Galactic Center Transit

September 9, 2016

1 Centro Galactico sobre el cielo de Chacaltaya

```
In [1]: %matplotlib inline
        import ephem
        import numpy as np
        from numpy import sin, cos,tan, arcsin, arctan,pi, arccos,radians, degrees
        import matplotlib as mpl
        from mpl_toolkits.mplot3d import Axes3D
        import matplotlib.pyplot as plt
```

```
In [2]: %%latex
         Si describimos el movimiento de la fuente en la bóveda celeste en coordenadas esfe
         \begin{equation}
          \theta'_{t} =  \arccos(\cos(\delta)\sin( \omega t ))
        \end{equation}
        \begin{equation}
          \phi'_{t} = \arctan\left(\frac{\tan(\delta)}{\cos(\omega t)}\right)
        \end{equation}
        Donde t corresponde al tiempo y $\omega$ a la velocidad de rotación de la Tierra y
```

Si describimos el movimiento de la fuente en la bóveda celeste en coordenadas esféricas tenemos únicamente dos grados de libertad ya que r se considera constante, y entonces su movimiento queda descrito de la siguiente manera:

$$\theta'_t = \arccos(\cos(\delta)\sin(\omega t)) \tag{1}$$

$$\phi'_t = \arctan\left(\frac{\tan(\delta)}{\cos(\omega t)}\right) \tag{2}$$

Donde t corresponde al tiempo y ω a la velocidad de rotación de la Tierra y δ es la DEC a la que se encuentra el CG.

```
In [3]: la=radians(0)

        def phi(t,w):
            W=radians(w) #velocidad angular de rotacion de la tierra
            wt= (W*t)+(pi/2)
            p=arctan((sin(wt))/tan(la))
            Pdegrees=degrees(p)
        #    return Pdegrees
            return p


        def theta(t,w):
```

```
        W=radians(w) #velocidad angular de rotacion de la tierra
        wt= W*t+(pi/2)
        th=arccos(cos(la)*cos(wt))
        THdegrees=degrees(th)
        return th
```

In [22]:
```
%%latex
Este sistema primado será el sistema que corresponde al observador ubicado en el e

 \begin{equation}
  x = x'
\end{equation}

 \begin{equation}
  y = y'\cos\nu+z'\sin\nu
\end{equation}

 \begin{equation}
  z = -y'\sin\nu+ z'\cos\nu
\end{equation}
 Donde $\nu$ son los grados que nos movemos hacia el sur. volviendo a coordenadas

 \begin{equation}
 \label{teta}
\theta_t = \arccos\left(-\sin\nu\,\sin\theta'(t)\,\sin\phi'(t)+\cos\nu\,\cos\theta

\end{equation}

\begin{equation}
\label{fi}
\phi_t = \arctan \left(\cos\nu\,\tan\phi'(t)+ \frac{\sin\nu}{\sin\theta'(t)}  \rig
\end{equation}

El camino recorrido por el Centro galctico en la boveda celeste local de Chacaltaya
```

Este sistema primado será el sistema que corresponde al observador ubicado en el ecuador. Ahora, si nos movemos hacia el sur, hacia donde el CG pasa por el cenit, 29º sur, habría que hacer una rotación del sistema de coordenas de 29º en torno al eje X ya que el eje Y va de norte a sur y nos movemos hacia el sur. Lo que resulta en un sistema no primado tal que:

$$x = x' \tag{3}$$

$$y = y' \cos\nu + z' \sin\nu \tag{4}$$

$$z = -y' \sin\nu + z' \cos\nu \tag{5}$$

Donde ν son los grados que nos movemos hacia el sur. volviendo a coordenadas esféricas locales habiendo desplazado el observador ν grados hacia el Sur las coordenadas que describirían su movimiento por la bóveda celeste serían:

$$\theta_t = \arccos\left(-\sin\nu\,\sin\theta'(t)\,\sin\phi'(t)+\cos\nu\,\cos\theta'(t)\right) \tag{6}$$

$$\phi_t = \arctan \left(\cos\nu\,\tan\phi'(t)+ \frac{\sin\nu}{\sin\theta'(t)} \right) \tag{7}$$

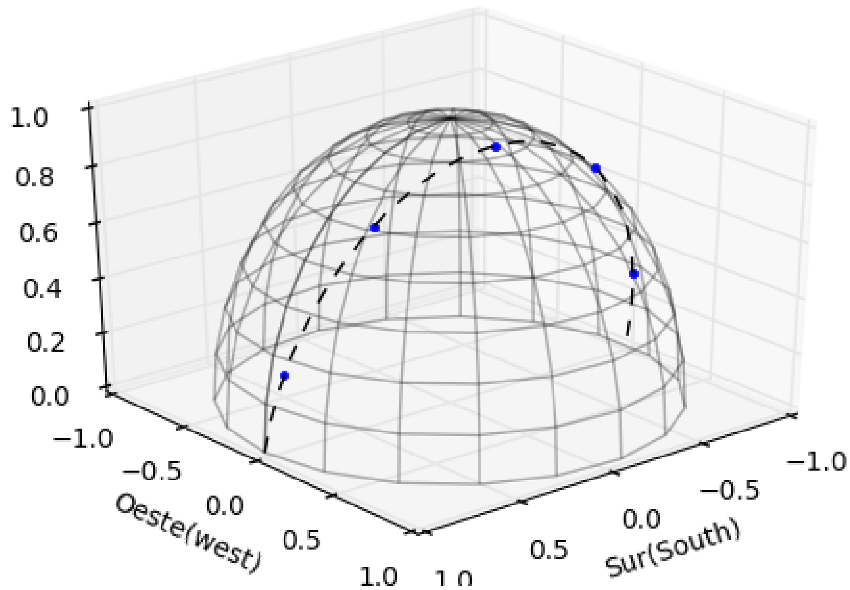
El camino recorrido por el Centro galctico en la boveda celeste local de Chacaltaya se muestra en en la siguiente figura.

In [79]:
```
np.seterr(divide='ignore')
C=cos(radians(0)) #this is equators latitude
S=sin(radians(0))
t=0
lcha=radians(16)
time=np.arange(12./10,12.+(12/10.),12./5.)
t_detail = np.arange(0.,12.5,0.5)
def get_all(t):
    TEdegrees=degrees(arcsin(-S*cos(theta(t,15))*cos(phi(t,15))+C*sin(theta(t,15)
    inside = (cos(theta(t,15))*sin(phi(t,15)))/((C*cos(theta(t,15))*cos(phi(t,15)
    Phi= -degrees(arctan(inside))+180
    xcha=sin(radians(Phi))*sin(radians(TEdegrees))
    ycha=(cos(lcha)*cos(radians(Phi))*sin(radians(TEdegrees)))+(sin(lcha)*cos(radi
    zcha=(-sin(lcha)*cos(radians(Phi))*sin(radians(TEdegrees)))+(cos(lcha)*cos(rac
    return xcha, ycha, zcha
xcha, ycha, zcha = get_all(time)
xi, yi, zi = get_all(t_detail)


theta_final = degrees(arccos(zcha))
phi_2 = degrees(xcha/sin(radians(theta_final)))
phi_final = phi_2[::-1]
```

In [80]:
```
fig = plt.figure()
ax = fig.gca(projection='3d')
u, v = np.mgrid[0:2*pi:20j, 0:pi/2:10j]
x=cos(u)*sin(v)
y=sin(u)*sin(v)
z=cos(v)
ax.plot_wireframe(x, y, z, color="black",alpha=0.3)
ax.azim = 50
ax.plot(xcha,ycha, zcha, 'b.')
ax.plot(xi,yi, zi,'k--')
plt.xlabel("Sur(South)")
plt.ylabel("Oeste(west)")
```

Out[80]:
```
<matplotlib.text.Text at 0x7fa6754e3510>
```




```
In [81]: %%latex
         La tabla presenta 24 puntos en el camino del GC. Representados en coordenadas $\th
```

La tabla presenta 24 puntos en el camino del GC. Representados en coordenadas θ: grados desde el cenit hacia el horizonte. Y ϕ: grados desde la direccion norte en el sentido de las agujas del reloj en el plano.

```
In [82]: class ListTable(list):
             """ Overridden list class which takes a 2-dimensional list of
                 the form [[1,2,3],[4,5,6]], and renders an HTML Table in
                 IPython Notebook. """

             def _repr_html_(self):
                 html = ["<table>"]
                 for row in self:
                     html.append("<tr>")

                     for col in row:
                         html.append("<td>{0}</td>".format(col))

                     html.append("</tr>")
                 html.append("</table>")
                 return ''.join(html)


         import random
         table = ListTable()
         table.append(['theta(deg)', 'phi(deg)'])
         for i in range(0, len(theta_final)):
             table.append([theta_final[i], phi_final[i]+180.])

         table
```

```
Out[82]: [['theta(deg)', 'phi(deg)'],
          [72.719722870319544, 122.93263190739341],
          [38.951622580719082, 126.42986572038294],
          [15.999999999999995, 180.00000000000003],
          [38.951622580719032, 233.57013427961706],
          [72.719722870319472, 237.06736809260659]]

In [ ]:
```