# Sphere $\mathbb{S}^2$

This notebook demonstrates some differential geometry capabilities of SageMath on the example of the 2-dimensional sphere. The corresponding tools have been developed within the SageManifolds (http://sagemanifolds.obspm.fr) project.

Click here (https://raw.githubusercontent.com/sagemanifolds/SageManifolds/master/Notebooks/SM_sphere_S2.ipynb) to download the notebook file (ipynb format). To run it, you must start SageMath with the Jupyter interface, via the command `sage -n jupyter`

*NB:* a version of SageMath at least equal to 9.2 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 display math formulas using LaTeX formatting:

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

and we initialize a time counter for benchmarking:

In [3]:
```
import time
comput_time0 = time.perf_counter()
```

## $\mathbb{S}^2$ as a 2-dimensional differentiable manifold

We start by declaring $\mathbb{S}^2$ as a differentiable manifold of dimension 2 over $\mathbb{R}$:

In [4]:
```
S2 = Manifold(2, 'S^2', latex_name=r'\mathbb{S}^2', start_index=1)
```

The first argument, `2`, is the dimension of the manifold, while the second argument is the symbol used to label the manifold.

The argument `start_index` sets the index range to be used on the manifold for labelling components w.r.t. a basis or a frame: `start_index=1` corresponds to $\{1, 2\}$; the default value is `start_index=0` and yields $\{0, 1\}$.

In [5]:
```
print(S2)
```
```
2-dimensional differentiable manifold S^2
```

In [6]:
```
S2
```
Out[6]: $\mathbb{S}^2$

The manifold is a `Parent` object:

In [7]:
```
isinstance(S2, Parent)
```
Out[7]: True

in the category of smooth manifolds over $\mathbb{R}$:

In [8]:
```
S2.category()
```
Out[8]: $\mathbf{Smooth}_{\mathbf{R}}$

## Coordinate charts on $\mathbb{S}^2$

The sphere cannot be covered by a single chart. At least two charts are necessary, for instance the charts associated with the stereographic projections from the North pole and the South pole respectively. Let us introduce the open subsets covered by these two charts:

$$U := \mathbb{S}^2 \setminus \{N\},$$

$$V := \mathbb{S}^2 \setminus \{S\},$$

where $N$ is a point of $\mathbb{S}^2$, which we shall call the *North pole*, and $S$ is the point of $U$ of stereographic coordinates $(0, 0)$, which we call the *South pole*:

In [9]:
```
U = S2.open_subset('U') ; print(U)
```
```
Open subset U of the 2-dimensional differentiable manifold S^2
```

In [10]:
```
V = S2.open_subset('V') ; print(V)
```
```
Open subset V of the 2-dimensional differentiable manifold S^2
```

We declare that $\mathbb{S}^2 = U \cup V$:

In [11]:
```
S2.declare_union(U, V)
```

Then we declare the stereographic chart on $U$, denoting by $(x, y)$ the coordinates resulting from the stereographic projection from the North pole:

In [12]:
```
stereoN.<x,y> = U.chart()
```

The expression `.<x,y>` in the left-hand side means that the Python variables `x` and `y` are set to the two coordinates of the chart. This allows one to refer subsequently to the coordinates by their names. In the present case, the function `chart()` has no argument, which implies that the coordinate symbols will be `x` and `y` (i.e. exactly the characters appearing in the `<...>` operator) and that each coordinate range is $(-\infty, +\infty)$. As we will see below, for other cases, an argument must be passed to `chart()` to specify each coordinate symbol and range, as well as some specific LaTeX symbol.

In [13]:
```
stereoN
```
Out[13]: $(U, (x, y))$

The coordinates can be accessed individually, either by means of their indices in the chart ( following the convention `start_index=1` set in the manifold's definition) or by their names as Python variables:

In [14]:
```
stereoN[1]
```
Out[14]: $x$

In [15]:
```
y is stereoN[2]
```
Out[15]: True

Similarly, we introduce on $V$ the coordinates $(x', y')$ corresponding to the stereographic projection from the South pole:

In [16]:
```
stereoS.<xp,yp> = V.chart("xp:x' yp:y'")
```

In this case, the string argument passed to `chart` stipulates that the text-only names of the coordinates are xp and yp (same as the Python variables names defined within the `<...>` operator in the left-hand side), while their LaTeX names are $x'$ and $y'$.

In [17]:
```
stereoS
```

Out[17]: $\left(V, (x', y')\right)$

At this stage, the user's atlas on the manifold has two charts:

In [18]:
```
S2.atlas()
```

Out[18]: $\left[(U, (x, y)), \left(V, (x', y')\right)\right]$

We have to specify the **transition map** between the charts 'stereoN' = $(U, (x, y))$ and 'stereoS' = $(V, (x', y'))$; it is given by the standard inversion formulas:

In [19]:
```
stereoN_to_S = stereoN.transition_map(stereoS,
                                      (x/(x^2+y^2), y/(x^2+y^2)),
                                      intersection_name='W',
                                      restrictions1= x^2+y^2!=0,
                                      restrictions2= xp^2+yp^2!=0)
stereoN_to_S.display()
```

Out[19]:
$$\begin{cases} x' &= \frac{x}{x^2+y^2} \\ y' &= \frac{y}{x^2+y^2} \end{cases}$$

In the above declaration, 'W' is the name given to the chart-overlap subset: $W := U \cap V$, the condition $x^2 + y^2 \neq 0$ defines $W$ as a subset of $U$, and the condition $x'^2 + y'^2 \neq 0$ defines $W$ as a subset of $V$.

The inverse coordinate transformation is computed by means of the method `inverse()` :

In [20]:
```
stereoS_to_N = stereoN_to_S.inverse()
stereoS_to_N.display()
```

Out[20]:
$$\begin{cases} x &= \frac{x'}{x'^2+y'^2} \\ y &= \frac{y'}{x'^2+y'^2} \end{cases}$$

In the present case, the situation is of course perfectly symmetric regarding the coordinates $(x, y)$ and $(x', y')$.

At this stage, the user's atlas has four charts:

In [21]:
```
S2.atlas()
```

Out[21]: $\left[(U, (x, y)), \left(V, (x', y')\right), (W, (x, y)), \left(W, (x', y')\right)\right]$

Let us store $W = U \cap V$ into a Python variable for future use:

In [22]:
```
W = U.intersection(V)
```

Similarly we store the charts $(W, (x, y))$ (the restriction of $(U, (x, y))$ to $W$) and $(W, (x', y'))$ (the restriction of $(V, (x', y'))$ to $W$) into Python variables:

In [23]:
```
stereoN_W = stereoN.restrict(W)
stereoN_W
```

Out[23]: $(W, (x, y))$

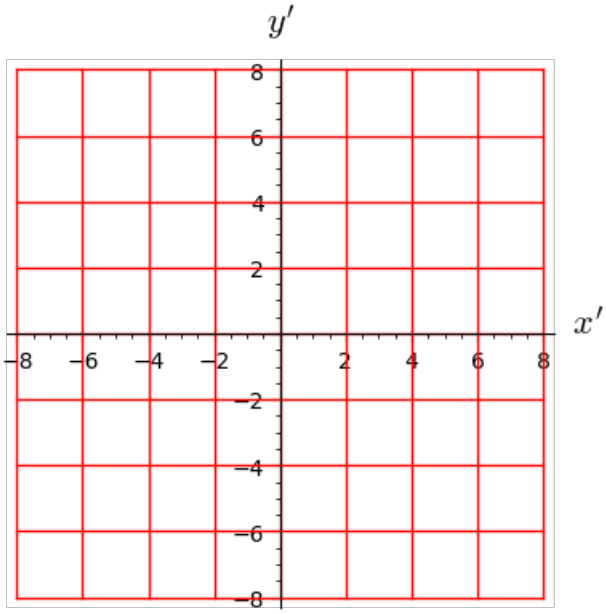
In [24]:
```
stereoS_W = stereoS.restrict(W)
stereoS_W
```

Out[24]: $\left(W, (x', y')\right)$

We may plot the chart $(W, (x', y'))$ in terms of itself, as a grid:

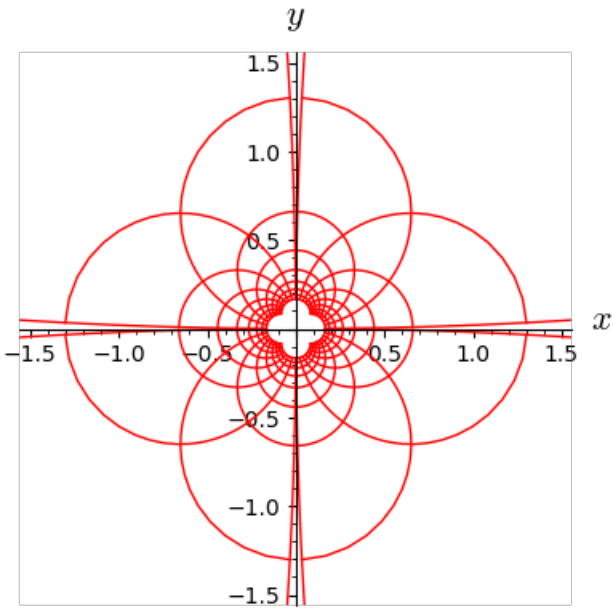
In [25]:
```
stereoS_W.plot()
```

Out[25]:




More interestingly, let us plot the stereographic chart $(x', y')$ in terms of the stereographic chart $(x, y)$ on the domain $W$ where both systems overlap (we split the plot in four parts to avoid the singularity at $(x', y') = (0, 0)$):

In [26]:
```
graphSN1 = stereoS_W.plot(stereoN, ranges={xp:[-6,-0.02], yp:[-6,-0.02]})
graphSN2 = stereoS_W.plot(stereoN, ranges={xp:[-6,-0.02], yp:[0.02,6]})
graphSN3 = stereoS_W.plot(stereoN, ranges={xp:[0.02,6], yp:[-6,-0.02]})
graphSN4 = stereoS_W.plot(stereoN, ranges={xp:[0.02,6], yp:[0.02,6]})
show(graphSN1+graphSN2+graphSN3+graphSN4,
     xmin=-1.5, xmax=1.5, ymin=-1.5, ymax=1.5)
```




## Spherical coordinates

The standard spherical (or polar) coordinates $(\theta, \phi)$ are defined on the open domain $A \subset W \subset \mathbb{S}^2$ that is the complement of the "origin meridian"; since the latter is the half-circle defined by $y = 0$ and $x \geq 0$, we declare:

In [27]:
```
A = W.open_subset('A', coord_def={stereoN_W: (y!=0, x<0),
                                  stereoS_W: (yp!=0, xp<0)})
print(A)
```

```
Open subset A of the 2-dimensional differentiable manifold S^2
```

The restriction of the stereographic chart from the North pole to $A$ is

In [28]:
```
stereoN_A = stereoN_W.restrict(A)
stereoN_A
```

Out[28]: $(A, (x, y))$

We then declare the chart $(A, (\theta, \phi))$ by specifying the intervals $(0, \pi)$ and $(0, 2\pi)$ spanned by respectively $\theta$ and $\phi$:

In [29]:
```
spher.<th,ph> = A.chart(r'th:(0,pi):\theta ph:(0,2*pi):\phi') ; spher
```

Out[29]: $(A, (\theta, \phi))$

The specification of the spherical coordinates is completed by providing the transition map with the stereographic chart $(A, (x, y))$:

In [30]:
```
spher_to_stereoN = spher.transition_map(stereoN_A,
                                        (sin(th)*cos(ph)/(1-cos(th)),
                                         sin(th)*sin(ph)/(1-cos(th))))
spher_to_stereoN.display()
```

Out[30]:
$$\begin{cases} x &= -\frac{\cos(\phi)\,\sin(\theta)}{\cos(\theta)-1} \\ y &= -\frac{\sin(\phi)\,\sin(\theta)}{\cos(\theta)-1} \end{cases}$$

We also provide the inverse transition map:

In [31]:
```
spher_to_stereoN.set_inverse(2*atan(1/sqrt(x^2+y^2)), atan2(-y,-x)+pi)
```

```
Check of the inverse coordinate transformation:
  th == 2*arctan(sqrt(-cos(th) + 1)/sqrt(cos(th) + 1))  **failed**
  ph == pi + arctan2(sin(ph)*sin(th)/(cos(th) - 1), cos(ph)*sin(th)/(cos(th) - 1))  **
failed**
  x == x  *passed*
  y == y  *passed*
NB: a failed report can reflect a mere lack of simplification.
```

The check is passed, modulo some lack of trigonometric simplifications in the first two lines.

In [32]:
```
spher_to_stereoN.inverse().display()
```

Out[32]:
$$\begin{cases} \theta &= 2\,\arctan\left(\frac{1}{\sqrt{x^2+y^2}}\right) \\ \phi &= \pi + \arctan(-y, -x) \end{cases}$$

The transition map $(A, (\theta, \phi)) \to (A, (x', y'))$ is obtained by combining the transition maps $(A, (\theta, \phi)) \to (A, (x, y))$ and $(A, (x, y)) \to (A, (x', y'))$:

In [33]:
```
stereoN_to_S_A = stereoN_to_S.restrict(A)
spher_to_stereoS = stereoN_to_S_A * spher_to_stereoN
spher_to_stereoS.display()
```

Out[33]:
$$\begin{cases} x' &= -\frac{\cos(\phi)\,\cos(\theta)-\cos(\phi)}{\sin(\theta)} \\ y' &= -\frac{\cos(\theta)\,\sin(\phi)-\sin(\phi)}{\sin(\theta)} \end{cases}$$

Similarly, the transition map $(A,(x',y')) \to (A,(\theta,\phi))$ is obtained by combining the transition maps $(A,(x',y')) \to (A,(x,y))$ and $(A,(x,y)) \to (A,(\theta,\phi))$:

In [34]:
```
stereoS_to_N_A = stereoN_to_S.inverse().restrict(A)
stereoS_to_spher = spher_to_stereoN.inverse() * stereoS_to_N_A
stereoS_to_spher.display()
```

Out[34]:
$$\begin{cases} \theta &= 2\arctan\left(\sqrt{x'^2+y'^2}\right) \\ \phi &= \pi - \arctan\left(\frac{y'}{x'^2+y'^2}, -\frac{x'}{x'^2+y'^2}\right) \end{cases}$$

The user atlas of $\mathbb{S}^2$ is now

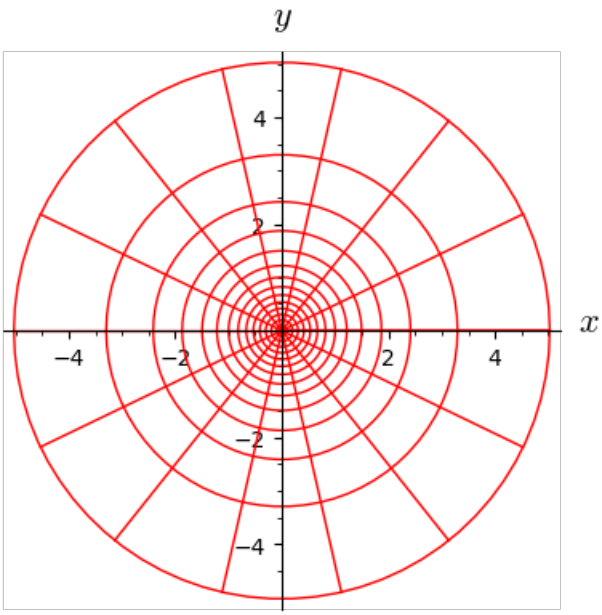
In [35]:
```
S2.atlas()
```

Out[35]:
$$\left[(U,(x,y)), \left(V,(x',y')\right), (W,(x,y)), \left(W,(x',y')\right), (A,(x,y)), \left(A,(x',y')\right), (A,(\theta,\phi))\right]$$

Let us draw the grid of spherical coordinates $(\theta,\phi)$ in terms of stereographic coordinates from the North pole $(x,y)$:

In [36]:
```
spher.plot(stereoN, number_values=15, ranges={th: (pi/8,pi)})
```

Out[36]:




## Points on $\mathbb{S}^2$

We declare the **North pole** (resp. the **South pole**) as the point of coordinates $(0,0)$ in the chart $(V,(x',y'))$ (resp. in the chart $(U,(x,y))$):

In [37]:
```
N = V.point((0,0), chart=stereoS, name='N') ; print(N)
S = U.point((0,0), chart=stereoN, name='S') ; print(S)
```

```
Point N on the 2-dimensional differentiable manifold S^2
Point S on the 2-dimensional differentiable manifold S^2
```

Since points are Sage `Element`'s, the corresponding `Parent` being the manifold subsets, an equivalent writing of the above declarations is

In [38]:
```
N = V((0,0), chart=stereoS, name='N') ; print(N)
S = U((0,0), chart=stereoN, name='S') ; print(S)
```

```
Point N on the 2-dimensional differentiable manifold S^2
Point S on the 2-dimensional differentiable manifold S^2
```

Moreover, since stereoS in the default chart on $V$ and stereoN is the default one on $U$, their mentions can be omitted, so that the above can be shortened to

In [39]:
```
N = V((0,0), name='N') ; print(N)
S = U((0,0), name='S') ; print(S)
```

```
Point N on the 2-dimensional differentiable manifold S^2
Point S on the 2-dimensional differentiable manifold S^2
```

In [40]:
```
N.parent()
```

Out[40]: $V$

In [41]:
```
S.parent()
```

Out[41]: $U$

We have of course

In [42]:
```
N in V
```

Out[42]: True

In [43]:
```
N in S2
```

Out[43]: True

In [44]:
```
N in U
```

Out[44]: False

In [45]:
```
N in A
```

Out[45]: False

Let us introduce some point at the equator:

In [46]:
```
E = S2((0,1), chart=stereoN, name='E')
```

The point $E$ is in the open subset $A$:

In [47]:
```
E in A
```

Out[47]: True

We may then ask for its spherical coordinates $(\theta, \phi)$:

In [48]:
```
E.coord(spher)
```

Out[48]: $\left(\frac{1}{2}\,\pi, \frac{1}{2}\,\pi\right)$

which is not possible for the point $N$:

In [49]:
```
try:
    N.coord(spher)
except ValueError as exc:
    print('Error: ' + str(exc))
```

```
Error: the point does not belong to the domain of Chart (A, (th, ph))
```

## Maps between manifolds: the embedding of $\mathbb{S}^2$ into $\mathbb{R}^3$

Let us first declare $\mathbb{R}^3$ as the 3-dimensional Euclidean space, denoting the Cartesian coordinates by $(X, Y, Z)$:

In [50]:
```
R3.<X,Y,Z> = EuclideanSpace(name='R^3', latex_name=r'\mathbb{R}^3', metric_name='h')
cart = R3.cartesian_coordinates()
cart
```

Out[50]: $\left(\mathbb{R}^3, (X, Y, Z)\right)$

The embedding of $\mathbb{S}^2$ into $\mathbb{R}^3$ is then defined by the standard formulas relating the stereographic coordinates to the ambient Cartesian ones when considering the **stereographic projection** from the point $(0, 0, 1)$ (North pole) or $(0, 0, -1)$ (South pole) to the equatorial plane $Z = 0$:

In [51]:
```
Phi = S2.diff_map(R3, {(stereoN, cart):
                       [2*x/(1+x^2+y^2), 2*y/(1+x^2+y^2),
                        (x^2+y^2-1)/(1+x^2+y^2)],
                       (stereoS, cart):
                       [2*xp/(1+xp^2+yp^2), 2*yp/(1+xp^2+yp^2),
                        (1-xp^2-yp^2)/(1+xp^2+yp^2)]},
                  name='Phi', latex_name=r'\Phi')
```

In [52]:
```
Phi.display()
```

Out[52]:
$$\begin{aligned}
\Phi :\quad & \mathbb{S}^2 & \longrightarrow\quad & \mathbb{R}^3 \\
\text{on } U :\quad & (x, y) & \longmapsto\quad & (X, Y, Z) = \left(\frac{2\,x}{x^2+y^2+1}, \frac{2\,y}{x^2+y^2+1}, \frac{x^2+y^2-1}{x^2+y^2+1}\right) \\
\text{on } V :\quad & \left(x', y'\right) & \longmapsto\quad & (X, Y, Z) = \left(\frac{2\,x'}{x'^2+y'^2+1}, \frac{2\,y'}{x'^2+y'^2+1}, -\frac{x'^2+y'^2-1}{x'^2+y'^2+1}\right)
\end{aligned}$$

In [53]:
```
Phi.parent()
```

Out[53]: $\mathrm{Hom}\left(\mathbb{S}^2, \mathbb{R}^3\right)$

In [54]:
```
print(Phi.parent())
```

```
Set of Morphisms from 2-dimensional differentiable manifold S^2 to Euclidean space R^3
in Category of smooth manifolds over Real Field with 53 bits of precision
```

In [55]:
```
Phi.parent() is Hom(S2, R3)
```

Out[55]: True

$\Phi$ maps points of $\mathbb{S}^2$ to points of $\mathbb{R}^3$:

In [56]:
```
N1 = Phi(N) ; print(N1) ; N1 ; N1.coord()
```

```
Point Phi(N) on the Euclidean space R^3
```

Out[56]: $(0, 0, 1)$

In [57]:
```
S1 = Phi(S) ; print(S1) ; S1 ; S1.coord()
```

```
Point Phi(S) on the Euclidean space R^3
```

Out[57]: $(0, 0, -1)$

In [58]:
```
E1 = Phi(E) ; print(E1) ; E1 ; E1.coord()
```

```
Point Phi(E) on the Euclidean space R^3
```

Out[58]: $(0, 1, 0)$

$\Phi$ has been defined in terms of the stereographic charts $(U, (x, y))$ and $(V, (x', y'))$, but we may ask its expression in terms of spherical coordinates. The latter is then computed by means of the transition map $(A, (x, y)) \to (A, (\theta, \phi))$:

In [59]:
```
Phi.expr(stereoN_A, cart)
```

Out[59]: $\left(\frac{2\,x}{x^2+y^2+1}, \frac{2\,y}{x^2+y^2+1}, \frac{x^2+y^2-1}{x^2+y^2+1}\right)$

In [60]:
```
Phi.expr(spher, cart)
```

Out[60]: $(\cos(\phi)\sin(\theta), \sin(\phi)\sin(\theta), \cos(\theta))$

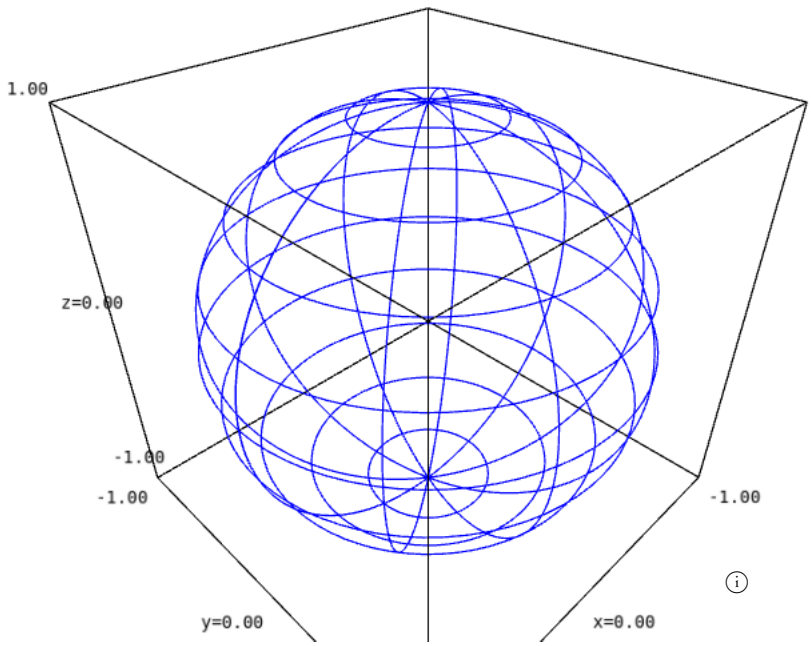
In [61]:
```
Phi.display(spher, cart)
```

Out[61]:
$$\begin{array}{llcl} \Phi: & \mathbb{S}^2 & \longrightarrow & \mathbb{R}^3 \\ \text{on } A: & (\theta, \phi) & \longmapsto & (X, Y, Z) = (\cos(\phi)\sin(\theta), \sin(\phi)\sin(\theta), \cos(\theta)) \end{array}$$

Let us use $\Phi$ to draw the grid of spherical coordinates $(\theta, \phi)$ in terms of the Cartesian coordinates $(X, Y, Z)$ of $\mathbb{R}^3$:

In [62]:
```
graph_spher = spher.plot(chart=cart, mapping=Phi, number_values=11,
                         color='blue', label_axes=False)
graph_spher
```

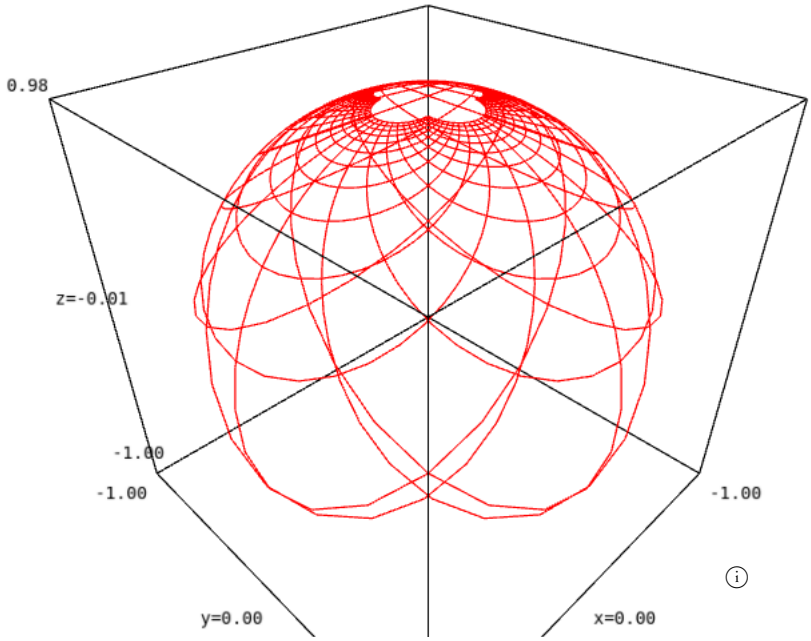
Out[62]:




We may also use the embedding $\Phi$ to display the stereographic coordinate grid in terms of the Cartesian coordinates in $\mathbb{R}^3$. First for the stereographic coordinates from the North pole:

In [63]:
```
graph_stereoN = stereoN.plot(chart=cart, mapping=Phi, number_values=25,
                             label_axes=False)
graph_stereoN
```

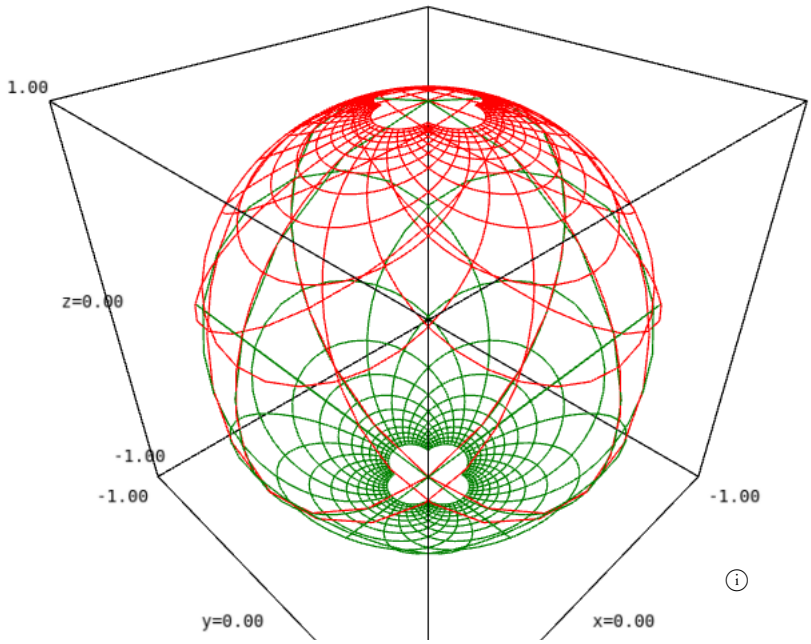
Out[63]:




and then have a view with the stereographic coordinates from the South pole superposed (in green):

In [64]:
```
graph_stereoS = stereoS.plot(chart=cart, mapping=Phi, number_values=25,
                             color='green', label_axes=False)
graph_stereoN + graph_stereoS
```

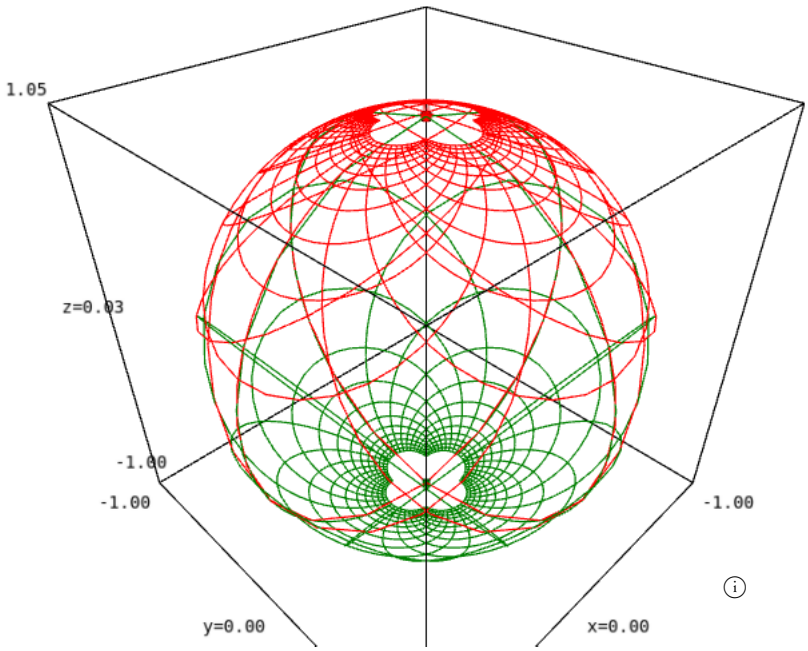
Out[64]:




We may also add the two poles to the graphic:

In [65]:
```
pointN = N.plot(chart=cart, mapping=Phi, color='red',
                label_offset=0.05)
pointS = S.plot(chart=cart, mapping=Phi, color='green',
                label_offset=0.05)
graph_stereoN + graph_stereoS + pointN + pointS
```

Out[65]:




## Tangent spaces

The tangent space to the manifold $\mathbb{S}^2$ at the point $N$ is

In [66]:
```
T_N = S2.tangent_space(N)
print(T_N) ; T_N
```

```
Tangent space at Point N on the 2-dimensional differentiable manifold S^2
```

Out[66]: $T_N\,\mathbb{S}^2$

$T_N\mathbb{S}^2$ is a vector space over $\mathbb{R}$ (represented here by Sage's symbolic ring SR):

In [67]:
```
print(T_N.category())
```

```
Category of finite dimensional vector spaces over Symbolic Ring
```

Its dimension equals the manifold's dimension:

In [68]:
```
dim(T_N)
```

Out[68]: $2$

In [69]:
```
dim(T_N) == dim(S2)
```

Out[69]: True

$T_N\mathbb{S}^2$ is endowed with a basis inherited from the coordinate frame defined around $N$, namely the frame associated with the chart $(V, (x', y'))$:

In [70]:
```
T_N.bases()
```

Out[70]:
$$\left[\left(\frac{\partial}{\partial x'}, \frac{\partial}{\partial y'}\right)\right]$$

$(V, (x', y'))$ is the only chart defined so far around the point $N$. If various charts have been defined around a point, then the tangent space at this point is automatically endowed with the bases inherited from the coordinate frames associated to all these charts. For instance, for the equator point $E$:

In [71]:
```
T_E = S2.tangent_space(E)
print(T_E) ; T_E
```

```
Tangent space at Point E on the 2-dimensional differentiable manifold S^2
```

Out[71]:
$$T_E\,\mathbb{S}^2$$

In [72]:
```
T_E.bases()
```

Out[72]:
$$\left[\left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right), \left(\frac{\partial}{\partial x'}, \frac{\partial}{\partial y'}\right), \left(\frac{\partial}{\partial \theta}, \frac{\partial}{\partial \phi}\right)\right]$$

In [73]:
```
T_E.default_basis()
```

Out[73]:
$$\left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right)$$

An element of $T_E\mathbb{S}^2$:

In [74]:
```
v = T_E((-3, 2), name='v')
print(v)
```

```
Tangent vector v at Point E on the 2-dimensional differentiable manifold S^2
```

In [75]:
```
v in T_E
```

Out[75]:
$$\text{True}$$

In [76]:
```
v.parent()
```

Out[76]:
$$T_E\,\mathbb{S}^2$$

In [77]:
```
v.display()
```

Out[77]:
$$v = -3\frac{\partial}{\partial x} + 2\frac{\partial}{\partial y}$$

In [78]:
```
v.display(T_E.bases()[1])
```

Out[78]:
$$v = -3\frac{\partial}{\partial x'} - 2\frac{\partial}{\partial y'}$$

In [79]:
```
v.display(T_E.bases()[2])
```

Out[79]:
$$v = -2\frac{\partial}{\partial \theta} + 3\frac{\partial}{\partial \phi}$$

## Differential of a smooth mapping

The differential of the mapping $\Phi$ at the point $E$ is

In [80]:
```
dPhi_E = Phi.differential(E)
print(dPhi_E) ; dPhi_E
```

```
Generic morphism:
  From: Tangent space at Point E on the 2-dimensional differentiable manifold S^2
  To:   Tangent space at Point Phi(E) on the Euclidean space R^3
```

Out[80]: $\mathrm{d}\Phi_E$

In [81]:
```
dPhi_E.domain()
```

Out[81]: $T_E\,\mathbb{S}^2$

In [82]:
```
dPhi_E.codomain()
```

Out[82]: $T_{\Phi(E)}\,\mathbb{R}^3$

In [83]:
```
dPhi_E.parent()
```

Out[83]: $\mathrm{Hom}\left(T_E\,\mathbb{S}^2, T_{\Phi(E)}\,\mathbb{R}^3\right)$

The image by $\mathrm{d}\Phi_E$ of the vector $v \in T_E\mathbb{S}^2$ introduced above is

In [84]:
```
dPhi_E(v)
```

Out[84]: $\mathrm{d}\Phi_E\,(v)$

In [85]:
```
print(dPhi_E(v))
```

```
Vector dPhi_E(v) at Point Phi(E) on the Euclidean space R^3
```

In [86]:
```
dPhi_E(v) in R3.tangent_space(Phi(E))
```

Out[86]: True

In [87]:
```
dPhi_E(v).display()
```

Out[87]: $\mathrm{d}\Phi_E\,(v) = -3e_X + 2e_Z$

## Algebra of scalar fields

The set $C^\infty(\mathbb{S}^2)$ of all smooth functions $\mathbb{S}^2 \to \mathbb{R}$ has naturally the structure of a commutative algebra over $\mathbb{R}$. $C^\infty(\mathbb{S}^2)$ is obtained via the method `scalar_field_algebra()` applied to the manifold $\mathbb{S}^2$:

In [88]:
```
CS = S2.scalar_field_algebra() ; CS
```

Out[88]: $C^\infty\left(\mathbb{S}^2\right)$

Since the algebra internal product is the pointwise multiplication, it is clearly commutative, so that $C^\infty(\mathbb{S}^2)$ belongs to Sage's category of commutative algebras:

In [89]:
```
CS.category()
```

Out[89]: $\mathbf{CommutativeAlgebras}_{\mathrm{SR}}$

The base ring of the algebra $C^\infty(\mathbb{S}^2)$ is the field $\mathbb{R}$, which is represented here by Sage's Symbolic Ring (SR):

In [90]:
```
CS.base_ring()
```

Out[90]: SR

Elements of $C^\infty(\mathbb{S}^2)$ are of course (smooth) scalar fields:

In [91]:
```
print(CS.an_element())
```

```
Scalar field on the 2-dimensional differentiable manifold S^2
```

This example element is the constant scalar field that takes the value 2:

In [92]:
```
CS.an_element().display()
```

Out[92]:

$$\begin{array}{llcl} & \mathbb{S}^2 & \longrightarrow & \mathbb{R} \\ \text{on } U: & (x, y) & \longmapsto & 2 \\ \text{on } V: & \left(x', y'\right) & \longmapsto & 2 \\ \text{on } A: & (\theta, \phi) & \longmapsto & 2 \end{array}$$

A specific element is the zero one:

In [93]:
```
f = CS.zero()
print(f)
```

```
Scalar field zero on the 2-dimensional differentiable manifold S^2
```

Scalar fields map points of $\mathbb{S}^2$ to real numbers:

In [94]:
```
f(N), f(E), f(S)
```

Out[94]: $(0, 0, 0)$

In [95]:
```
f.display()
```

Out[95]:

$$\begin{array}{llcl} 0: & \mathbb{S}^2 & \longrightarrow & \mathbb{R} \\ \text{on } U: & (x, y) & \longmapsto & 0 \\ \text{on } V: & \left(x', y'\right) & \longmapsto & 0 \\ \text{on } A: & (\theta, \phi) & \longmapsto & 0 \end{array}$$

Another specific element is the algebra unit element, i.e. the constant scalar field 1:

In [96]:
```
f = CS.one()
print(f)
```

```
Scalar field 1 on the 2-dimensional differentiable manifold S^2
```

In [97]:
```
f(N), f(E), f(S)
```

Out[97]: $(1, 1, 1)$

In [98]:
```
f.display()
```

Out[98]:

$$\begin{array}{llcl} 1: & \mathbb{S}^2 & \longrightarrow & \mathbb{R} \\ \text{on } U: & (x, y) & \longmapsto & 1 \\ \text{on } V: & \left(x', y'\right) & \longmapsto & 1 \\ \text{on } A: & (\theta, \phi) & \longmapsto & 1 \end{array}$$

Let us define a scalar field by its coordinate expression in the two stereographic charts:

In [99]:
```
f = CS({stereoN: pi - 2*atan(x^2+y^2), stereoS: 2*atan(xp^2+yp^2)})
f.display()
```

Out[99]:

$$\begin{array}{llcl} & \mathbb{S}^2 & \longrightarrow & \mathbb{R} \\ \text{on } U: & (x, y) & \longmapsto & \pi - 2\,\arctan\left(x^2 + y^2\right) \\ \text{on } V: & \left(x', y'\right) & \longmapsto & 2\,\arctan\left(x'^2 + y'^2\right) \\ \text{on } A: & (\theta, \phi) & \longmapsto & \pi + 2\,\arctan\left(\frac{\cos(\theta)+1}{\cos(\theta)-1}\right) \end{array}$$

Instead of using `CS()` (i.e. the Parent __call__ function), we may invoke the `scalar_field` method on the manifold to create $f$; this allows to pass the name of the scalar field:

In [100]:
```
f = S2.scalar_field({stereoN: pi - 2*atan(x^2+y^2), stereoS: 2*atan(xp^2+yp^2)}, name
='f')
f.display()
```

Out[100]:

$$\begin{array}{llcl} f: & \mathbb{S}^2 & \longrightarrow & \mathbb{R} \\ \text{on } U: & (x, y) & \longmapsto & \pi - 2\,\arctan\left(x^2 + y^2\right) \\ \text{on } V: & \left(x', y'\right) & \longmapsto & 2\,\arctan\left(x'^2 + y'^2\right) \\ \text{on } A: & (\theta, \phi) & \longmapsto & \pi + 2\,\arctan\left(\frac{\cos(\theta)+1}{\cos(\theta)-1}\right) \end{array}$$

In [101]:
```
f.parent()
```

Out[101]:

$$C^\infty\left(\mathbb{S}^2\right)$$

Internally, the various coordinate expressions of the scalar field are stored in the dictionary `_express`, whose keys are the charts:

In [102]:
```
f._express
```

Out[102]:

$$\left\{(U, (x, y)) : \pi - 2\,\arctan\left(x^2 + y^2\right), \left(V, \left(x', y'\right)\right) : 2\,\arctan\left(x'^2 + y'^2\right), (W, (x, y)) : \pi - 2\,\arctan\left(x^2 + \right.\right.$$
$$\left.\left.\left(x'^2 + y'^2\right), (A, (x, y)) : \pi - 2\,\arctan\left(x^2 + y^2\right), (A, (\theta, \phi)) : \pi + 2\,\arctan\left(\frac{\cos(\theta)}{\cos(\theta)}\right.\right.\right.$$

The expression in a specific chart is recovered by passing the chart as the argument of the method `display()`:

In [103]:
```
f.display(stereoS)
```

Out[103]:

$$\begin{array}{llcl} f: & \mathbb{S}^2 & \longrightarrow & \mathbb{R} \\ \text{on } V: & \left(x', y'\right) & \longmapsto & 2\,\arctan\left(x'^2 + y'^2\right) \end{array}$$

Scalar fields map the manifold's points to real numbers:

In [104]:
```
f(N)
```

Out[104]:

$$0$$

In [105]:
```
f(E)
```

Out[105]:

$$\frac{1}{2}\,\pi$$

In [106]:
```
f(S)
```

Out[106]:

$$\pi$$

We may define the restrictions of $f$ to the open subsets $U$ and $V$:

In [107]:
```
fU = f.restrict(U)
fU.display()
```

Out[107]:
$$\begin{aligned} f: \quad U &\longrightarrow \mathbb{R} \\ (x, y) &\longmapsto \pi - 2\,\arctan\left(x^2 + y^2\right) \\ \text{on } W: \quad (x', y') &\longmapsto 2\,\arctan\left(x'^2 + y'^2\right) \\ \text{on } A: \quad (\theta, \phi) &\longmapsto \pi + 2\,\arctan\left(\frac{\cos(\theta)+1}{\cos(\theta)-1}\right) \end{aligned}$$

In [108]:
```
fV = f.restrict(V)
fV.display()
```

Out[108]:
$$\begin{aligned} f: \quad V &\longrightarrow \mathbb{R} \\ (x', y') &\longmapsto 2\,\arctan\left(x'^2 + y'^2\right) \\ \text{on } W: \quad (x, y) &\longmapsto \pi - 2\,\arctan\left(x^2 + y^2\right) \\ \text{on } A: \quad (\theta, \phi) &\longmapsto \pi + 2\,\arctan\left(\frac{\cos(\theta)+1}{\cos(\theta)-1}\right) \end{aligned}$$

In [109]:
```
fU(E), fU(S)
```

Out[109]:
$$\left(\frac{1}{2}\,\pi, \pi\right)$$

In [110]:
```
fU.parent()
```

Out[110]:
$$C^\infty(U)$$

In [111]:
```
fV.parent()
```

Out[111]:
$$C^\infty(V)$$

In [112]:
```
CU = U.scalar_field_algebra()
fU.parent() is CU
```

Out[112]: True

A scalar field on $\mathbb{S}^2$ can be coerced to a scalar field on $U$, the coercion being simply the restriction:

In [113]:
```
CU.has_coerce_map_from(CS)
```

Out[113]: True

In [114]:
```
fU == CU(f)
```

Out[114]: True

The arithmetic of scalar fields:

In [115]:
```
g = f*f - 2*f
g.display()
```

Out[115]:
$$\begin{aligned} \mathbb{S}^2 &\longrightarrow \mathbb{R} \\ \text{on } U: \quad (x, y) &\longmapsto -2\,\pi + \pi^2 - 4\,(\pi - 1)\arctan\left(x^2 + y^2\right) + 4\,\arctan\left(x^2 + y^2\right)^2 \\ \text{on } V: \quad (x', y') &\longmapsto 4\,\arctan\left(x'^2 + y'^2\right)^2 - 4\,\arctan\left(x'^2 + y'^2\right) \\ \text{on } A: \quad (\theta, \phi) &\longmapsto -2\,\pi + \pi^2 + 4\,(\pi - 1)\arctan\left(\frac{\cos(\theta)+1}{\cos(\theta)-1}\right) + 4\,\arctan\left(\frac{\cos(\theta)+1}{\cos(\theta)-1}\right)^2 \end{aligned}$$

## Module of vector fields

The set $\mathfrak{X}(\mathbb{S}^2)$ of all smooth vector fields on $\mathbb{S}^2$ is a module over the algebra (ring) $C^\infty(\mathbb{S}^2)$. It is obtained by the method `vector_field_module()`:

In [116]:
```
XS = S2.vector_field_module()
XS
```

Out[116]: $\mathfrak{X}\left(\mathbb{S}^2\right)$

In [117]:
```
print(XS)
```

```
Module X(S^2) of vector fields on the 2-dimensional differentiable manifold S^2
```

In [118]:
```
XS.base_ring()
```

Out[118]: $C^\infty\left(\mathbb{S}^2\right)$

In [119]:
```
XS.category()
```

Out[119]: $\mathbf{Modules}_{C^\infty\left(\mathbb{S}^2\right)}$

$\mathfrak{X}(\mathbb{S}^2)$ is not a free module:

In [120]:
```
isinstance(XS, FiniteRankFreeModule)
```

Out[120]: False

because $\mathbb{S}^2$ is not a parallelizable manifold:

In [121]:
```
S2.is_manifestly_parallelizable()
```

Out[121]: False

On the contrary, the set $\mathfrak{X}(U)$ of smooth vector fields on $U$ is a free module:

In [122]:
```
XU = U.vector_field_module()
isinstance(XU, FiniteRankFreeModule)
```

Out[122]: True

because $U$ is parallelizable:

In [123]:
```
U.is_manifestly_parallelizable()
```

Out[123]: True

Due to the introduction of the stereographic coordinates $(x, y)$ on $U$, a basis has already been defined on the free module $\mathfrak{X}(U)$, namely the coordinate basis $(\partial/\partial x, \partial/\partial y)$:

In [124]:
```
XU.print_bases()
```

```
Bases defined on the Free module X(U) of vector fields on the Open subset U of the 2-d
imensional differentiable manifold S^2:
 - (U, (d/dx,d/dy)) (default basis)
```

In [125]:
```
XU.default_basis()
```

Out[125]: $\left(U, \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right)\right)$

Similarly

In [126]:
```
XV = V.vector_field_module()
XV.default_basis()
```

Out[126]: $\left(V, \left(\frac{\partial}{\partial x'}, \frac{\partial}{\partial y'}\right)\right)$

In [127]:
```
eU = XU.default_basis()
eV = XV.default_basis()
```

From the point of view of the open set $U$, eU is also the default vector frame:

In [128]:
```
eU is U.default_frame()
```

Out[128]: True

It is also the default vector frame on $\mathbb{S}^2$ (although not defined on the whole $\mathbb{S}^2$), for it is the first vector frame defined on an open subset of $\mathbb{S}^2$:

In [129]:
```
eU is S2.default_frame()
```

Out[129]: True

In [130]:
```
eV is V.default_frame()
```

Out[130]: True

Let us introduce a vector field on $\mathbb{S}^2$:

In [131]:
```
v = S2.vector_field(name='v')
v[eU,:] = [1, -2]
v.display(eU)
```

Out[131]: $v = \frac{\partial}{\partial x} - 2\frac{\partial}{\partial y}$

In [132]:
```
v.parent()
```

Out[132]: $\mathfrak{X}\left(\mathbb{S}^2\right)$

In [133]:
```
stereoSW = stereoS.restrict(W)
eSW = stereoSW.frame()
eSW
```

Out[133]: $\left(W, \left(\frac{\partial}{\partial x'}, \frac{\partial}{\partial y'}\right)\right)$

In [134]:
```
vW = v.restrict(W)
vW.display()
```

Out[134]: $v = \frac{\partial}{\partial x} - 2\frac{\partial}{\partial y}$

In [135]:
```
vW.parent()
```

Out[135]: $\mathfrak{X}(W)$

In [136]:
```
print(vW.parent())
```

```
Free module X(W) of vector fields on the Open subset W of the 2-dimensional differenti
able manifold S^2
```

In [137]:
```
vW.display(eSW)
```

Out[137]:
$$v = \left(-\frac{x^2 - 4\,xy - y^2}{x^4 + 2\,x^2y^2 + y^4}\right)\frac{\partial}{\partial x'} + \left(-\frac{2\left(x^2 + xy - y^2\right)}{x^4 + 2\,x^2y^2 + y^4}\right)\frac{\partial}{\partial y'}$$

In [138]:
```
vW.display(eSW, stereoSW)
```

Out[138]:
$$v = \left(-x'^2 + 4\,x'y' + y'^2\right)\frac{\partial}{\partial x'} + \left(-2\,x'^2 - 2\,x'y' + 2\,y'^2\right)\frac{\partial}{\partial y'}$$

We extend the definition of $v$ to $V$ thanks to the above expression:

In [139]:
```
v.add_comp_by_continuation(eV, W, chart=stereoS)
```

In [140]:
```
v.display(eV)
```

Out[140]:
$$v = \left(-x'^2 + 4\,x'y' + y'^2\right)\frac{\partial}{\partial x'} + \left(-2\,x'^2 - 2\,x'y' + 2\,y'^2\right)\frac{\partial}{\partial y'}$$

At this stage, the vector field $v$ is defined on the whole manifold $\mathbb{S}^2$; it has expressions in each of the two frames eU and eV which cover $\mathbb{S}^2$:

In [141]:
```
print(v)
v.display(eU)
```

```
Vector field v on the 2-dimensional differentiable manifold S^2
```

Out[141]:
$$v = \frac{\partial}{\partial x} - 2\frac{\partial}{\partial y}$$

In [142]:
```
v.display(eV)
```

Out[142]:
$$v = \left(-x'^2 + 4\,x'y' + y'^2\right)\frac{\partial}{\partial x'} + \left(-2\,x'^2 - 2\,x'y' + 2\,y'^2\right)\frac{\partial}{\partial y'}$$

According to the hairy ball theorem, $v$ has to vanish somewhere. This occurs at the North pole:

In [143]:
```
vN = v.at(N)
print(v)
```

```
Vector field v on the 2-dimensional differentiable manifold S^2
```

In [144]:
```
vN.display()
```

Out[144]:
$$v = 0$$

$v|_N$ is the zero vector of the tangent vector space $T_N\mathbb{S}^2$:

In [145]:
```
vN.parent()
```

Out[145]:
$$T_N\,\mathbb{S}^2$$

In [146]:
```
vN.parent() is S2.tangent_space(N)
```

Out[146]:
True

In [147]:
```
vN == S2.tangent_space(N).zero()
```

Out[147]: True

On the contrary, $v$ is non-zero at the South pole:

In [148]:
```
vS = v.at(S)
print(v)
```

```
Vector field v on the 2-dimensional differentiable manifold S^2
```

In [149]:
```
vS.display()
```

Out[149]: $v = \frac{\partial}{\partial x} - 2\frac{\partial}{\partial y}$

In [150]:
```
vS.parent()
```

Out[150]: $T_S\,\mathbb{S}^2$

In [151]:
```
vS.parent() is S2.tangent_space(S)
```

Out[151]: True

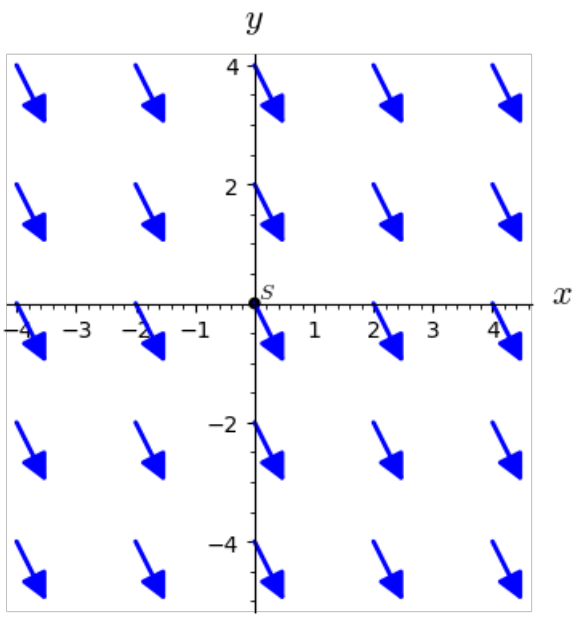
In [152]:
```
vS != S2.tangent_space(S).zero()
```

Out[152]: True

Let us plot the vector field $v$ is terms of the stereographic chart $(U, (x, y))$, with the South pole $S$ superposed:

In [153]:
```
v.plot(chart=stereoN, chart_domain=stereoN, max_range=4,
       number_values=5, scale=0.5, aspect_ratio=1) + \
 S.plot(stereoN, size=30, label_offset=0.2)
```

Out[153]:




The vector field appears homogeneous because its components w.r.t. the frame $\left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right)$ are constant:

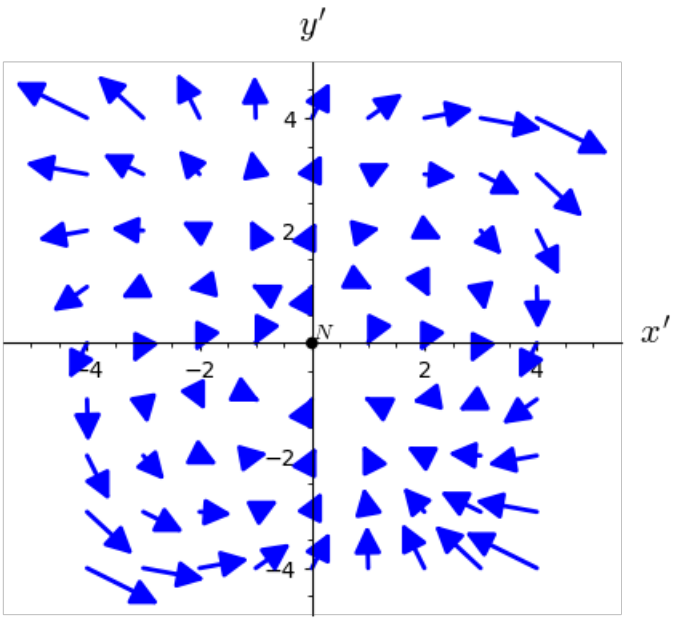
In [154]:
```
v.display(stereoN.frame())
```

Out[154]: $v = \frac{\partial}{\partial x} - 2\frac{\partial}{\partial y}$

On the contrary, once drawn in terms of the stereographic chart $(V, (x', y'))$, $v$ does no longer appears homogeneous:

In [155]:
```
v.plot(chart=stereoS, chart_domain=stereoS, max_range=4, scale=0.02,
       aspect_ratio=1) + \
 N.plot(chart=stereoS, size=30, label_offset=0.2)
```

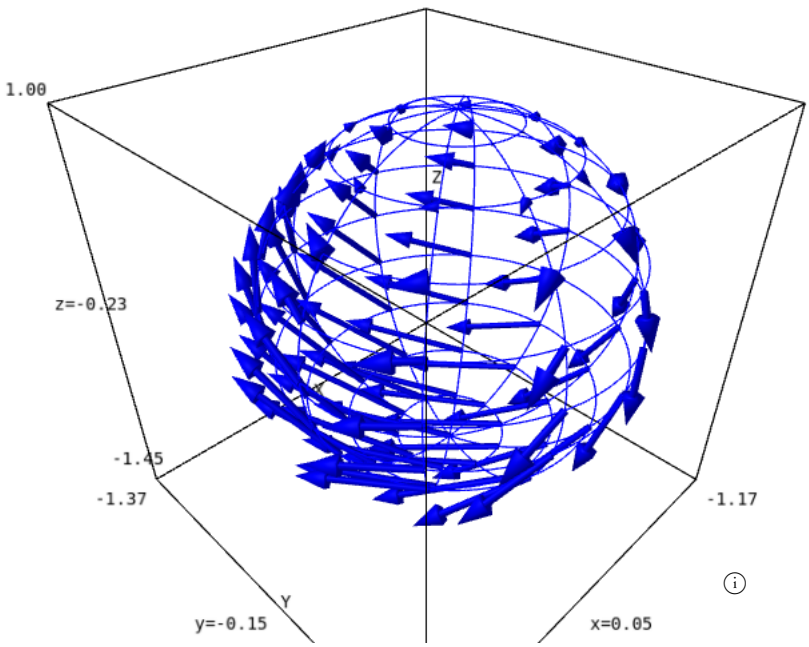
Out[155]:




Finally, a 3D view of the vector field $v$ is obtained via the embedding $\Phi$:

In [156]:
```
graph_v = v.plot(chart=cart, mapping=Phi, chart_domain=spher,
                 number_values=11, scale=0.2)
graph_spher + graph_v
```

Out[156]:



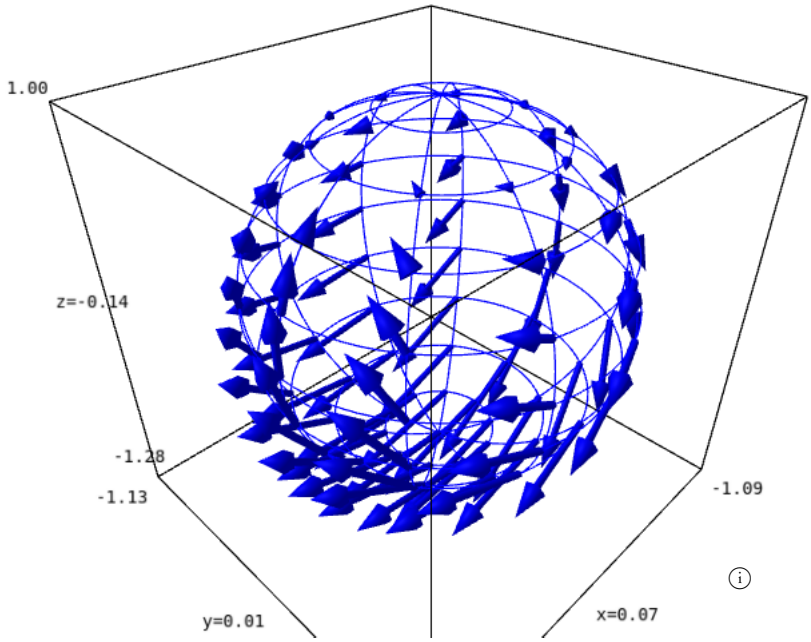

Similarly, let us draw the first vector field of the stereographic frame from the North pole, namely $\frac{\partial}{\partial x}$:

In [157]:
```
ex = stereoN.frame()[1]
ex
```

Out[157]: $\frac{\partial}{\partial x}$

In [158]:
```
graph_ex = ex.plot(chart=cart, mapping=Phi, chart_domain=spher,
                   number_values=11, scale=0.4, width=1,
                   label_axes=False)
graph_spher + graph_ex
```

Out[158]:



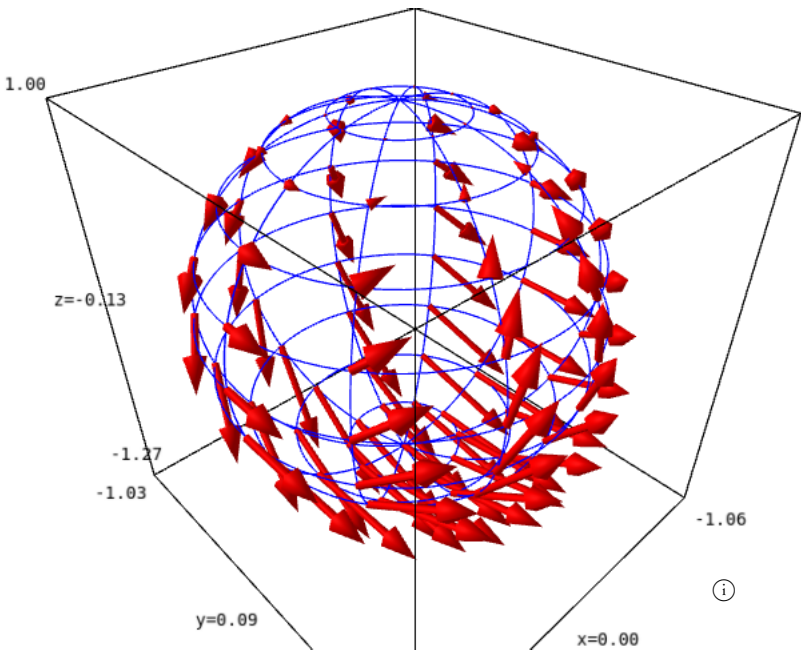

For the second vector field of the stereographic frame from the North pole, namely $\frac{\partial}{\partial y}$, we get

In [159]:
```
ey = stereoN.frame()[2]
ey
```

Out[159]: $\frac{\partial}{\partial y}$

In [160]:
```
graph_ey = ey.plot(chart=cart, mapping=Phi, chart_domain=spher,
                   number_values=11, scale=0.4, width=1, color='red',
                   label_axes=False)
graph_spher + graph_ey
```

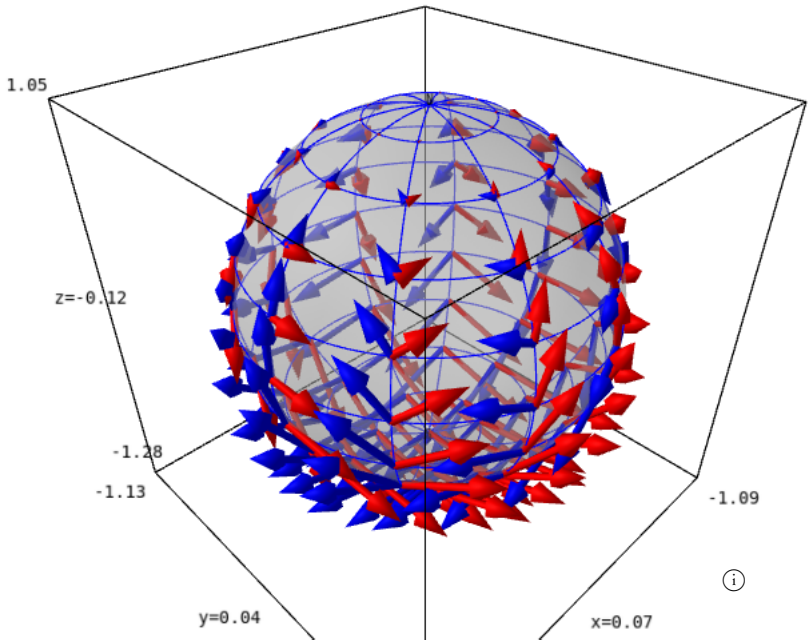
Out[160]:




We may combine the two graphs, to get a 3D view of the vector frame associated with the stereographic coordinates from the North pole:

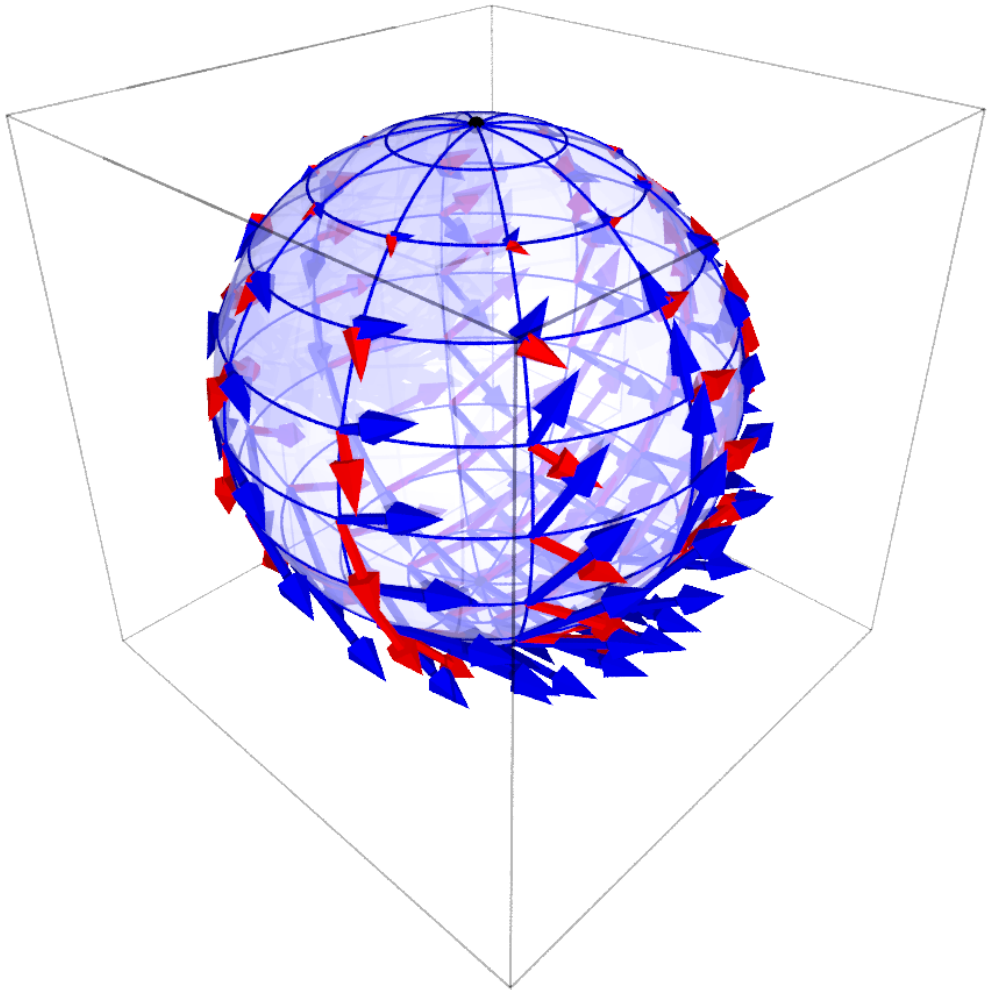
In [161]:
```
graph_frame = graph_spher + graph_ex + graph_ey + \
              N.plot(cart, mapping=Phi, label_offset=0.05, size=5) + \
              S.plot(cart, mapping=Phi, label_offset=0.05, size=5)
graph_frame + sphere(color='lightgrey', opacity=0.4)
```

Out[161]:




The same scene rendered with Tachyon:

In [162]:
```
show(graph_frame + sphere(opacity=0.5), viewer='tachyon', figsize=10)
```



## Vector fields acting on scalar fields

$v$ and $f$ are both fields defined on the whole sphere (respectively a vector field and a scalar field). By the very definition of a vector field, $v$ acts on $f$:

In [163]:
```
vf = v(f)
print(vf)
vf.display()
```

```
Scalar field v(f) on the 2-dimensional differentiable manifold S^2
```

Out[163]:

$$
\begin{aligned}
v(f):\ & \mathbb{S}^2 && \longrightarrow && \mathbb{R} \\
\text{on } U:\ & (x, y) && \longmapsto && -\frac{4\,(x-2\,y)}{x^4+2\,x^2y^2+y^4+1} \\
\text{on } V:\ & \left(x', y'\right) && \longmapsto && -\frac{4\left(x'^3-2\,x'^2y'+x'y'^2-2\,y'^3\right)}{x'^4+2\,x'^2y'^2+y'^4+1} \\
\text{on } A:\ & (\theta, \phi) && \longmapsto && -\frac{2\left(((\cos(\phi)-2\ \sin(\phi))\cos(\theta)-3\ \cos(\phi)+6\ \sin(\phi))\sin(\theta)^2-4\,(\cos(\phi)-2\ \sin(\phi))\cos(\theta)+4\ \cos(\phi)-8\ \sin(\right.}{\cos(\theta)^4-2\ \cos(\theta)^3+2\ \cos(\theta)^2-2\ \cos(\theta)+1}
\end{aligned}
$$

Values of $v(f)$ at the North pole, at the equator point $E$ and at the South pole:

In [164]:
```
vf(N)
```

Out[164]: $0$

In [165]:
```
vf(E)
```

Out[165]: $4$

In [166]:
```
vf(S)
```

Out[166]: $0$

## 1-forms

A 1-form on $\mathbb{S}^2$ is a field of linear forms on the tangent spaces. For instance it can the differential of a scalar field:

In [167]:
```
df = diff(f)
print(df)
```

```
1-form df on the 2-dimensional differentiable manifold S^2
```

In [168]:
```
df.display()
```

Out[168]:
$$\mathrm{d}f = \left(-\frac{4\,x}{x^4 + 2\,x^2y^2 + y^4 + 1}\right)\mathrm{d}x + \left(-\frac{4\,y}{x^4 + 2\,x^2y^2 + y^4 + 1}\right)\mathrm{d}y$$

In [169]:
```
print(df.parent())
```

```
Module Omega^1(S^2) of 1-forms on the 2-dimensional differentiable manifold S^2
```

In [170]:
```
df.parent()
```

Out[170]: $\Omega^1\left(\mathbb{S}^2\right)$

The 1-form acting on a vector field:

In [171]:
```
print(df(v)) ; df(v).display()
```

```
Scalar field df(v) on the 2-dimensional differentiable manifold S^2
```

Out[171]:
$$\begin{array}{llll} \mathrm{d}f(v): & \mathbb{S}^2 & \longrightarrow & \mathbb{R} \\ \text{on } U: & (x, y) & \longmapsto & -\frac{4\,(x-2\,y)}{x^4+2\,x^2y^2+y^4+1} \\ \text{on } V: & \left(x', y'\right) & \longmapsto & -\frac{4\left(x'^3-2\,x'^2y'+x'y'^2-2\,y'^3\right)}{x'^4+2\,x'^2y'^2+y'^4+1} \\ \text{on } A: & (\theta, \phi) & \longmapsto & -\frac{2\,(((\cos(\phi)-2\,\sin(\phi))\cos(\theta)-3\,\cos(\phi)+6\,\sin(\phi))\sin(\theta)^3-4\,((\cos(\phi)-2\,\sin(\phi))\cos(\theta)-\cos(\phi)+2\,\sin(}{\sin(\theta)^4+2\,(\cos(\theta)-2)\sin(\theta)^2-4\,\cos(\theta)+4} \end{array}$$

Let us check the identity $\mathrm{d}f(v) = v(f)$:

In [172]:
```
df(v) == v(f)
```

Out[172]: True

Similarly, we have $\mathcal{L}_v f = v(f)$:

In [173]:
```
f.lie_derivative(v) == v(f)
```

Out[173]: True

## Curves in $\mathbb{S}^2$

In order to define curves in $\mathbb{S}^2$, we first introduce the field of real numbers $\mathbb{R}$ as a 1-dimensional smooth manifold with a canonical coordinate chart:

In [174]:
```
R.<t> = RealLine() ; print(R)
```
```
Real number line R
```

In [175]:
```
R.category()
```
Out[175]: $\mathbf{Smooth_R}$

In [176]:
```
dim(R)
```
Out[176]: $1$

In [177]:
```
R.atlas()
```
Out[177]: $[(\mathbf{R}, (t))]$

Let us define a **loxodrome of the sphere** in terms of its parametric equation with respect to the chart `spher` = $(A, (\theta, \phi))$

In [178]:
```
c = S2.curve({spher: [2*atan(exp(-t/10)), t]}, (t, -oo, +oo), name='c')
```

Curves in $\mathbb{S}^2$ are considered as morphisms from the manifold $\mathbb{R}$ to the manifold $\mathbb{S}^2$:

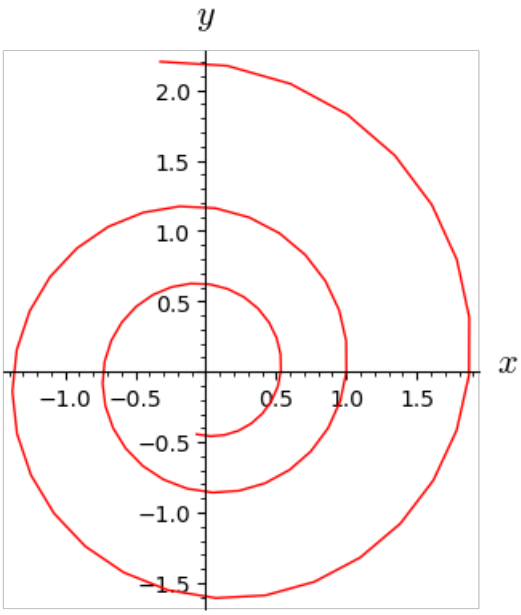
In [179]:
```
c.parent()
```
Out[179]: $\mathrm{Hom}\left(\mathbf{R}, \mathbb{S}^2\right)$

In [180]:
```
c.display()
```
Out[180]:
$$\begin{aligned} c: \mathbf{R} &\longrightarrow \mathbb{S}^2 \\ t &\longmapsto (x, y) = \left(\cos(t) e^{\left(\frac{1}{10}\, t\right)}, e^{\left(\frac{1}{10}\, t\right)} \sin(t)\right) \\ t &\longmapsto \left(x', y'\right) = \left(\cos(t) e^{\left(-\frac{1}{10}\, t\right)}, e^{\left(-\frac{1}{10}\, t\right)} \sin(t)\right) \\ t &\longmapsto (\theta, \phi) = \left(2\, \arctan\left(e^{\left(-\frac{1}{10}\, t\right)}\right), t\right) \end{aligned}$$

The curve $c$ can be plotted in terms of stereographic coordinates $(x, y)$:

In [181]:
```
c.plot(chart=stereoN, aspect_ratio=1)
```

Out[181]:

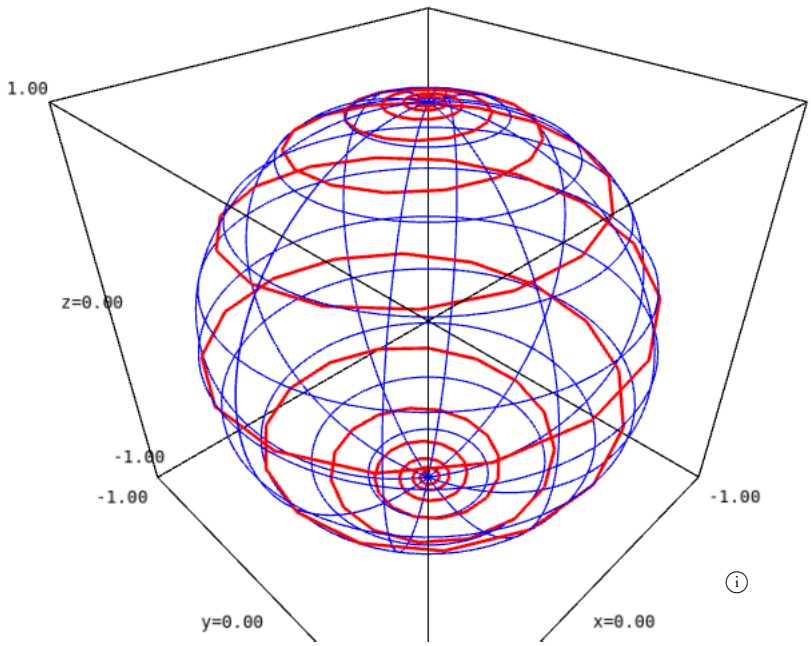



We recover the well-known fact that the graph of a loxodrome in terms of stereographic coordinates is a **logarithmic spiral**.

Thanks to the embedding $\Phi$, we may also plot $c$ in terms of the Cartesian coordinates of $\mathbb{R}^3$:

In [182]:
```
graph_c = c.plot(mapping=Phi, max_range=40, plot_points=200,
                 thickness=2, label_axes=False)
graph_spher + graph_c
```

Out[182]:




The **tangent vector field** (or **velocity vector**) to the curve $c$ is

In [183]:
```
vc = c.tangent_vector_field()
vc
```

Out[183]: $c'$

$c'$ is a vector field *along* $\mathbb{R}$ taking its values in tangent spaces to $\mathbb{S}^2$:

In [184]:
```
print(vc)
```

```
Vector field c' along the Real number line R with values on the 2-dimensional differen
tiable manifold S^2
```

The set of vector fields along $\mathbb{R}$ taking their values on $\mathbb{S}^2$ via the differential mapping $c : \mathbb{R} \to \mathbb{S}^2$ is denoted by $\mathfrak{X}(\mathbb{R}, c)$; it is a module over the algebra $C^\infty(\mathbb{R})$:

In [185]:
```
vc.parent()
```

Out[185]: $\mathfrak{X}\left(\mathbf{R}, c\right)$

In [186]:
```
vc.parent().category()
```

Out[186]: $\mathbf{Modules}_{C^\infty(\mathbf{R})}$

In [187]:
```
vc.parent().base_ring()
```

Out[187]: $C^\infty\left(\mathbf{R}\right)$

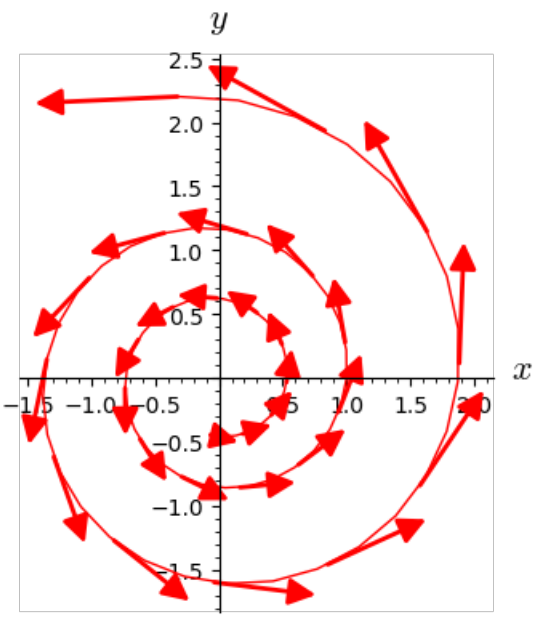
A coordinate view of $c'$:

In [188]:
```
vc.display()
```

Out[188]: $$c' = \left(\frac{1}{10}\cos(t)e^{\left(\frac{1}{10}\,t\right)} - e^{\left(\frac{1}{10}\,t\right)}\sin(t)\right)\frac{\partial}{\partial x} + \left(\cos(t)e^{\left(\frac{1}{10}\,t\right)} + \frac{1}{10}\,e^{\left(\frac{1}{10}\,t\right)}\sin(t)\right)\frac{\partial}{\partial y}$$

Let us plot the vector field $c'$ in terms of the stereographic chart $(U, (x, y))$:

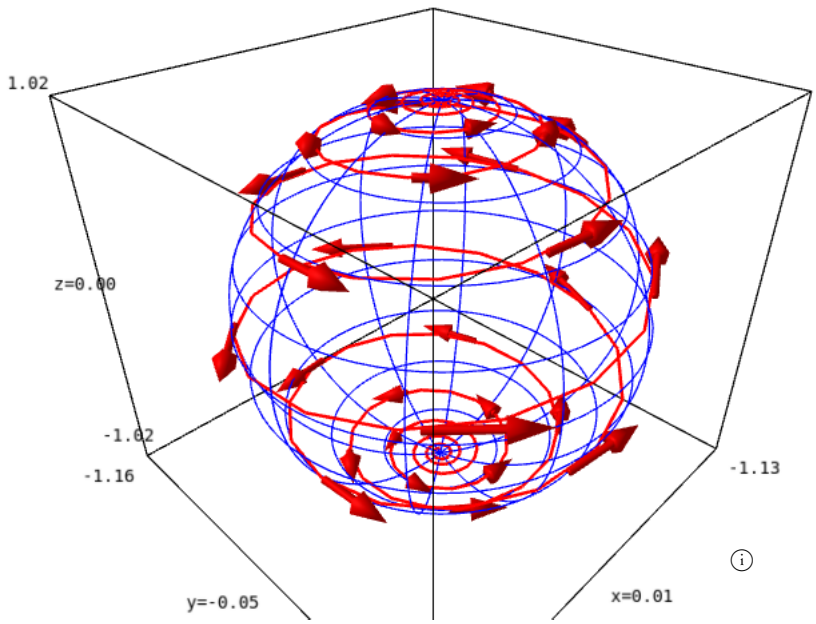
In [189]:
```
show(vc.plot(chart=stereoN, number_values=30, scale=0.5, color='red') +
     c.plot(chart=stereoN), aspect_ratio=1)
```




A 3D view of $c'$ is obtained via the embedding $\Phi$:

In [190]:
```
graph_vc = vc.plot(chart=cart, mapping=Phi, ranges={t: (-20, 20)},
                   number_values=30, scale=0.5, color='red',
                   label_axes=False)
graph_spher + graph_c + graph_vc
```

Out[190]:




## Riemannian metric on $\mathbb{S}^2$

The standard metric on $\mathbb{S}^2$ is that induced by the Euclidean metric of $\mathbb{R}^3$. The latter is obtained by:

In [191]:
```
h = R3.metric()
h.display()
```

Out[191]: $h = \mathrm{d}X \otimes \mathrm{d}X + \mathrm{d}Y \otimes \mathrm{d}Y + \mathrm{d}Z \otimes \mathrm{d}Z$

The metric $g$ on $\mathbb{S}^2$ is the pullback of $h$ by the embedding $\Phi$:

In [192]:
```
g = S2.metric('g')
g.set( Phi.pullback(h) )
print(g)
```

```
Riemannian metric g on the 2-dimensional differentiable manifold S^2
```

Note that we could have defined $g$ intrinsically, i.e. by providing its components in the two frames `stereoN` and `stereoS`, as we did for the metric $h$ on $\mathbb{R}^3$. Instead, we have chosen to get it as the pullback of $h$, as an example of pullback associated with some differential map.

The metric is a symmetric tensor field of type (0,2):

In [193]:
```
print(g.parent())
```

```
Module T^(0,2)(S^2) of type-(0,2) tensors fields on the 2-dimensional differentiable m
anifold S^2
```

In [194]:
```
g.tensor_type()
```

Out[194]: $(0, 2)$

In [195]:
```
g.symmetries()
```

```
symmetry: (0, 1); no antisymmetry
```

The expression of the metric in terms of the default frame on $\mathbb{S}^2$ (stereoN):

In [196]:
```
g.display()
```

Out[196]:
$$g = \left(\frac{4}{x^4 + y^4 + 2\left(x^2+1\right)y^2 + 2\,x^2 + 1}\right)\mathrm{d}x \otimes \mathrm{d}x + \left(\frac{4}{x^4 + y^4 + 2\left(x^2+1\right)y^2 + 2\,x^2 + 1}\right)\mathrm{d}y \otimes \mathrm{d}y$$

We may factorize the metric components:

In [197]:
```
g[1,1].factor(); g[2,2].factor()
```

Out[197]:
$$\frac{4}{\left(x^2 + y^2 + 1\right)^2}$$

In [198]:
```
g.display()
```

Out[198]:
$$g = \frac{4}{\left(x^2 + y^2 + 1\right)^2}\mathrm{d}x \otimes \mathrm{d}x + \frac{4}{\left(x^2 + y^2 + 1\right)^2}\mathrm{d}y \otimes \mathrm{d}y$$

A matrix view of the components of $g$ in the manifold's default frame:

In [199]:
```
g[:]
```

Out[199]:
$$\begin{pmatrix} \frac{4}{\left(x^2+y^2+1\right)^2} & 0 \\ 0 & \frac{4}{\left(x^2+y^2+1\right)^2} \end{pmatrix}$$

Display in terms of the vector frame $(V, (\partial_{x'}, \partial_{y'}))$:

In [200]:
```
g.display(stereoS.frame())
```

Out[200]:
$$g = \left(\frac{4}{x'^4 + y'^4 + 2\left(x'^2+1\right)y'^2 + 2\,x'^2 + 1}\right)\mathrm{d}x' \otimes \mathrm{d}x' + \left(\frac{4}{x'^4 + y'^4 + 2\left(x'^2+1\right)y'^2 + 2\,x'^2 + 1}\right)$$

In [201]:
```
g.display(spher.frame(), chart=spher)
```

Out[201]:
$$g = \mathrm{d}\theta \otimes \mathrm{d}\theta + \sin\left(\theta\right)^2\mathrm{d}\phi \otimes \mathrm{d}\phi$$

The metric acts on vector field pairs, resulting in a scalar field:

In [202]:
```
print(g(v,v))
```

```
Scalar field g(v,v) on the 2-dimensional differentiable manifold S^2
```

In [203]:
```
g(v,v).parent()
```

Out[203]:
$$C^\infty\left(\mathbb{S}^2\right)$$

In [204]:
```
g(v,v).display()
```

Out[204]:
$$\begin{array}{llll} g\left(v, v\right): & \mathbb{S}^2 & \longrightarrow & \mathbb{R} \\ \text{on } U: & (x, y) & \longmapsto & \frac{20}{x^4+y^4+2\left(x^2+1\right)y^2+2\,x^2+1} \\ \text{on } V: & \left(x', y'\right) & \longmapsto & \frac{20\left(x'^4+2\,x'^2y'^2+y'^4\right)}{x'^4+y'^4+2\left(x'^2+1\right)y'^2+2\,x'^2+1} \\ \text{on } A: & (\theta, \phi) & \longmapsto & 5\,\cos\left(\theta\right)^2 - 10\,\cos(\theta) + 5 \end{array}$$

The **Levi-Civita connection** associated with the metric $g$:

In [205]:
```
nab = g.connection()
print(nab)
nab
```

```
Levi-Civita connection nabla_g associated with the Riemannian metric g on the 2-dimens
ional differentiable manifold S^2
```

Out[205]: $\nabla_g$

As a test, we verify that $\nabla_g$ acting on $g$ results in zero:

In [206]:
```
nab(g).display()
```

Out[206]: $\nabla_g g = 0$

The nonzero Christoffel symbols of $g$ (skipping those that can be deduced by symmetry on the last two indices) w.r.t. two charts:

In [207]:
```
g.christoffel_symbols_display(chart=stereoN)
```

Out[207]:
$$\Gamma^{x}_{\,x\,x} = -\frac{2\,x}{x^2+y^2+1}$$
$$\Gamma^{x}_{\,x\,y} = -\frac{2\,y}{x^2+y^2+1}$$
$$\Gamma^{x}_{\,y\,y} = \frac{2\,x}{x^2+y^2+1}$$
$$\Gamma^{y}_{\,x\,x} = \frac{2\,y}{x^2+y^2+1}$$
$$\Gamma^{y}_{\,x\,y} = -\frac{2\,x}{x^2+y^2+1}$$
$$\Gamma^{y}_{\,y\,y} = -\frac{2\,y}{x^2+y^2+1}$$

In [208]:
```
g.christoffel_symbols_display(chart=spher)
```

Out[208]:
$$\Gamma^{\theta}_{\,\phi\,\phi} = -\cos(\theta)\sin(\theta)$$
$$\Gamma^{\phi}_{\,\theta\,\phi} = \frac{\cos(\theta)}{\sin(\theta)}$$

$\nabla_g$ acting on the vector field $v$:

In [209]:
```
print(nab(v))
```

```
Tensor field nabla_g(v) of type (1,1) on the 2-dimensional differentiable manifold S^2
```

In [210]:
```
nab(v).display(stereoN.frame())
```

Out[210]:
$$\nabla_g v = \left(-\frac{2\,(x-2\,y)}{x^2+y^2+1}\right)\frac{\partial}{\partial x}\otimes \mathrm{d}x + \left(-\frac{2\,(2\,x+y)}{x^2+y^2+1}\right)\frac{\partial}{\partial x}\otimes \mathrm{d}y + \left(\frac{2\,(2\,x+y)}{x^2+y^2+1}\right)\frac{\partial}{\partial y}\otimes \mathrm{d}x + \left(-\frac{2}{x^2}\right.$$

## Curvature

The Riemann tensor associated with the metric $g$:

In [211]:
```
Riem = g.riemann()
print(Riem)
Riem.display()
```

```
Tensor field Riem(g) of type (1,3) on the 2-dimensional differentiable manifold S^2
```

Out[211]:

$$\mathrm{Riem}\,(g) = \left(\frac{4}{x^4 + y^4 + 2\left(x^2+1\right)y^2 + 2\,x^2 + 1}\right)\frac{\partial}{\partial x}\otimes \mathrm{d}y\otimes \mathrm{d}x\otimes \mathrm{d}y + \left(-\frac{4}{x^4 + y^4 + 2\left(x^2+1\right)y^2}\right.$$

$$\left.\otimes \mathrm{d}x + \left(-\frac{4}{x^4 + y^4 + 2\left(x^2+1\right)y^2 + 2\,x^2 + 1}\right)\frac{\partial}{\partial y}\otimes \mathrm{d}x\otimes \mathrm{d}x\otimes \mathrm{d}y + \left(\frac{4}{x^4 + y^4 + 2\left(x^2+1\right)y^2 + 2\right.}\right.$$

The components of the Riemann tensor in the default frame on $\mathbb{S}^2$:

In [212]:
```
Riem.display_comp()
```

Out[212]:

$$\begin{aligned}
\mathrm{Riem}(g)^{x}{}_{y\,x\,y} &= \frac{4}{x^4+y^4+2\left(x^2+1\right)y^2+2\,x^2+1}\\
\mathrm{Riem}(g)^{x}{}_{y\,y\,x} &= -\frac{4}{x^4+y^4+2\left(x^2+1\right)y^2+2\,x^2+1}\\
\mathrm{Riem}(g)^{y}{}_{x\,x\,y} &= -\frac{4}{x^4+y^4+2\left(x^2+1\right)y^2+2\,x^2+1}\\
\mathrm{Riem}(g)^{y}{}_{x\,y\,x} &= \frac{4}{x^4+y^4+2\left(x^2+1\right)y^2+2\,x^2+1}
\end{aligned}$$

The components in the frame associated with spherical coordinates:

In [213]:
```
Riem.display_comp(spher.frame(), chart=spher)
```

Out[213]:

$$\begin{aligned}
\mathrm{Riem}(g)^{\theta}{}_{\phi\,\theta\,\phi} &= \sin(\theta)^2\\
\mathrm{Riem}(g)^{\theta}{}_{\phi\,\phi\,\theta} &= -\sin(\theta)^2\\
\mathrm{Riem}(g)^{\phi}{}_{\theta\,\theta\,\phi} &= -1\\
\mathrm{Riem}(g)^{\phi}{}_{\theta\,\phi\,\theta} &= 1
\end{aligned}$$

In [214]:
```
print(Riem.parent())
```

```
Module T^(1,3)(S^2) of type-(1,3) tensors fields on the 2-dimensional differentiable m
anifold S^2
```

In [215]:
```
Riem.symmetries()
```

```
no symmetry; antisymmetry: (2, 3)
```

The Riemann tensor associated with the Euclidean metric $h$ on $\mathbb{R}^3$ is identically zero:

In [216]:
```
h.riemann().display()
```

Out[216]:

$$\mathrm{Riem}\,(h) = 0$$

The Ricci tensor and the Ricci scalar:

In [217]:
```
Ric = g.ricci()
Ric.display()
```

Out[217]:

$$\mathrm{Ric}\,(g) = \left(\frac{4}{x^4 + y^4 + 2\left(x^2+1\right)y^2 + 2\,x^2 + 1}\right)\mathrm{d}x\otimes \mathrm{d}x + \left(\frac{4}{x^4 + y^4 + 2\left(x^2+1\right)y^2 + 2\,x^2 + 1}\right)\mathrm{d}y$$

In [218]:
```
R = g.ricci_scalar()
R.display()
```

Out[218]:
$$\begin{aligned} \mathrm{r}(g): \quad \mathbb{S}^2 &\longrightarrow \mathbb{R} \\ \text{on } U: \quad (x, y) &\longmapsto 2 \\ \text{on } V: \quad \left(x', y'\right) &\longmapsto 2 \\ \text{on } A: \quad (\theta, \phi) &\longmapsto 2 \end{aligned}$$

Hence we recover the fact that $(\mathbb{S}^2, g)$ is a Riemannian manifold of constant positive curvature.

In dimension 2, the Riemann curvature tensor is entirely determined by the Ricci scalar $R$ according to

$$R^i{}_{jlk} = \frac{R}{2}\left(\delta^i{}_k g_{jl} - \delta^i{}_l g_{jk}\right)$$

Let us check this formula here, under the form $R^i{}_{jlk} = -R g_{j[k}\delta^i{}_{l]}$:

In [219]:
```
delta = S2.tangent_identity_field()
Riem == - R*(g*delta).antisymmetrize(2,3)
```

Out[219]: True

Similarly the relation $\mathrm{Ric} = (R/2)\, g$ must hold:

In [220]:
```
Ric == (R/2)*g
```

Out[220]: True

## Manifold orientation and volume 2-form

In order to introduce the volume 2-form associated with the metric $g$, we need to define an orientation on $\mathbb{S}^2$ first. We choose the orientation so that the vector frame $(\partial/\partial x', \partial/\partial y')$ of the stereographic coordinates from the South pole is right-handed. This is somewhat natural, because the triplet $(\partial/\partial x', \partial/\partial y', n)$, where $n$ is the unit outward normal to $\mathbb{S}^2$, is right-handed with respect to the standard orientation of $\mathbb{R}^3$. On the contrary the triplet $(\partial/\partial x, \partial/\partial y, n)$ formed from stereographic coordinates from the North pole is left-handed (see the plot in `Out[161]` ). Actually, we can check that $(\partial/\partial x, \partial/\partial y)$ and $(\partial/\partial x', \partial/\partial y')$ lead to two opposite orientations, because the transition map $(x, y) \mapsto (x', y')$ has a negative Jacobian determinant:

In [221]:
```
stereoN_to_S.jacobian_det()
```

Out[221]:
$$-\frac{1}{x^4 + 2\,x^2y^2 + y^4}$$

We define the orientation via the method `set_orientation()` with a list of right-handed vector frames, whose domains form an open cover of $\mathbb{S}^2$. We therefore provide `stereoS.frame()` $= (\partial/\partial x', \partial/\partial y')$ (domain: $V$) and the "reversed" frame $(\partial/\partial y, \partial/\partial x)$ on $U$:

In [222]:
```
ex, ey = stereoN.frame()[:]
f = U.vector_frame('f', (ey, ex))
f[1].display(stereoN.frame()), f[2].display(stereoN.frame())
```

Out[222]:
$$\left(f_1 = \frac{\partial}{\partial y}, f_2 = \frac{\partial}{\partial x}\right)$$

In [223]:
```
S2.set_orientation([stereoS.frame(), f])
```

The **volume 2-form** or **Levi-Civita tensor** associated with $g$ is then

In [224]:
```
eps = g.volume_form()
print(eps)
eps.display()
```

```
2-form eps_g on the 2-dimensional differentiable manifold S^2
```

Out[224]:
$$\epsilon_g = \left( -\frac{4}{x^4 + y^4 + 2\left(x^2+1\right)y^2 + 2\,x^2 + 1} \right) \mathrm{d}x \wedge \mathrm{d}y$$

Notice the minus sign in the the above expression, which reflects the fact that the default frame $(\partial/\partial x, \partial/\partial y)$ is left-handed. On the contrary, we have

In [225]:
```
eps.display(stereoS.frame())
```

Out[225]:
$$\epsilon_g = \left( \frac{4}{x'^4 + y'^4 + 2\left(x'^2+1\right)y'^2 + 2\,x'^2 + 1} \right) \mathrm{d}x' \wedge \mathrm{d}y'$$

A nicer display is obtained by factorizing the components:

In [226]:
```
eps.apply_map(factor, frame=stereoS.frame(), keep_other_components=True)
eps.display(stereoS.frame())
```

Out[226]:
$$\epsilon_g = \frac{4}{\left(x'^2 + y'^2 + 1\right)^2} \mathrm{d}x' \wedge \mathrm{d}y'$$

The frame associated with spherical coordinates is right-handed and we recover the standard expression of the volume 2-form:

In [227]:
```
eps.display(spher.frame(), chart=spher)
```

Out[227]:
$$\epsilon_g = \sin(\theta) \mathrm{d}\theta \wedge \mathrm{d}\phi$$

The exterior derivative of the 2-form $\epsilon_g$:

In [228]:
```
print(diff(eps))
```

```
3-form deps_g on the 2-dimensional differentiable manifold S^2
```

Of course, since $\mathbb{S}^2$ has dimension 2, all 3-forms vanish identically:

In [229]:
```
diff(eps).display()
```

Out[229]:
$$\mathrm{d}\epsilon_g = 0$$

## Non-holonomic frames

Up to know, all the vector frames introduced on $\mathbb{S}^2$ have been coordinate frames. Let us introduce a non-coordinate frame on the open subset $A$. To ease the notations, we change first the default chart and default frame on $A$ to the spherical coordinate ones:

In [230]:
```
A.default_chart()
```

Out[230]:
$$(A, (x, y))$$

In [231]:
```
A.default_frame()
```

Out[231]:
$$\left(A, \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right)\right)$$

In [232]:
```
A.set_default_chart(spher)
A.set_default_frame(spher.frame())
A.default_chart()
```

Out[232]: $(A, (\theta, \phi))$

In [233]:
```
A.default_frame()
```

Out[233]: $\left(A, \left(\frac{\partial}{\partial\theta}, \frac{\partial}{\partial\phi}\right)\right)$

We introduce the new vector frame $e = \left(\frac{\partial}{\partial\theta}, \frac{1}{\sin\theta}\frac{\partial}{\partial\phi}\right)$:

In [234]:
```
d_dth, d_dph = spher.frame()[:]
e = A.vector_frame('e', (d_dth, 1/sin(th)*d_dph))
print(e)
e
```

```
Vector frame (A, (e_1,e_2))
```

Out[234]: $(A, (e_1, e_2))$

In [235]:
```
(e[1].display(), e[2].display())
```

Out[235]: $\left(e_1 = \frac{\partial}{\partial\theta}, e_2 = \frac{1}{\sin(\theta)}\frac{\partial}{\partial\phi}\right)$

In [236]:
```
A.frames()
```

Out[236]: $\left[\left(A, \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right)\right), \left(A, \left(\frac{\partial}{\partial x'}, \frac{\partial}{\partial y'}\right)\right), \left(A, \left(\frac{\partial}{\partial\theta}, \frac{\partial}{\partial\phi}\right)\right), (A, (e_X, e_Y, e_Z)), (A, (e_1, e_2))\right]$

The new frame is an orthonormal frame for the metric $g$:

In [237]:
```
g(e[1],e[1]).expr()
```

Out[237]: $1$

In [238]:
```
g(e[1],e[2]).expr()
```

Out[238]: $0$

In [239]:
```
g(e[2],e[2]).expr()
```

Out[239]: $1$

In [240]:
```
g[e,:]
```

Out[240]: $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

In [241]:
```
g.display(e)
```

Out[241]: $g = e^1 \otimes e^1 + e^2 \otimes e^2$

In [242]:
```
eps.display(e)
```

Out[242]: $\epsilon_g = e^1 \wedge e^2$

It is non-holonomic, since its structure coefficients are not identically zero:

In [243]:
```
e.structure_coeff()[:]
```

Out[243]:
$$\left[[[0, 0], [0, 0]], \left[\left[0, -\frac{\cos(\theta)}{\sin(\theta)}\right], \left[\frac{\cos(\theta)}{\sin(\theta)}, 0\right]\right]\right]$$

In [244]:
```
e[2].lie_derivative(e[1]).display(e)
```

Out[244]:
$$-\frac{\cos(\theta)}{\sin(\theta)} e_2$$

while we have of course

In [245]:
```
spher.frame().structure_coeff()[:]
```

Out[245]: $[[[0, 0], [0, 0]], [[0, 0], [0, 0]]]$

In [246]:
```
print("Total elapsed time: {} s".format(time.perf_counter() - comput_time0))
```

```
Total elapsed time: 329.37821549399996 s
```

In [ ]:
```

```