Sphere $\mathbb{S}^2$

This worksheet demonstrates a few capabilities of SageManifolds (version 1.0, as included in SageMath 7.5) on the example of the 2-dimensional sphere.

Click here to download the worksheet file (ipynb format). To run it, you must start SageMath with the Jupyter notebook, via the command `sage -n jupyter`

NB: a version of SageMath at least equal to 7.5 is required to run this worksheet:

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

Out[1]:
```
'SageMath version 7.5, Release Date: 2017-01-11'
```

First we set up the notebook to display mathematical objects using LaTeX formatting:

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

We also define a viewer for 3D plots (use `'threejs'` or `'jmol'` for interactive 3D graphics):

In [3]:
```
viewer3D = 'jmol' # must be 'threejs', jmol', 'tachyon' or None (defaul
t)
```

$\mathbb{S}^2$ 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 to $\{0, 1\}$.

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()
```

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

In [14]:
```
y is stereoN[2]
```
Out[14]: True

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

In [15]:
```
stereoS.<xp,yp> = V.chart(r"xp:x' yp:y'")
```

In [16]:
```
stereoS
```
Out[16]: $\left(V, (x', y')\right)$

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

In [17]:
```
S2.atlas()
```
Out[17]: $\left[(U, (x, y)), \left(V, (x', y')\right)\right]$

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 [18]:
```
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, restri
ctions2= xp^2+xp^2!=0)
stereoN_to_S.display()
```

Out[18]:
$$\left\{\begin{array}{lcl} x' & = & \frac{x}{x^2+y^2} \\ y' & = & \frac{y}{x^2+y^2} \end{array}\right.$$

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 [19]:
```
stereoS_to_N = stereoN_to_S.inverse()
stereoS_to_N.display()
```

Out[19]:
$$\left\{\begin{array}{lcl} x & = & \frac{x'}{x'^2+y'^2} \\ y & = & \frac{y'}{x'^2+y'^2} \end{array}\right.$$

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 [20]:
```
S2.atlas()
```

Out[20]:
$$\left[(U, (x, y)), (V, (x', y')), (W, (x, y)), (W, (x', y'))\right]$$

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

In [21]:
```
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 [22]:
```
stereoN_W = stereoN.restrict(W)
stereoN_W
```

Out[22]:
$$(W, (x, y))$$

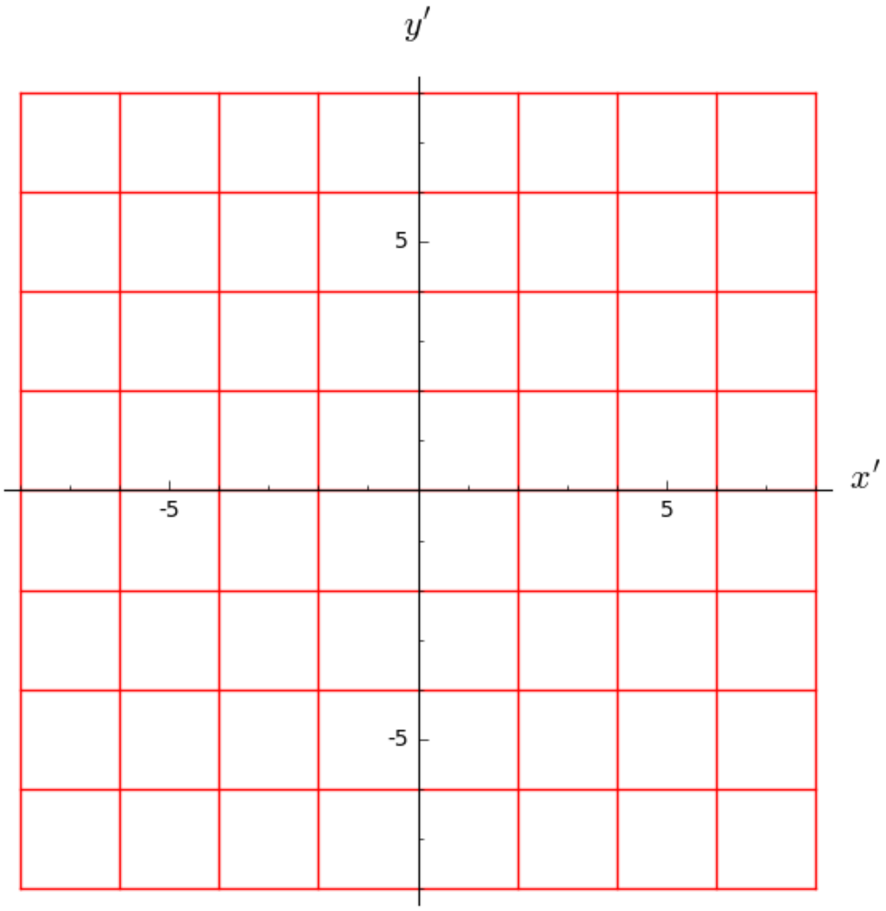
In [23]:
```
stereoS_W = stereoS.restrict(W)
stereoS_W
```

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

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

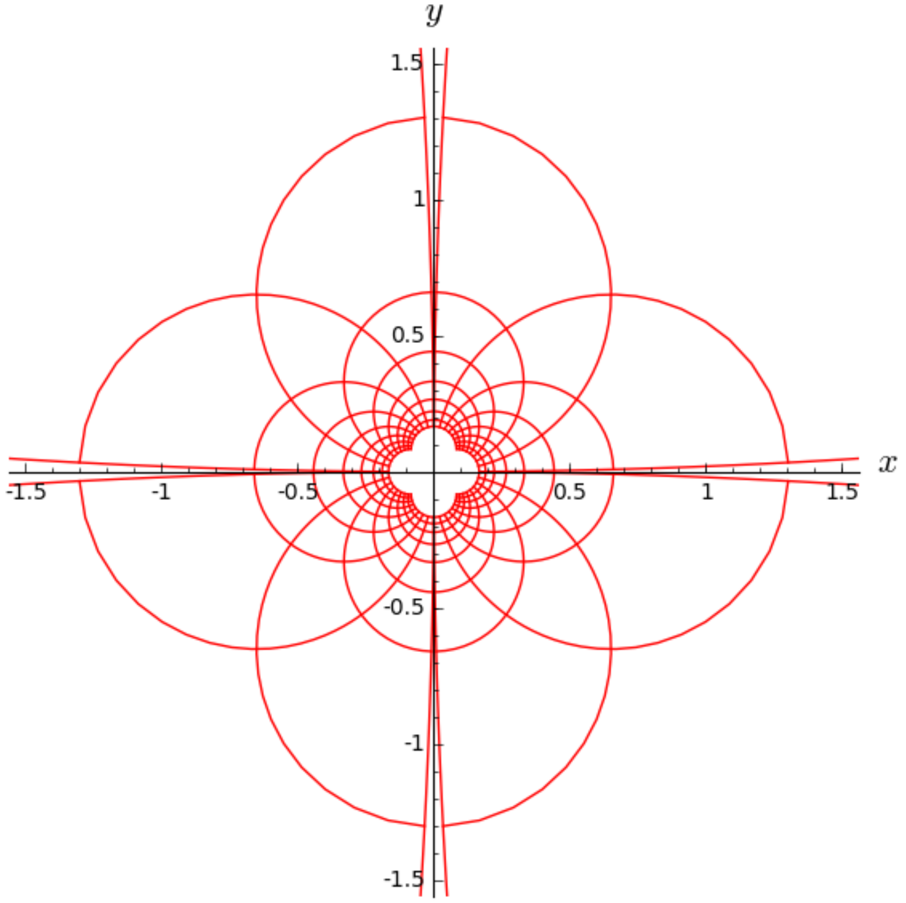
In [24]:
```
stereoS_W.plot()
```

Out[24]:



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 [25]:
```
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 (θ, ϕ) 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 [26]:
```
A = W.open_subset('A', coord_def={stereoN_W: (y!=0, x<0), stereoS_W: (y
p!=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 [27]:
```
stereoN_A = stereoN_W.restrict(A)
stereoN_A
```

Out[27]: $(A, (x, y))$

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

In [28]:
```
spher.<th,ph> = A.chart(r'th:(0,pi):\theta ph:(0,2*pi):\phi') ; spher
```

Out[28]: $(A,(\theta,\phi))$

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

In [29]:
```
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[29]:
$$\left\{\begin{array}{lcl} x & = & -\frac{\cos(\phi)\sin(\theta)}{\cos(\theta)-1} \\ y & = & -\frac{\sin(\phi)\sin(\theta)}{\cos(\theta)-1} \end{array}\right.$$

We also provide the inverse transition map:

In [30]:
```
spher_to_stereoN.set_inverse(2*atan(1/sqrt(x^2+y^2)), atan2(-y,-x)+pi,
                             check=True)
```

In [31]:
```
spher_to_stereoN.inverse().display()
```

Out[31]:
$$\left\{\begin{array}{lcl} \theta & = & 2\arctan\left(\frac{1}{\sqrt{x^2+y^2}}\right) \\ \phi & = & \pi+\arctan(-y,-x) \end{array}\right.$$

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

In [32]:
```
stereoN_to_S_A = stereoN_to_S.restrict(A)
spher_to_stereoS = stereoN_to_S_A * spher_to_stereoN
spher_to_stereoS.display()
```

Out[32]:
$$\left\{\begin{array}{lcl} x' & = & -\frac{\cos(\phi)\cos(\theta)-\cos(\phi)}{\sin(\theta)} \\ y' & = & -\frac{\cos(\theta)\sin(\phi)-\sin(\phi)}{\sin(\theta)} \end{array}\right.$$

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

In [33]:
```
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[33]:
$$\left\{\begin{array}{lcl} \theta & = & 2\arctan\left(\sqrt{x'^2+y'^2}\right) \\ \phi & = & \pi-\arctan\left(\frac{y'}{x'^2+y'^2},-\frac{x'}{x'^2+y'^2}\right) \end{array}\right.$$

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

In [34]:
```
S2.atlas()
```

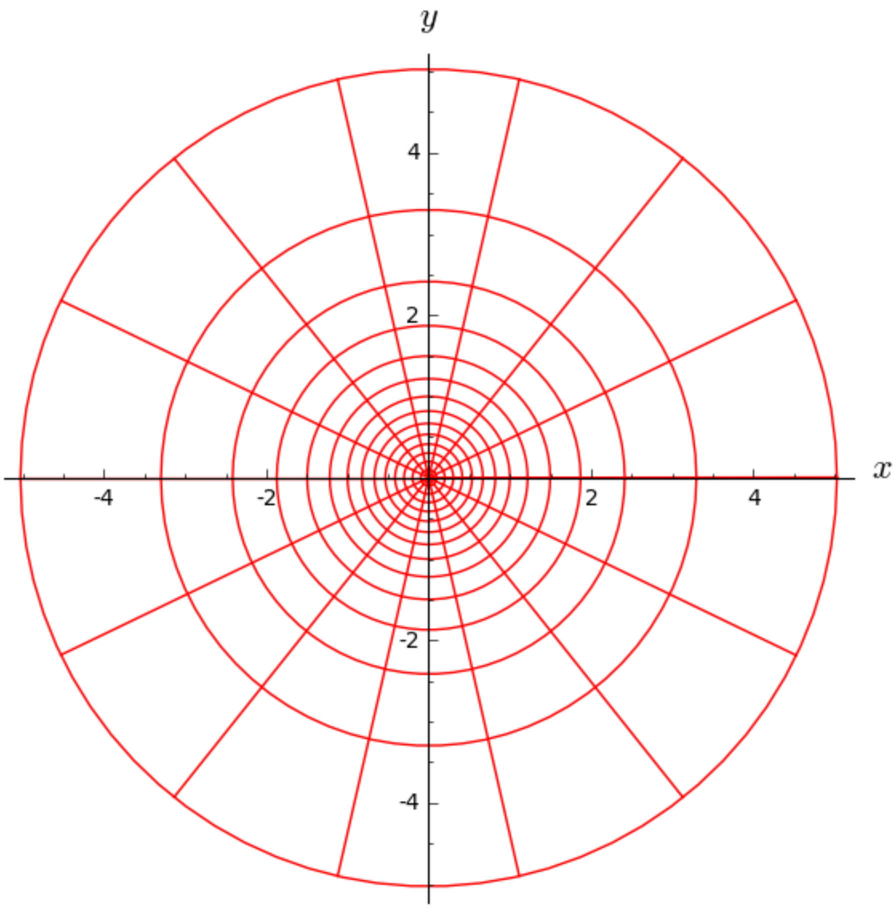
Out[34]:

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

Let us draw the grid of spherical coordinates (θ, ϕ) in terms of stereographic coordinates from the North pole (x, y):

In [35]:
```
spher.plot(stereoN, number_values=15, ranges={th: (pi/8,pi)})
```

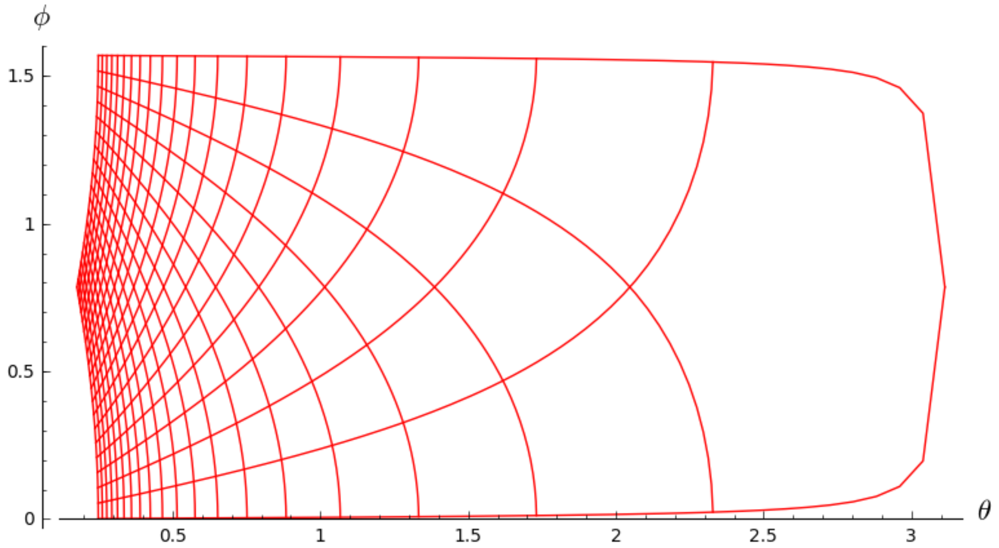
Out[35]:



Conversly, we may represent the grid of the stereographic coordinates (x, y) restricted to A in terms of the spherical coordinates (θ, ϕ). We limit ourselves to one quarter (cf. the argument `ranges`):

In [36]:
```
stereoN_A.plot(spher, ranges={x: (0.01,8), y: (0.01,8)}, number_values=
20, plot_points=200)
```

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 (θ, ϕ):

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))
```

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

Let us first declare $\mathbb{R}^3$ as a 3-dimensional manifold covered by a single chart (the so-called **Cartesian coordinates**):

In [50]:
```
R3 = Manifold(3, 'R^3', r'\mathbb{R}^3', start_index=1)
cart.<X,Y,Z> = R3.chart() ; cart
```
Out[50]: $\left(\mathbb{R}^3, (X, Y, Z)\right)$

The embedding of the sphere is defined as a differential mapping $\Phi : \mathbb{S}^2 \to \mathbb{R}^3$:

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]:

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

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 3-di
mensional differentiable manifold R^3 in Category of smooth manifolds o
ver Real Field with 53 bits of precision
```

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

Out[55]: True

Φ 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 3-dimensional differentiable manifold R^3
```

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

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

```
Point Phi(S) on the 3-dimensional differentiable manifold R^3
```

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

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

```
Point Phi(E) on the 3-dimensional differentiable manifold R^3
```

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

Φ 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)
```

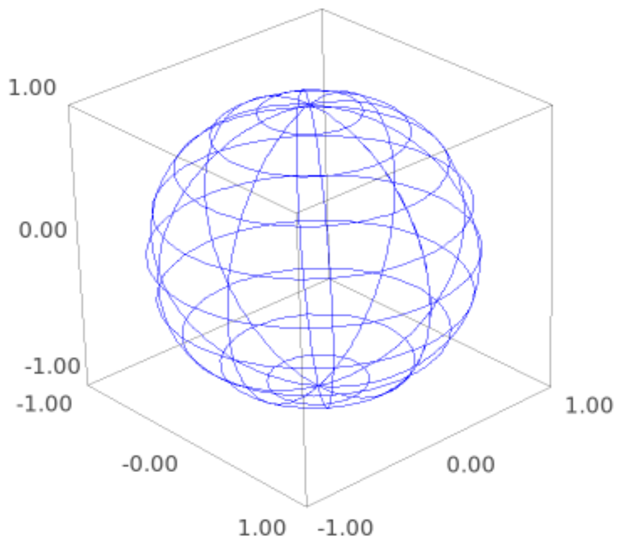
Out[60]: $(\cos(\phi)\sin(\theta), \sin(\phi)\sin(\theta), \cos(\theta))$

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

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

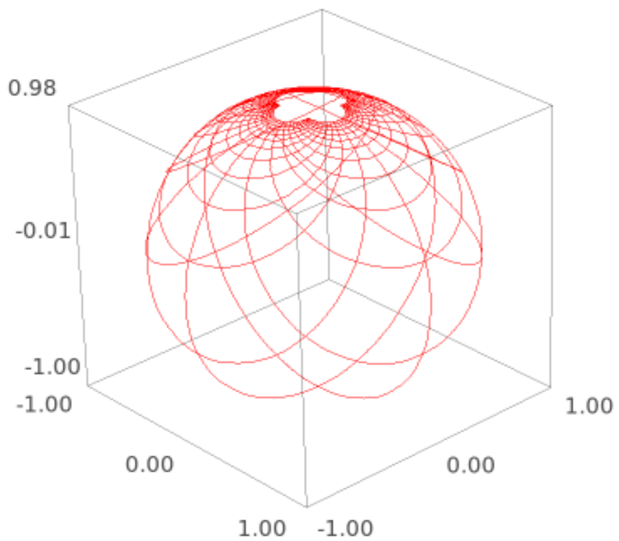
Let us use Φ to draw the grid of spherical coordinates (θ, ϕ) in terms of the Cartesian coordinates (X, Y, Z) of $\mathbb{R}^3$:

In [62]:
```
graph_spher = spher.plot(chart=cart, mapping=Phi, number_values=11, col
or='blue',
                         label_axes=False)
show(graph_spher, viewer=viewer3D)
```



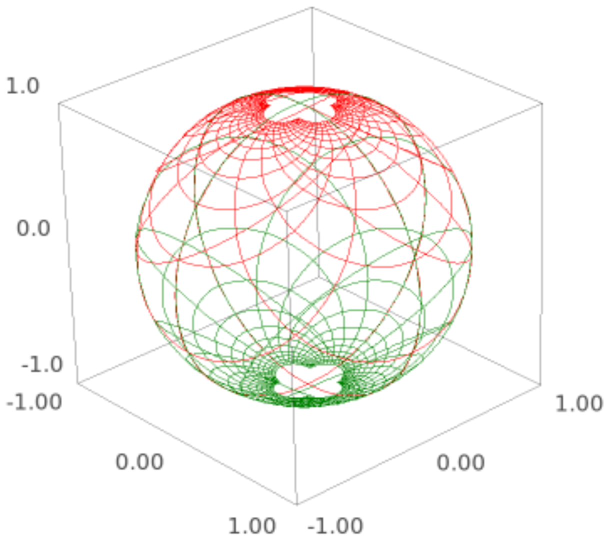
We may also use the embedding Φ to display the stereographic coordinate grid in terms of the Cartesian coordinates in $\mathbb{R}^3$. First for the stereographic coordinates from the North pole:

In [63]:
```
graph_stereoN = stereoN.plot(chart=cart, mapping=Phi, number_values=25,
 label_axes=False)
show(graph_stereoN, viewer=viewer3D)
```



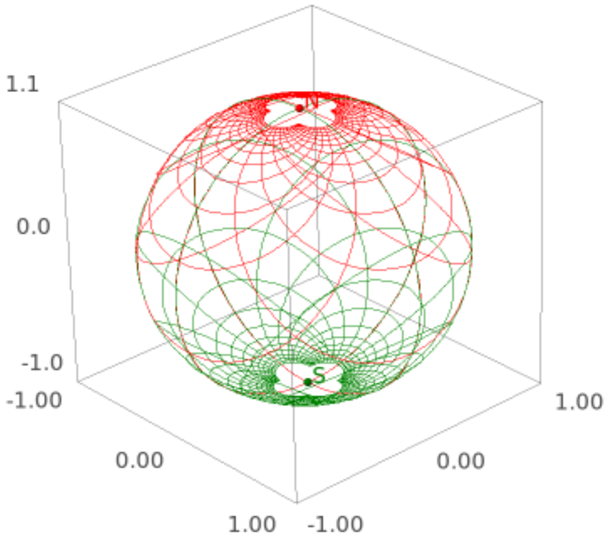
and then have a view with the stereographic coordinates from the South pole superposed (in green):

In [64]:
```
graph_stereoS = stereoS.plot(chart=cart, mapping=Phi, number_values=25,
 color='green',
                                label_axes=False)
show(graph_stereoN + graph_stereoS, viewer=viewer3D)
```



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)
show(graph_stereoN + graph_stereoS + pointN + pointS, viewer=viewer3D)
```



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 manifol
d S^2
```

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

Out[75]: 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 Φ 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 ma
nifold S^2
  To:   Tangent space at Point Phi(E) on the 3-dimensional differentiab
le manifold 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))
```

```
Tangent vector dPhi_E(v) at Point Phi(E) on the 3-dimensional different
iable manifold R^3
```

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

Out[86]: True

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

Out[87]:

$$\mathrm{d}\Phi_E\,(v) = 3\frac{\partial}{\partial X} - 2\frac{\partial}{\partial Z}$$

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

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

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

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]:
$$\left\{(U, (x, y)) : \pi - 2 \arctan\left(x^2 + y^2\right), (A, (x, y)) : \pi - 2 \arctan\left(x^2 + y^2\right), \right.$$
$$\left. (A, (\theta, \phi)) : \pi + 2 \arctan\left(\frac{\cos(\theta) + 1}{\cos(\theta) - 1}\right), \left(V, (x', y')\right) : 2 \arctan\left(x'^2 + y'^2\right)\right\}$$

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

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]: π

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

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

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

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

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

In [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]:

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

Module of vector fields

The set $\mathcal{X}(\mathbb{S}^2)$ 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]: $\mathcal{X}\left(\mathbb{S}^2\right)$

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

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

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

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

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

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

$\mathcal{X}(\mathbb{S}^2)$ 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 $\mathcal{X}(U)$ 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 $\mathcal{X}(U)$, 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 subs
et U of the 2-dimensional 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]: $\mathcal{X}\left(\mathbb{S}^2\right)$

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]: $\mathcal{X}(W)$

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

```
Free module X(W) of vector fields on the Open subset W of the 2-dimensi
onal differentiable 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]:

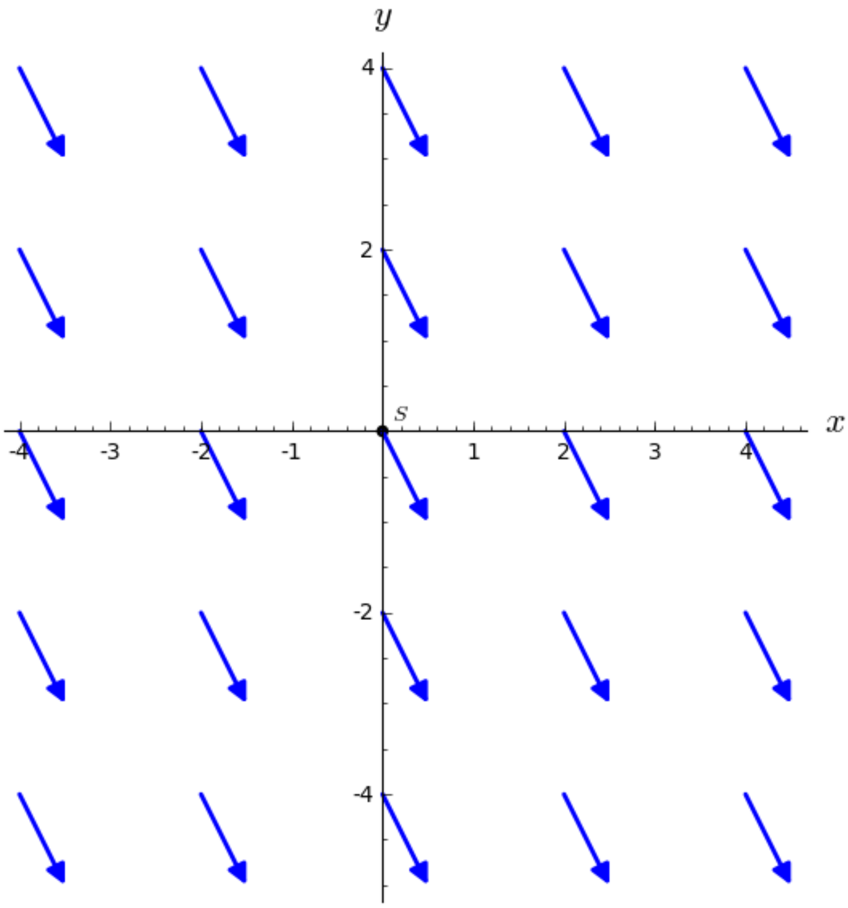
$$T_S\,\mathbb{S}^2$$

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:

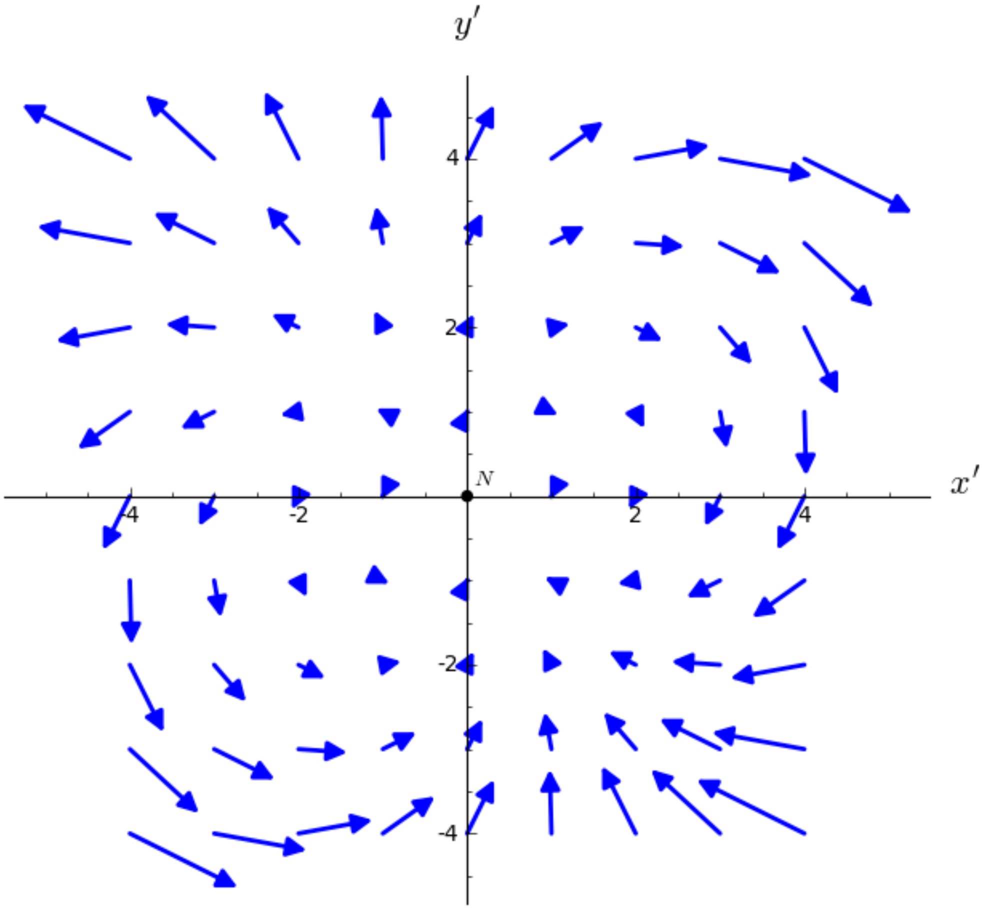
In [154]:
```
v.display(stereoN.frame())
```
Out[154]:

$$v = \frac{\partial}{\partial x} - 2\frac{\partial}{\partial y}$$

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

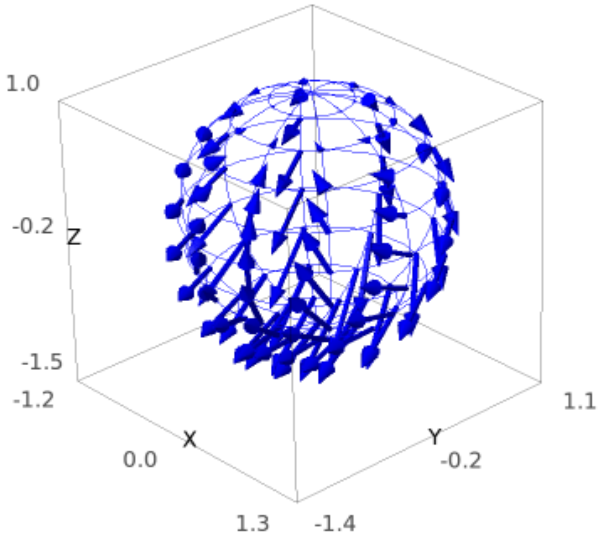
In [155]:
```
v.plot(chart=stereoS, chart_domain=stereoS, max_range=4, scale=0.02, as
pect_ratio=1) + \
 N.plot(chart=stereoS, size=30, label_offset=0.2)
```

Out[155]:



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

In [156]:
```
graph_v = v.plot(chart=cart, mapping=Phi, chart_domain=spher, number_va
lues=11, scale=0.2)
show(graph_spher + graph_v, viewer=viewer3D)
```

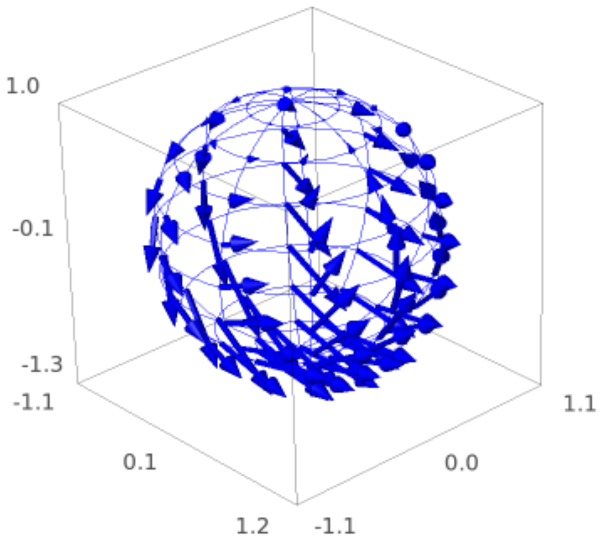


Similarly, let us draw the first vector field of the stereographic frame from the North pole, namely $\frac{\partial}{\partial x}$:

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=Fal
se)
show(graph_spher + graph_ex, viewer=viewer3D)
```

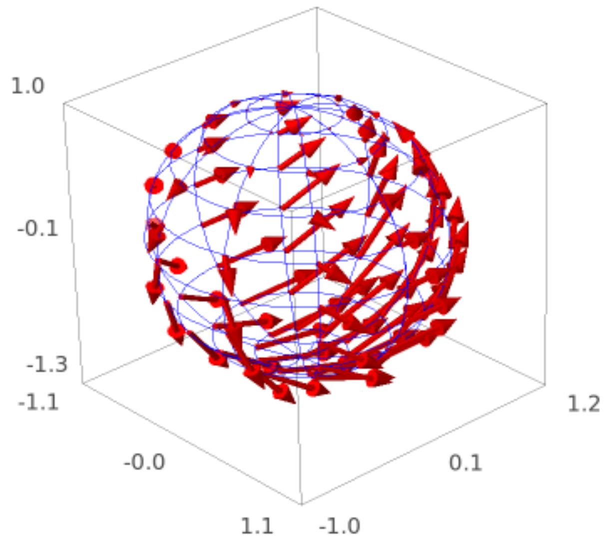


For the second vector field of the stereographic frame from the North pole, namely $\frac{\partial}{\partial y}$, we get

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

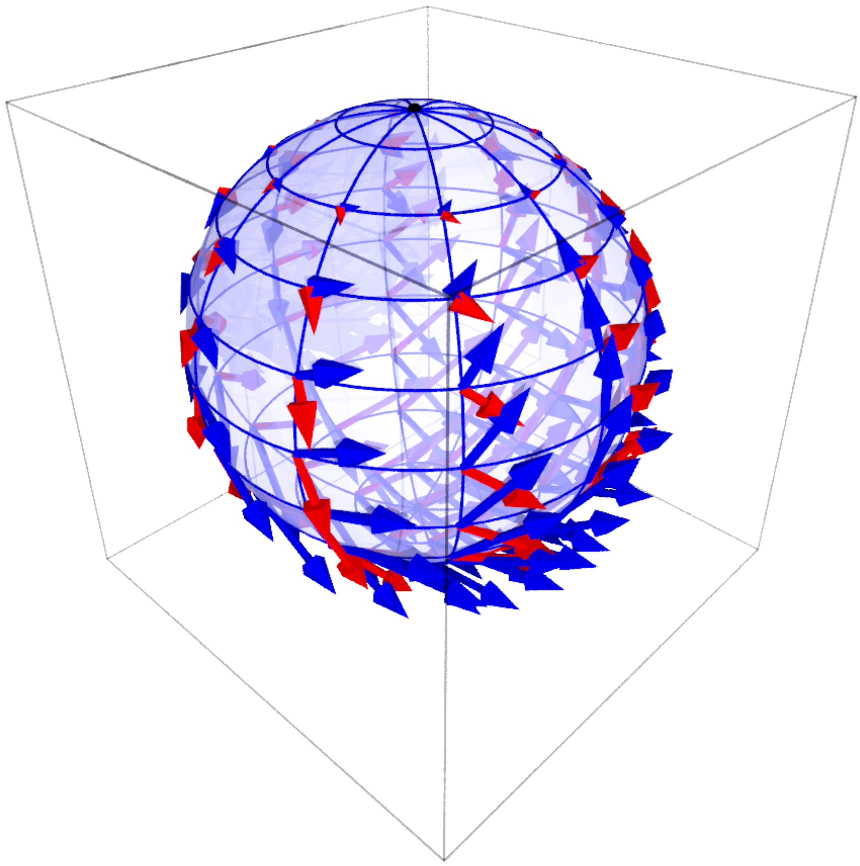
Out[159]: $\frac{\partial}{\partial y}$

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)
show(graph_spher + graph_ey, viewer=viewer3D)
```



We may superpose 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)
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 [162]:
```
vf = v(f)
print(vf)
vf.display()
```

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

Out[162]:

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

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

In [163]:
```
vf(N)
```

Out[163]: 0

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

Out[164]: 4

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

Out[165]: 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 [166]:
```
df = f.differential() ; print(df)
```

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

In [167]:
```
df.display()
```

Out[167]:

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

In [168]:
```
print(df.parent())
```

```
Module /\^1(S^2) of 1-forms on the 2-dimensional differentiable manifol
d S^2
```

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

Out[169]: $\Lambda^1\left(\mathbb{S}^2\right)$

The 1-form acting on a vector field:

In [170]:
```
print(df(v)) ; df(v).display()
```

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

Out[170]:

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

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

In [171]:
```
df(v) == v(f)
```

Out[171]: True

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

In [172]:
```
f.lie_derivative(v) == v(f)
```

Out[172]: 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 [173]:
```
R.<t> = RealLine() ; print(R)
```

```
Real number line R
```

In [174]:
```
R.category()
```

Out[174]: $\mathbf{Smooth}_{\mathbf{R}}$

In [175]:
```
dim(R)
```

Out[175]: 1

In [176]:
```
R.atlas()
```

Out[176]: $[(\mathbf{R}, (t))]$

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

In [177]:
```
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 [178]:
```
c.parent()
```

Out[178]: $\mathrm{Hom}\left(\mathbf{R}, \mathbb{S}^2\right)$

In [179]:
```
c.display()
```

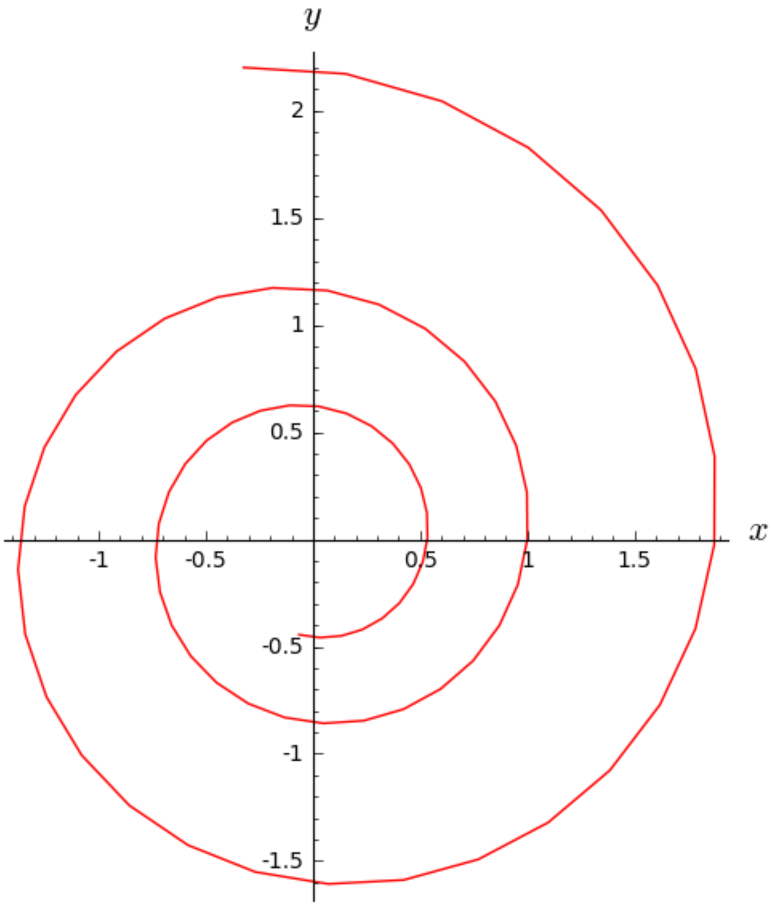
Out[179]:

$$
\begin{array}{llcl}
c: & \mathbf{R} & \longrightarrow & \mathbb{S}^2 \\
 & t & \longmapsto & (x, y) = \left(\cos(t) e^{\left(\frac{1}{10}\,t\right)}, e^{\left(\frac{1}{10}\,t\right)} \sin(t)\right) \\
 & t & \longmapsto & \left(x', y'\right) = \left(\cos(t) e^{\left(-\frac{1}{10}\,t\right)}, e^{\left(-\frac{1}{10}\,t\right)} \sin(t)\right) \\
 & t & \longmapsto & (\theta, \phi) = \left(2 \arctan\left(e^{\left(-\frac{1}{10}\,t\right)}\right), t\right)
\end{array}
$$

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

In [180]:
```
c.plot(chart=stereoN, aspect_ratio=1)
```

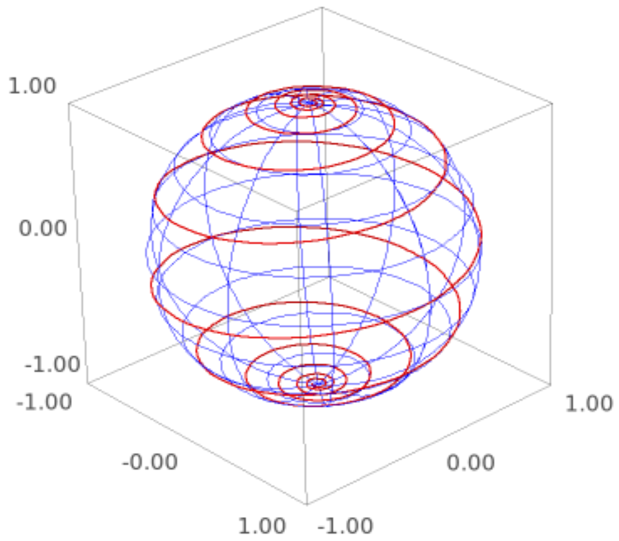
Out[180]:



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

In [181]:
```
graph_c = c.plot(mapping=Phi, max_range=40, plot_points=200, thickness=
2,
                 label_axes=False)
show(graph_spher + graph_c, viewer=viewer3D)
```



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

In [182]:
```
vc = c.tangent_vector_field()
vc
```

Out[182]: c'

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

In [183]:
```
print(vc)
```

```
Vector field c' along the Real number line R with values on the 2-dimen
sional differentiable 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 $\mathcal{X}(\mathbb{R}, c)$; it is a module over the algebra $C^\infty(\mathbb{R})$:

In [184]:
```
vc.parent()
```

Out[184]: $\mathcal{X}(\mathbf{R}, c)$

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

Out[185]: $\mathbf{Modules}_{C^\infty(\mathbf{R})}$

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

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

A coordinate view of c':

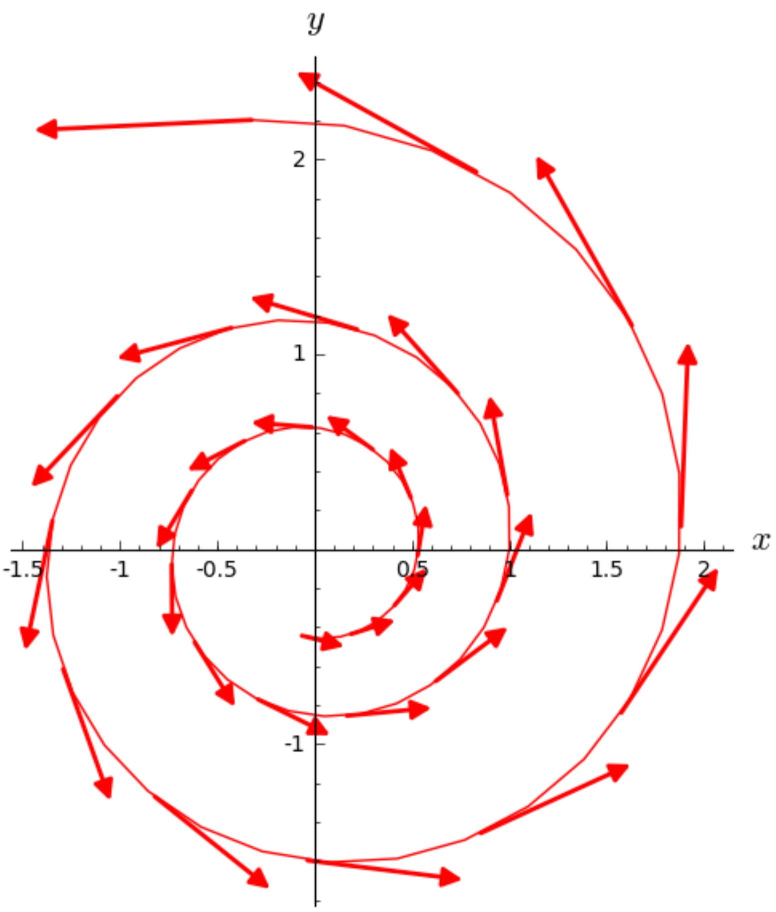
In [187]:
```
vc.display()
```

Out[187]:

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

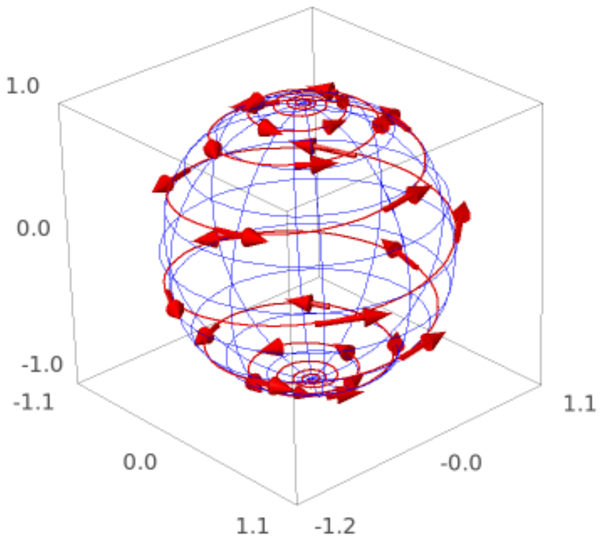
Let us plot the vector field c' in terms of the stereographic chart $(U, (x, y))$:

In [188]:
```
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 Φ:

In [189]:
```
graph_vc = vc.plot(chart=cart, mapping=Phi, ranges={t: (-20, 20)}, numb
er_values=30,
                   scale=0.5, color='red', label_axes=False)
show(graph_spher + graph_c + graph_vc , viewer=viewer3D)
```



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

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

In [190]:
```
h = R3.metric('h')
h[1,1], h[2,2], h[3, 3] = 1, 1, 1
h.display()
```

Out[190]: $h = \mathrm{d}X \otimes \mathrm{d}X + \mathrm{d}Y \otimes \mathrm{d}Y + \mathrm{d}Z \otimes \mathrm{d}Z$

The metric g on $\mathbb{S}^2$ is the pullback of h associated with the embedding Φ:

In [191]:
```
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 [192]:
```
print(g.parent())
```

```
Module T^(0,2)(S^2) of type-(0,2) tensors fields on the 2-dimensional d
ifferentiable manifold S^2
```

In [193]:
```
g.tensor_type()
```

Out[193]: $(0,2)$

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

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

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

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

Out[195]:

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

We may factorize the metric components:

In [196]:
```
g[1,1].factor() ; g[2,2].factor()
```

Out[196]:

$$\frac{4}{\left(x^2+y^2+1\right)^2}$$

In [197]:
```
g.display()
```

Out[197]:

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

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

In [198]:
```
g[:]
```

Out[198]:

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

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

In [199]:
```
g.display(stereoS.frame())
```

Out[199]:

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

In [200]:
```
g.display(spher.frame(), chart=spher)
```
Out[200]:
$$g = \mathrm{d}\theta \otimes \mathrm{d}\theta + \sin(\theta)^2 \mathrm{d}\phi \otimes \mathrm{d}\phi$$

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

In [201]:
```
print(g(v,v))
```
```
Scalar field g(v,v) on the 2-dimensional differentiable manifold S^2
```

In [202]:
```
g(v,v).parent()
```
Out[202]:
$$C^\infty\left(\mathbb{S}^2\right)$$

In [203]:
```
g(v,v).display()
```
Out[203]:
$$\begin{array}{llcl} g(v,v): & \mathbb{S}^2 & \longrightarrow & \mathbb{R} \\ \text{on } U: & (x,y) & \longmapsto & \frac{20}{x^4+y^4+2\,(x^2+1)y^2+2\,x^2+1} \\ \text{on } V: & (x',y') & \longmapsto & \frac{20\,(x'^4+2\,x'^2y'^2+y'^4)}{x'^4+y'^4+2\,(x'^2+1)y'^2+2\,x'^2+1} \\ \text{on } A: & (\theta,\phi) & \longmapsto & 5\,\cos(\theta)^2 - 10\,\cos(\theta) + 5 \end{array}$$

The **Levi-Civitation connection** associated with the metric g:

In [204]:
```
nab = g.connection()
print(nab)
nab
```
```
Levi-Civita connection nabla_g associated with the Riemannian metric g
on the 2-dimensional differentiable manifold S^2
```
Out[204]:
$$\nabla_g$$

As a test, we verify that ∇_g acting on g results in zero:

In [205]:
```
nab(g).display()
```
Out[205]:
$$\nabla_g g = 0$$

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 [206]:
```
g.christoffel_symbols_display(chart=stereoN)
```
Out[206]:
$$\begin{array}{lcl} \Gamma_{ \, x \, x }^{ \, x   } & = & -\frac{2\,x}{x^2+y^2+1} \\ \Gamma_{ \, x \, y }^{ \, x   } & = & -\frac{2\,y}{x^2+y^2+1} \\ \Gamma_{ \, y \, y }^{ \, x   } & = & \frac{2\,x}{x^2+y^2+1} \\ \Gamma_{ \, x \, x }^{ \, y   } & = & \frac{2\,y}{x^2+y^2+1} \\ \Gamma_{ \, x \, y }^{ \, y   } & = & -\frac{2\,x}{x^2+y^2+1} \\ \Gamma_{ \, y \, y }^{ \, y   } & = & -\frac{2\,y}{x^2+y^2+1} \end{array}$$

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

Out[207]:

$$\begin{array}{lcl} \Gamma_{ \, \phi \, \phi }^{ \, \theta   } & = & -\cos\left(\theta\right) \sin\left(\theta\right) \\ \Gamma_{ \, \theta \, \phi }^{ \, \phi   } & = & \frac{\cos\left(\theta\right)}{\sin\left(\theta\right)} \end{array}$$

∇_g acting on the vector field v:

In [208]:
```
print(nab(v))
```

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

In [209]:
```
nab(v).display(stereoN.frame())
```

Out[209]:

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

Curvature

The Riemann tensor associated with the metric g:

In [210]:
```
Riem = g.riemann()
print(Riem)
Riem.display()
```

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

Out[210]:

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

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

In [211]:
```
Riem.display_comp()
```

Out[211]:

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

The components in the frame associated with spherical coordinates:

In [212]:
```
Riem.display_comp(spher.frame(), chart=spher)
```

Out[212]:

$$\begin{array}{lcl} \mathrm{Riem}(g)^{\theta}{}_{\phi\,\theta\,\phi} & = & \sin\left(\theta\right)^{2} \\ \mathrm{Riem}(g)^{\theta}{}_{\phi\,\phi\,\theta} & = & -\sin\left(\theta\right)^{2} \\ \mathrm{Riem}(g)^{\phi}{}_{\theta\,\theta\,\phi} & = & \frac{\cos\left(\theta\right)^{2}-1}{\sin\left(\theta\right)^{2}} \\ \mathrm{Riem}(g)^{\phi}{}_{\theta\,\phi\,\theta} & = & 1 \end{array}$$

In [213]:
```
print(Riem.parent())
```

```
Module T^(1,3)(S^2) of type-(1,3) tensors fields on the 2-dimensional d
ifferentiable manifold S^2
```

In [214]:
```
Riem.symmetries()
```

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

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

In [215]:
```
h.riemann().display()
```

Out[215]: $\mathrm{Riem}\left(h\right) = 0$

The Ricci tensor and the Ricci scalar:

In [216]:
```
Ric = g.ricci()
Ric.display()
```

Out[216]:

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

In [217]:
```
R = g.ricci_scalar()
R.display()
```

Out[217]:

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

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 [218]:
```
delta = S2.tangent_identity_field()
Riem == - R*(g*delta).antisymmetrize(2,3)
```

Out[218]: True

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

In [219]:
```
Ric == (R/2)*g
```

Out[219]: True

The **Levi-Civita tensor** associated with g:

In [220]:
```
eps = g.volume_form()
print(eps)
eps.display()
```

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

Out[220]:

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

In [221]:
```
eps.display(spher.frame(), chart=spher)
```

Out[221]: $\epsilon_g = \sin(\theta)\mathrm{d}\theta \wedge \mathrm{d}\phi$

The exterior derivative of the 2-form ϵ_g:

In [222]:
```
print(eps.exterior_derivative())
```

```
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 [223]:
```
eps.exterior_derivative().display()
```

Out[223]: $\mathrm{d}\epsilon_g = 0$

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 [224]:
```
A.default_chart()
```

Out[224]: $(A, (x, y))$

In [225]:
```
A.default_frame()
```

Out[225]:

$$\left(A, \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right)\right)$$

In [226]:
```
A.set_default_chart(spher)
A.set_default_frame(spher.frame())
A.default_chart()
```

Out[226]: $(A, (\theta, \phi))$

In [227]:
```
A.default_frame()
```

Out[227]: $\left(A, \left(\frac{\partial}{\partial\theta}, \frac{\partial}{\partial\phi}\right)\right)$

We define the new frame e by relating it the coordinate frame $\left(\frac{\partial}{\partial\theta}, \frac{\partial}{\partial\phi}\right)$ via a field of tangent-space automorphisms:

In [228]:
```
a = A.automorphism_field()
a[1,1], a[2,2] = 1, 1/sin(th)
a.display()
```

Out[228]: $\frac{\partial}{\partial\theta} \otimes d\theta + \frac{1}{\sin(\theta)} \frac{\partial}{\partial\phi} \otimes d\phi$

In [229]:
```
a[:]
```

Out[229]: $\begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{\sin(\theta)} \end{pmatrix}$

In [230]:
```
e = spher.frame().new_frame(a, 'e')
print(e) ; e
```

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

Out[230]: $(A, (e_1, e_2))$

In [231]:
```
e[1].display()
```

Out[231]: $e_1 = \frac{\partial}{\partial\theta}$

In [232]:
```
e[2].display()
```

Out[232]: $e_2 = \frac{1}{\sin(\theta)} \frac{\partial}{\partial\phi}$

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

Out[233]:

$$\left[\left(A, \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right)\right), \left(A, \left(\frac{\partial}{\partial x'}, \frac{\partial}{\partial y'}\right)\right), \left(A, \left(\frac{\partial}{\partial\theta}, \frac{\partial}{\partial\phi}\right)\right), \left(A, \left(\frac{\partial}{\partial X}, \frac{\partial}{\partial Y}, \frac{\partial}{\partial Z}\right)\right), (A, (e_1, e_2))\right]$$

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

In [234]:
```
g(e[1],e[1]).expr()
```

Out[234]: 1

In [235]:
```
g(e[1],e[2]).expr()
```
Out[235]: 0

In [236]:
```
g(e[2],e[2]).expr()
```
Out[236]: 1

In [237]:
```
g[e,:]
```
Out[237]:
$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

In [238]:
```
g.display(e)
```
Out[238]: $g = e^1 \otimes e^1 + e^2 \otimes e^2$

In [239]:
```
eps.display(e)
```
Out[239]: $\epsilon_g = e^1 \wedge e^2$

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

In [240]:
```
e.structure_coeff()[:]
```
Out[240]:
$$\left[\left[\left[0,0\right],\left[0,0\right]\right],\left[\left[0,-\frac{\cos\left(\theta\right)}{\sin\left(\theta\right)}\right],\left[\frac{\cos\left(\theta\right)}{\sin\left(\theta\right)},0\right]\right]\right]$$

In [241]:
```
e[2].lie_derivative(e[1]).display(e)
```
Out[241]:
$$-\frac{\cos\left(\theta\right)}{\sin\left(\theta\right)} e_2$$

while we have of course

In [242]:
```
spher.frame().structure_coeff()[:]
```
Out[242]: $\left[\left[\left[0,0\right],\left[0,0\right]\right],\left[\left[0,0\right],\left[0,0\right]\right]\right]$