Read brdc file for specific satelite and compute

ephemiris for rest of orbit (96) every 900s (15min)

then check this against the sp3 file for the same

satelite

For this assignment i will be using satelite 12

from NavMsg import ReadNavMessage import matplotlib as mpl import matplotlib.pyplot as plt import math as mt import numpy as np

SatPRN = '12'

with open("NAVandSP3/igs18540.sp3", 'r') as f: Lr1 = for line in f: if line[2:4] == SatPRN: x = float(line[6:18]) y = float(line[18:32]) z = float(line[32:46]) rad1 = np.array([x, y, z]) rad1_sq = np.sqrt((rad1**2).sum()) Lr1 += [rad1_sq] precise = np.array(Lr1) * 1000

def getEK (mk,e): k = 0 while k < 4: if k == 0: ek = mk + e*np.sin(mk) else: ek = mk + e*np.sin(ek) k += 1 return ek

def calcXkYkZk(CurrSat,time):

```
return Xk,Yk,Zk
```

if **name** == "**main**":

```
sqrt = np.sqrt
#step 1: read nav message
Sat = ReadNavMessage(int(SatPRN))
Epochs = []
for i,j in Sat.items():
    Epochs.append(i)
Epochs.sort()
#done reading message

print (Epochs)

for Ep in Epochs:
    if Ep != '18-00-0':
        continue
```

```
lr2 = []
count = []


#step 2 calculate ephemeris
CurrSat = Sat[Ep]
e = CurrSat.e

m = 3.986008*10**14
omegaDot_e = 7.292115167*10**-5

A = CurrSat.sqrtA **2

n0 = sqrt(m/A**3)
n = n0 + CurrSat.DeltaN

time = 0
for num in range(96):
    time = num * 900
    print(CurrSat.Toe)
    tk = time - CurrSat.Toe

    if tk > 302400.0:
        tk = tk - 604800
    if tk < -302400.0:
        tk = tk + 604800

    Mk = CurrSat.M0 + n * tk

    Ek = getEK(Mk,e)

    neum = (sqrt(1-e**2) * np.sin(Ek))/(1-e*np.cos(Ek))
    denom = (np.cos(Ek) - e)/ (1 - e * np.cos(Ek))

    Vk = np.arctan( neum / denom )
    phiK = Vk + CurrSat.omega

    sigUk = CurrSat.Cus * np.sin(2*phiK) + CurrSat.Cuc * np.cos(2*phiK)
    sigRk = CurrSat.Crc * np.sin(2*phiK) + CurrSat.Crs * np.cos(2*phiK)
    sigIk = CurrSat.Cic * np.sin(2*phiK) + CurrSat.Cis * np.cos(2*phiK)

    Uk = phiK + sigUk
    Rk = A * ( 1 - e * np.cos(Ek) ) + sigRk
    Ik = CurrSat.i0 +sigIk + (CurrSat.IDOT)*tk
```

```
    xk_ = Rk * np.cos(Uk)
    yk_ = Rk * np.sin(Uk)

    OHMk = CurrSat.OMEGA + (CurrSat.OMEGADOT - omegaDot_e)*tk - omegaDot_e* CurrSat.Toe
    Xk = xk_ * np.cos(OHMk) - yk_ * np.cos(Ik) * np.sin(OHMk)
    Yk = xk_ * np.sin(OHMk) + yk_ * np.cos(Ik) * np.cos(OHMk)
    Zk = yk_ * np.sin(Ik)

    rad2 = np.array([Xk,Yk,Zk])
    # print(Xk,Yk,Zk)
    rad2_sqr = sqrt((rad2**2).sum())
    lr2 += [rad2_sqr]

    broadcast = np.array(lr2)


count += [CurrSat.Epoch]
length = len(count)
llr1 = length * Lr1
vllr1 = np.array(lr2)

r = precise - broadcast

counter = []
for c in range(1,97):
    counter += [c]
t = r[:96]

fig, ax = plt.subplots()
plt.plot(counter, t, '-', color='black')
# plt.ylim([-100, 100])
plt.title("Difference")
plt.plot()
plt.show()
```