An de grâce 2017, mercredi 12 juillet 9h:03

Grille yin yang

https://groups.google.com/forum/#!topic/sage-support/NswQgDClEoA (https://groups.google.com/forum/#!topic/sage-support/NswQgDClEoA) (Nils bruin)

Affichage latex et déclaration des variables

In [1]:
```
%display latex
var("r,theta,phi")
#var("r t p")
```

Out[1]: (r, θ, ϕ)

Définition de la fonction coordonnée : Je vais la renommer en plus général, de façon à m'en souvenir.

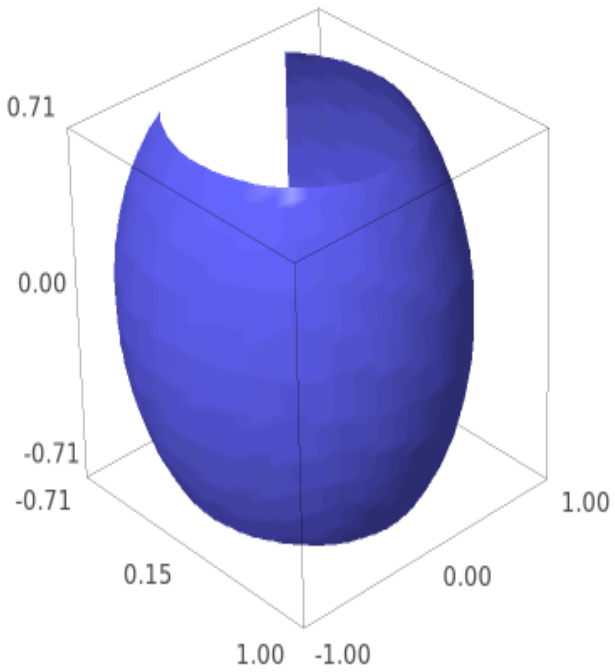
r = r , θ = t , ϕ = p , switch=u , shift =s

In [2]:
```
def coordfunc(r,theta,phi,switch,shift=0):
    if switch=="yin":
        return (r*sin(theta)*cos(phi)+shift,
                r*sin(theta)*sin(phi),r*cos(theta))
    else:
        return (-r*sin(theta)*cos(phi)+shift,
                r*cos(theta),r*sin(theta)*sin(phi))
```

Traçage de la partie yin (horizontale), première couche, bleu si pas d'indication autre.

In [3]:
```
yin=(parametric_plot3d(coordfunc(1,theta,phi,"yin"),
                       (theta,pi/4,3*pi/4),
                       (phi,-3*pi/4,3*pi/4)))
yin
```

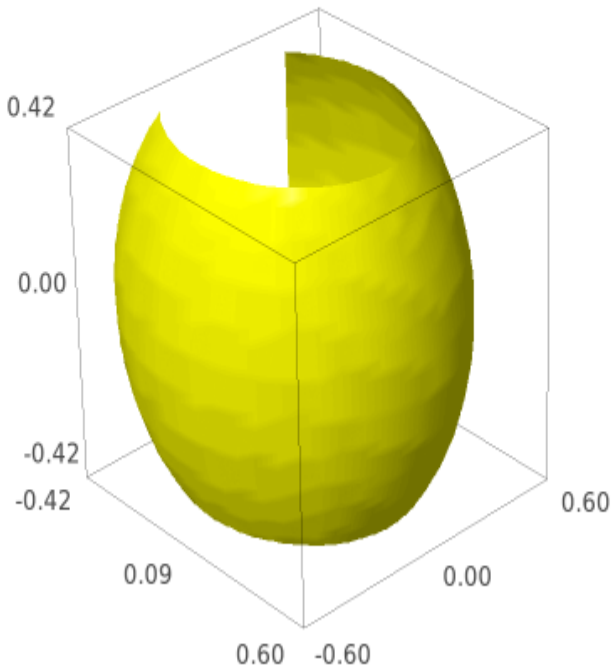
Out[3]:



Traçage du deuxième module, à l'intérieur

In [4]:
```
yin2=parametric_plot3d(coordfunc(0.6,theta,phi,"yin"),
                       (theta,pi/4,3*pi/4),
                       (phi,-3*pi/4,3*pi/4),
                       color="yellow")
yin2
```

Out[4]:

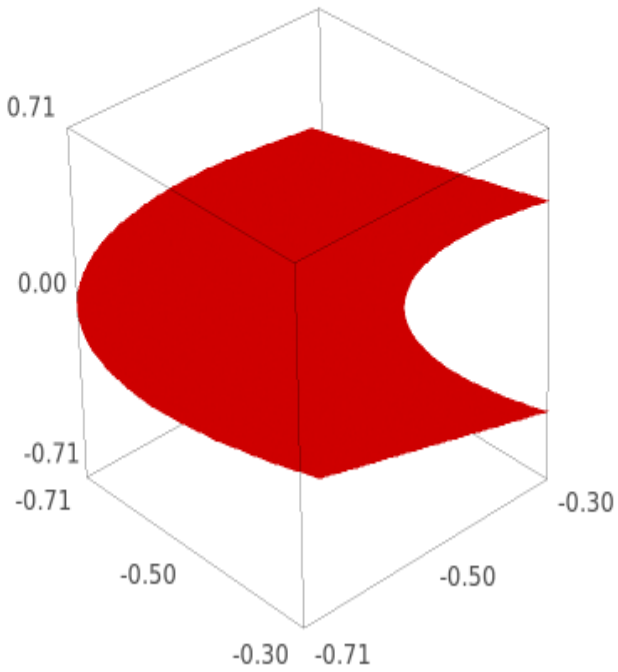


Traçage de yin3,bordure épaisse

Processing math:

In [5]:
```
yin3=parametric_plot3d(coordfunc(r,theta,-3*pi/4,"yin"),
                           (r,0.6,1),
                         (theta,pi/4,3*pi/4),color="red")
yin3
```

Out[5]:

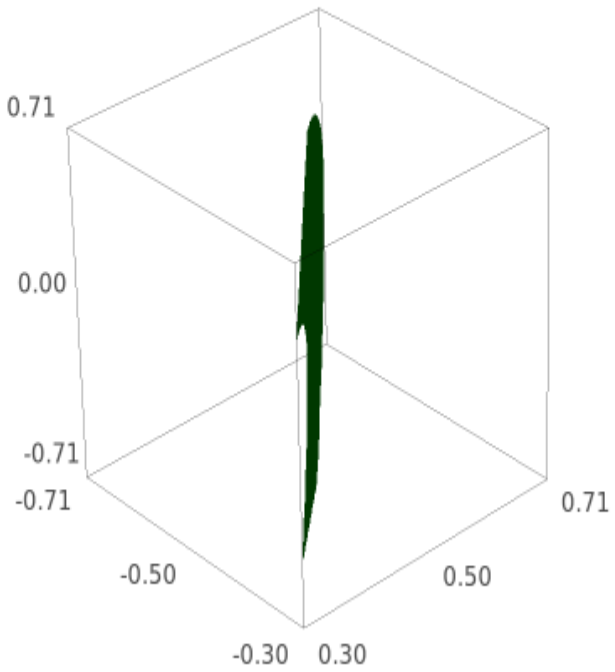


Yin4 pendant positif de yin3

Processing math:

In [6]:
```
yin4=parametric_plot3d(coordfunc(r,theta,3*pi/4,"yin"),
                           (r,0.6,1),(theta,pi/4,3*pi/4),
                       color="green")
yin4
```

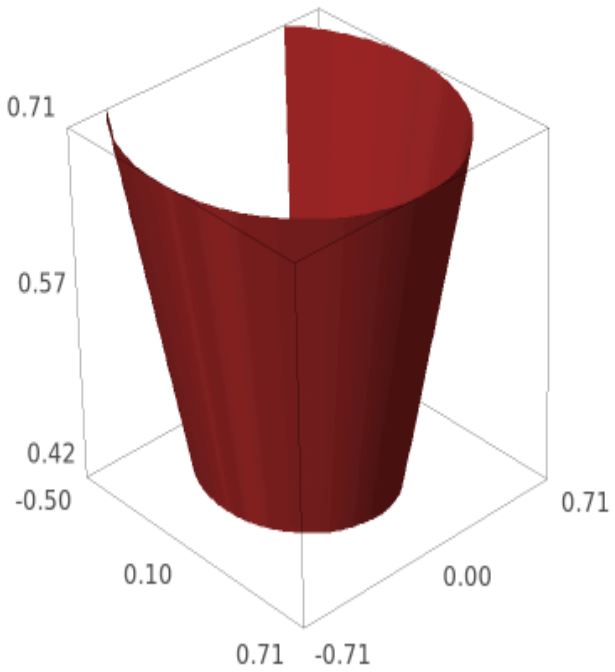
Out[6]:



Yin5, cone à l'intérieur

In [7]:
```
yin5=parametric_plot3d(coordfunc(r,pi/4,phi,"yin"),
                   (r,0.6,1),(phi,-3*pi/4,3*pi/4),
                        color="brown")
yin5
```

Out[7]:



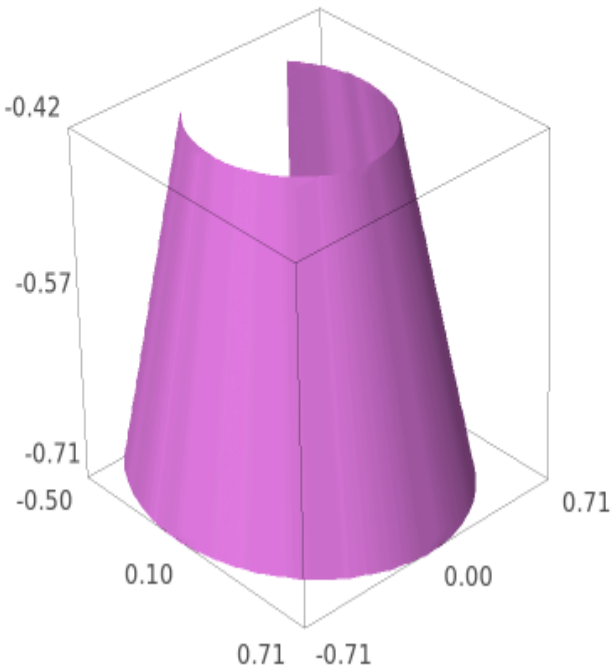
Application loaded

Yin6

Processing math:

In [8]:
```
yin6=parametric_plot3d(coordfunc(r,3*pi/4,phi,"yin"),
                               (r,0.6,1),
                     (phi,-3*pi/4,3*pi/4),color="violet")
yin6
```

Out[8]:



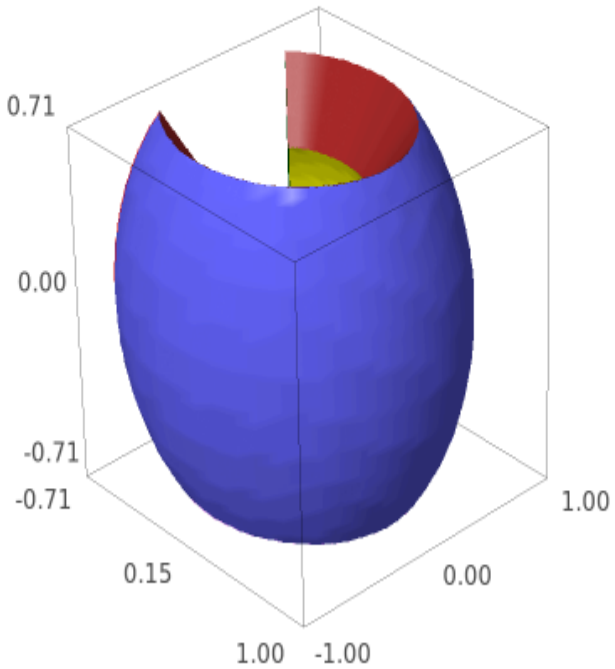
Application loaded.

Assemblage des 6 formes (trigrammes), SAGE ajoute les modules, ce qui en fait une amélioration par rapport à mathematica, beaucoup plus compliqué. Un peu de fioriture et l'emballage sera plus beau : L'élève dépasse le maître (sage>mathematica).

Processing math:

In [9]:
```
a=(yin+yin2+yin3+yin4+yin5+yin6);a
```

Out[9]:



Application loaded.

Maintenant tracer la partie yiang (verticale)

Processing math:

In [10]:
```
yang=(parametric_plot3d(coordfunc(1,theta,phi,"yang"),
                        (theta,pi/4,3*pi/4),
                        (phi,-3*pi/4,3*pi/4),color="cyan"))
yang
```

Out[10]:

Processing math:

In [11]:
```
yang2=parametric_plot3d(coordfunc(0.6,theta,phi,"yang"),
                        (theta,pi/4,3*pi/4),(phi,-3*pi/4,3*pi/4
),
                        color="salmon")
yang2
```

Out[11]:

Processing math:

In [12]:
```
yang3=parametric_plot3d(coordfunc(r,theta,-3*pi/4,"yang"),
                        (r,0.6,1),
                        (theta,pi/4,3*pi/4),color="purple")
yang3
```

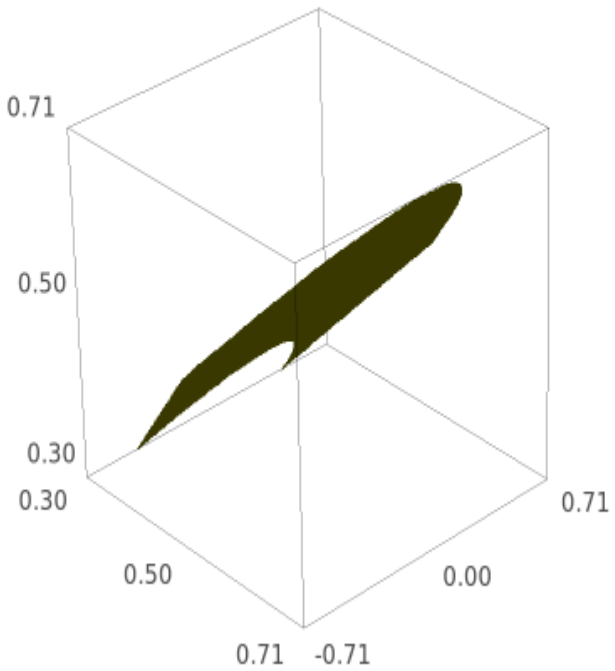
Out[12]:

Application loaded.

Processing math:

In [13]:
```
yang4=parametric_plot3d(coordfunc(r,theta,3*pi/4,"yang"),
                        (r,0.6,1),
                        (theta,pi/4,3*pi/4),
                        color="olive")
yang4
```

Out[13]:

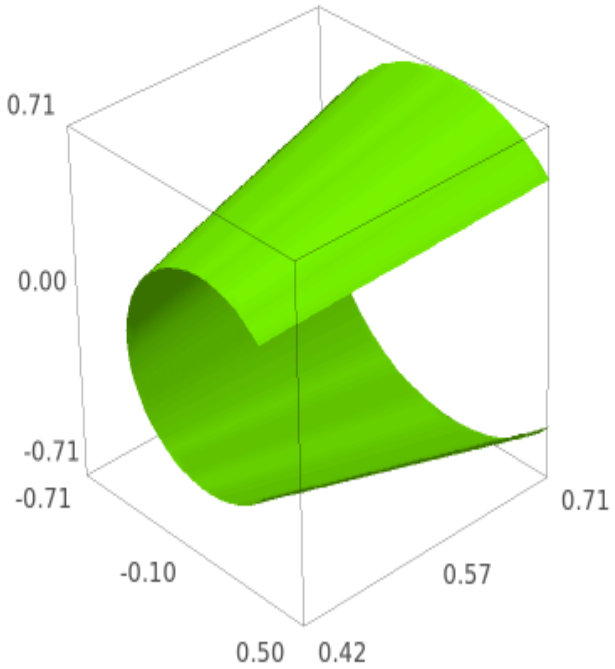


Application loaded.

Processing math:

In [14]:
```
yang5=parametric_plot3d(coordfunc(r,pi/4,phi,"yang"),
                        (r,0.6,1),
                        (phi,-3*pi/4,3*pi/4),
                       color="chartreuse")
yang5
```

Out[14]:



Application loaded.

Processing math:

In [15]:
```
yang6=parametric_plot3d(coordfunc(r,3*pi/4,phi,"yang"),
                        (r,0.6,1),
                           (phi,-3*pi/4,3*pi/4),
                          color="mistyrose")
yang6
```

Out[15]:

Application loaded.

Processing math:

In [16]:
```
b=(yang+yang2+yang3+yang4+yang5+yang6);b
```

Out[16]:

Processing math:

In [23]:
```
#sorted(colors)
t = var('t')
S2 = Manifold(2, 'S^2')
U = S2.open_subset('U')
XS.<th,ph> = U.chart(r'th:(0,pi):\theta ph:(0,2*pi):\phi')
R3 = Manifold(3, 'R^3')
X3.<x,y,z> = R3.chart()
F = S2.diff_map(R3, {(XS, X3): [sin(th)*cos(ph),
                                sin(th)*sin(ph),
                                cos(th)]}, name='F')
c = S2.curve([2*atan(exp(-t/10)), t],
             (t, -oo, +oo), name='c')
graph_c = c.plot(mapping=F, max_range=40,
                 plot_points=200,
                 thickness=2, label_axes=False)
graph_S2 = XS.plot(X3, mapping=F, number_values=50,
                   color='white')
graph_S2+a
```

Out[23]:

In [18]:
```
show(a+b,aspect_ratio=1)
```

Le graphique montre bien l'inclusion horizontal et vertical du yin / yang. Ce qui n'apparait pas au premier abord dans un graphique 2D. Un pseudo maillage ou grid à l'aide de sagemanifolds, inclus dans S2.

In [20]:
```
t = var('t')
S2 = Manifold(2, 'S^2')
U = S2.open_subset('U')
XS.<th,ph> = U.chart(r'th:(0,pi):\theta ph:(0,2*pi):\phi')
R3 = Manifold(3, 'R^3')
X3.<x,y,z> = R3.chart()
F = S2.diff_map(R3, {(XS, X3): [sin(th)*cos(ph),
                                sin(th)*sin(ph),
                                cos(th)]}, name='F')
c = S2.curve([2*atan(exp(-t/10)), t],
             (t, -oo, +oo), name='c')
graph_c = c.plot(mapping=F, max_range=40,
                 plot_points=200,
                 thickness=2, label_axes=False)
graph_S2 = XS.plot(X3, mapping=F, number_values=50,
                   color='black')
graph_S2+yin+yang
```

Out[20]:

In [ ]:
```

```