2,Y), (x2,-10,10), (y2,-10,10), plot_points=[100,10
0])
g3 = parametric_plot3d(Phi.expr(X3,Y), (x3,-10,10), (y3,-10,10), plot_points=[100,10
0])
g1+g2+g3
```

Out[25]:



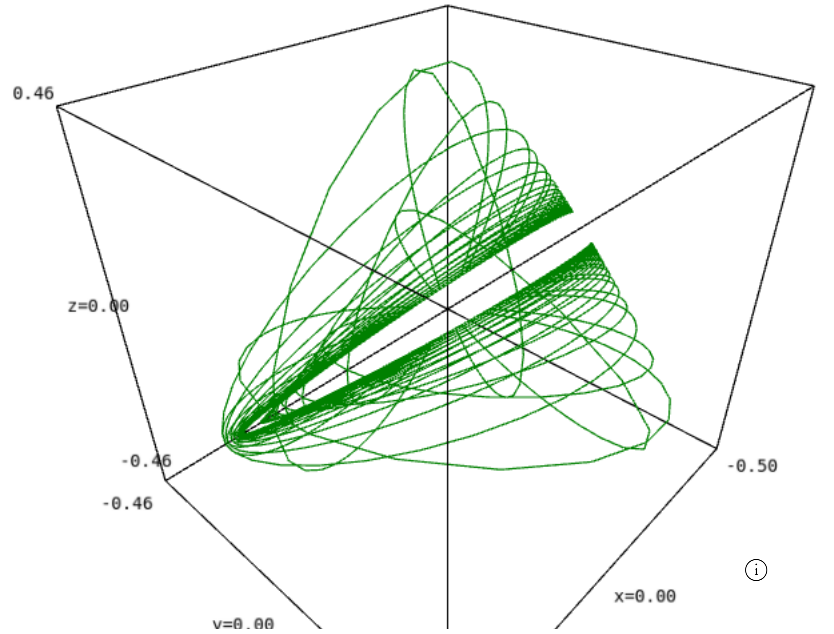
In [26]:
```
gX1 = X1.plot(Y, mapping=Phi, max_range=16, number_values=24, plot_points=100,
              label_axes=False)
gX2 = X2.plot(Y, mapping=Phi, max_range=16, number_values=24, plot_points=100,
              label_axes=False, color='green')
gX3 = X3.plot(Y, mapping=Phi, max_range=16, number_values=24, plot_points=100,
              label_axes=False, color='blue')
gX1
```

Out[26]:



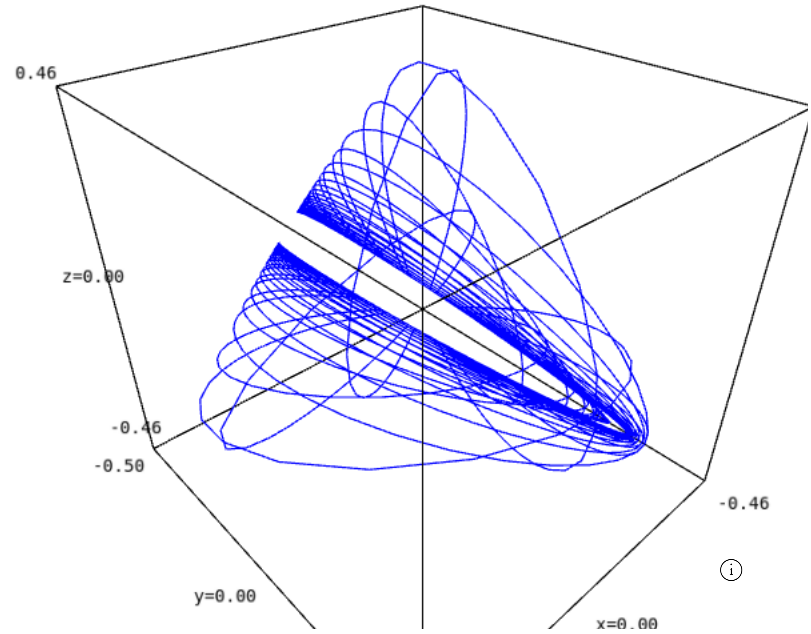
```
In [27]: gX2
```

Out[27]:



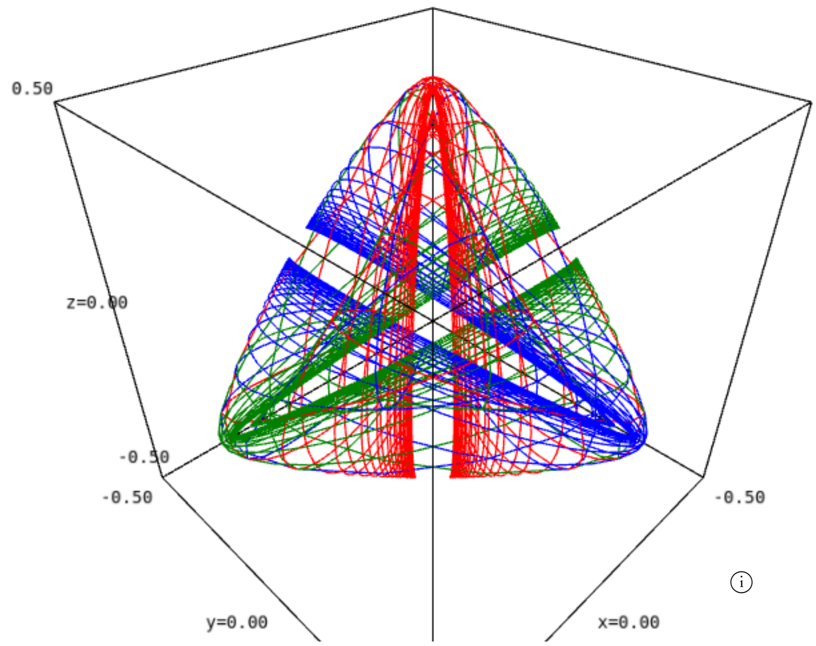

```
In [28]: gX3
```

Out[28]:



In [29]:
```
gX1+gX2+gX3
```

Out[29]:



Apéry map (Boy surface)

The Apéry map [Apéry, Adv. Math. **61**, 185 (1986) (http://dx.doi.org/10.1016/0001-8708(86)90080-0)] is a smooth immersion $\Psi : \mathbb{RP}^2 \to \mathbb{R}^3$. In terms of the charts X1, X2, X3 introduced above, it is defined as follows:

In [30]:
```
fx = ((2*x^2-y^2-z^2)*(x^2+y^2+z^2)+2*y*z*(y^2-z^2)+z*x*(x^2-z^2)+x*y*(y^2-z^2))/2
fx
```

Out[30]:

$$\frac{1}{2}\left(y^2 - z^2\right)xy + \frac{1}{2}\left(x^2 - z^2\right)xz + \left(y^2 - z^2\right)yz + \frac{1}{2}\left(2\,x^2 - y^2 - z^2\right)\left(x^2 + y^2 + z^2\right)$$

In [31]:
```
fy = sqrt(3)/2*((y^2-z^2)*(x^2+y^2+z^2)+z*x*(z^2-x^2)+x*y*(y^2-x^2))
fy
```

Out[31]:

$$-\frac{1}{2}\,\sqrt{3}\left(\left(x^2 - y^2\right)xy + \left(x^2 - z^2\right)xz - \left(x^2 + y^2 + z^2\right)\left(y^2 - z^2\right)\right)$$

In [32]:
```
fz = (x+y+z)*((x+y+z)^3/4+(y-x)*(z-y)*(x-z))
fz
```

Out[32]:

$$\frac{1}{4}\left((x + y + z)^3 + 4\,(x - y)(x - z)(y - z)\right)(x + y + z)$$

In [33]:
```
a = sqrt(1+x1^2+y1^2)
fx1 = fx.subs(x=x1/a, y=y1/a, z=1/a).simplify_full()
fy1 = fy.subs(x=x1/a, y=y1/a, z=1/a).simplify_full()
fz1 = fz.subs(x=x1/a, y=y1/a, z=1/a).simplify_full()
```

In [34]:
```
a = sqrt(1+x2^2+y2^2)
fx2 = fx.subs(x=1/a, y=x2/a, z=y2/a).simplify_full()
fy2 = fy.subs(x=1/a, y=x2/a, z=y2/a).simplify_full()
fz2 = fz.subs(x=1/a, y=x2/a, z=y2/a).simplify_full()
```

In [35]:
```
a = sqrt(1+x3^2+y3^2)
fx3 = fx.subs(x=y3/a, y=1/a, z=x3/a).simplify_full()
fy3 = fy.subs(x=y3/a, y=1/a, z=x3/a).simplify_full()
fz3 = fz.subs(x=y3/a, y=1/a, z=x3/a).simplify_full()
```

In [36]:
```
Psi = RP2.diff_map(R3, {(X1,Y): [fx1, fy1, fz1], (X2,Y): [fx2, fy2, fz2],
                        (X3,Y): [fx3, fy3, fz3]}, name='Psi', latex_name=r'\Psi')
Psi.display()
```

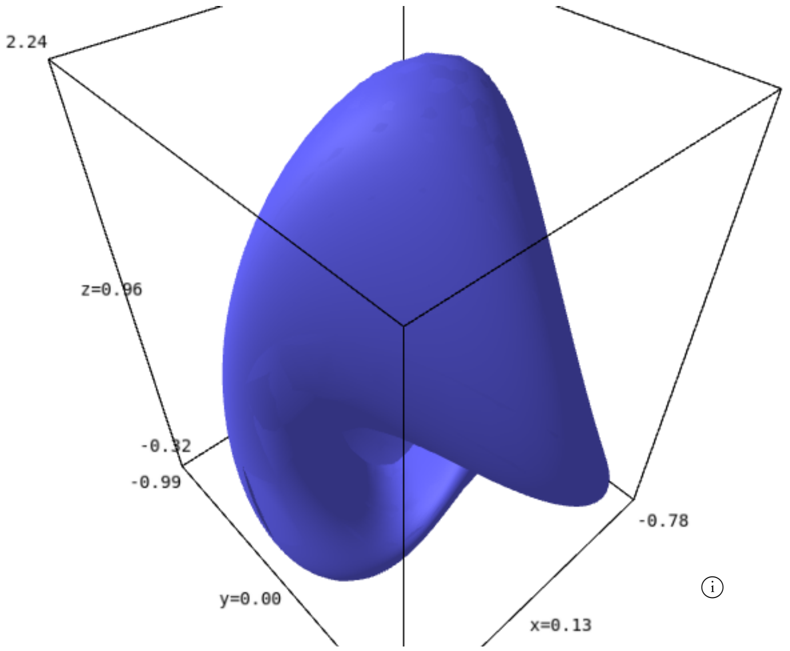
Out[36]:

$$\begin{array}{llll} \Psi: & \mathbb{RP}^2 & \longrightarrow & \mathbb{R}^3 \\ \text{on } U_1: & (x_1, y_1) & \longmapsto & (x, y, z) \\ & & & = \left(\frac{2\,x_1^4+(x_1+2)y_1^3-y_1^4+x_1^3+\left(x_1^2-2\right)y_1^2+x_1^2-(x_1+2)y_1-x_1-1}{2\,\left(x_1^4+y_1^4+2\,\left(x_1^2+1\right)y_1^2+2\,x_1^2+1\right)}, -\frac{\sqrt{3}x_1^3y_1-\sqrt{3}x_1^2y_1^2-\sqrt{3}x_1y_1^3-\sqrt{3}y_1^4+\sqrt{3}x}{2\,\left(x_1^4+y_1^4+2\,\left(x_1^2+1\right)y_1^2+2\right.} \right. \\ \text{on } U_2: & (x_2, y_2) & \longmapsto & (x, y, z) \\ & & & = \left(-\frac{x_2^4+(2\,x_2+1)y_2^3+y_2^4-x_2^3+\left(2\,x_2^2+x_2-1\right)y_2^2-x_2^2-\left(2\,x_2^3+1\right)y_2-2}{2\,\left(x_2^4+y_2^4+2\,\left(x_2^2+1\right)y_2^2+2\,x_2^2+1\right)}, \frac{\sqrt{3}x_2^4-\sqrt{3}y_2^4+\sqrt{3}x_2^3+\sqrt{3}y_2^3+\sqrt{3}x_2^2-}{2\,\left(x_2^4+y_2^4+2\,\left(x_2^2+1\right)y_2^2+\right.} \right. \\ \text{on } U_3: & (x_3, y_3) & \longmapsto & (x, y, z) \\ & & & = \left(-\frac{x_3^4-x_3y_3^3-2\,y_3^4+2\,x_3^3-\left(x_3^2+1\right)y_3^2+2\,x_3^2+\left(x_3^3+x_3^2-1\right)y_3-2\,x_3+1}{2\,\left(x_3^4+y_3^4+2\,\left(x_3^2+1\right)y_3^2+2\,x_3^2+1\right)}, - \right. \\ & & & \frac{x_3^4+y_3^4+6\,\left(x_3^2+2\,x_3+1\right)y_3^2+8\,y_3^3+6\,x_3^2+4\,(}{4\,\left(x_3^4+y_3^4+2\,\left(x_3^2+1\right)y_3^2\right.} \end{array}$$

The image of Ψ is a self-intersecting surface of $\mathbb{R}^3$, called the **Boy surface,** after Werner Boy (1879-1914):

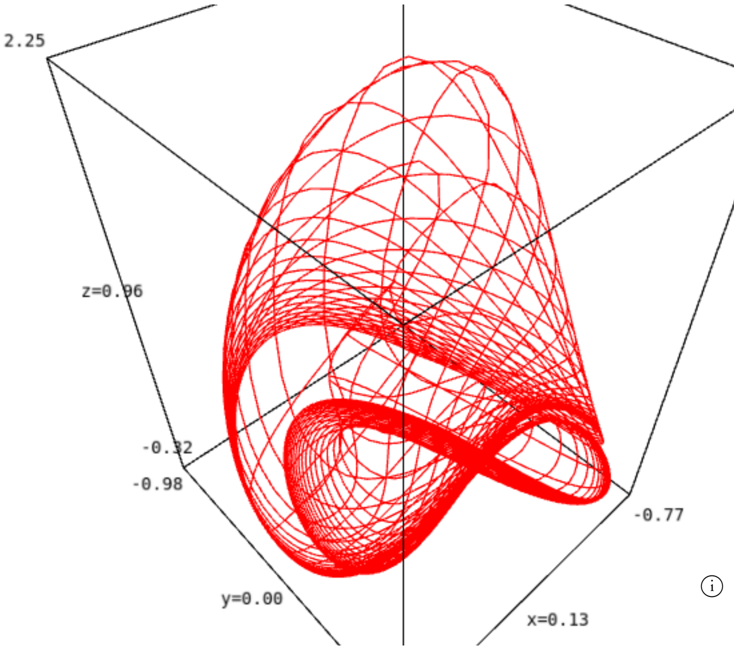
In [37]:
```
g1 = parametric_plot3d(Psi.expr(X1,Y), (x1,-10,10), (y1,-10,10), plot_points=[100,10
0])
g2 = parametric_plot3d(Psi.expr(X2,Y), (x2,-10,10), (y2,-10,10), plot_points=[100,10
0])
g3 = parametric_plot3d(Psi.expr(X3,Y), (x3,-10,10), (y3,-10,10), plot_points=[100,10
0])
g1+g2+g3
```

Out[37]:

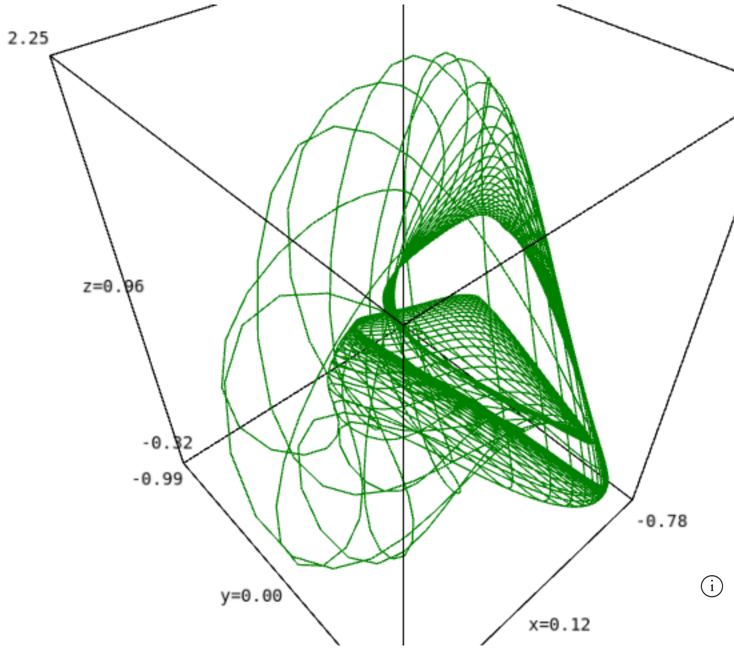


In [38]:
```
gX1 = X1.plot(Y, mapping=Psi, number_values=40, plot_points=100, label_axes=False)
gX2 = X2.plot(Y, mapping=Psi, number_values=40, plot_points=100, label_axes=False,
              color='green')
gX3 = X3.plot(Y, mapping=Psi, number_values=40, plot_points=100, label_axes=False,
              color='blue')
```

In [39]:
```
gX1
```
Out[39]:

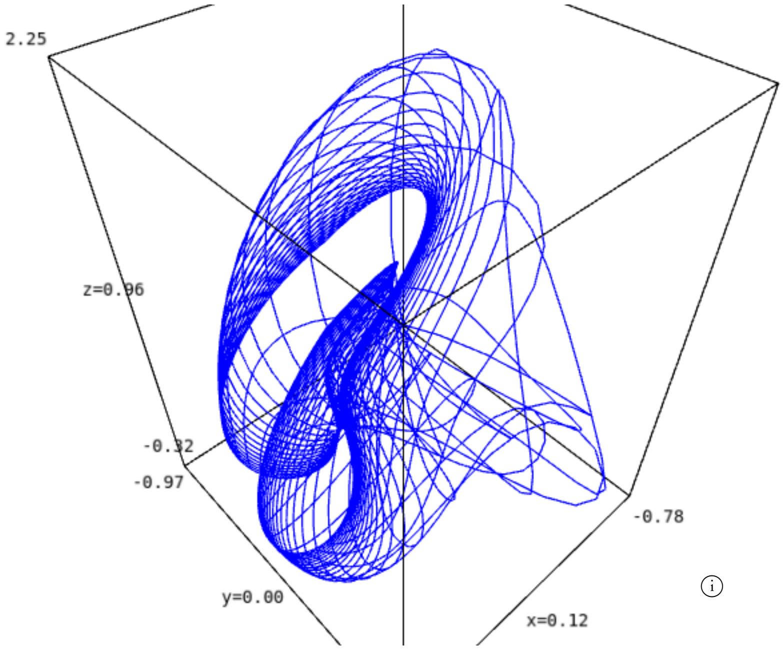


In [40]:
```
gX2
```
Out[40]:



In [41]: gX3

Out[41]:



In [42]: gX1+gX2+gX3

Out[42]: