func

March 25, 2020

[1]:

```
import swisseph as swe
#import math as m
from math import ceil, modf, floor
from datetime import datetime

def astottariDasha(lon):
    gName =␣
 ↪["Sukra","Rabi","Chandra","Mangal","Budha","Sani","Brihaspati","Rahu"]
    nakshatraID = ceil(lon / 13.33)
    cName = ''
    dElapsed, dRem = 0.0, 0.0
    if 333.333333334 <= lon or lon < 26.666666667:
        if 333.333333334 <= lon:
            dElapsed = ((lon - 333.33333334) * 21) / 53.333333 # Sukra
            dRem = 21 - (((lon - 333.33333334) * 21) / 53.333333) # Sukra
        else:
            dElapsed = ((lon + 26.66666667) * 21) / 53.33333
            dRem = 21 - (((lon + 26.66666667) * 21) / 53.33333)
    elif 26.66666667 <= lon < 66.66666667:
        dElapsed = ((lon - 26.66666667) * 6) / 40 # Rabi
        dRem = 6 - (((lon - 26.66666667) * 6) / 40) # Rabi
    elif 66.66666667 <= lon < 120.00000001:
        dElapsed = ((lon - 66.66666667) * 15) / 53.33333 # Chandra
        dRem = 15 - (((lon - 66.66666667) * 15) / 53.33333) # Chandra
    elif 120.00000001 <= lon < 160.00000001:
        dElapsed = ((lon - 120.00000001) * 8) / 40 # Mangal
        dRem = 8 - (((lon - 120.00000001) * 8) / 40) # Mangal
    elif 160.00000001 <= lon < 213.33333334:
        dElapsed = ((lon - 160.00000001) * 17) / 53.33333 # Budha
        dRem = 17 - (((lon - 160.00000001) * 17) / 53.33333) # Budha
    elif 213.33333334 <= lon < 253.33333334:
        dElapsed = ((lon - 213.33333334) * 10) / 40 # Sani
        dRem = 10 - (((lon - 213.33333334) * 10) / 40) # Sani
    elif 253.33333334 <= lon < 293.33333334:
        dElapsed = ((lon - 253.33333334) * 19) / 40 # Brihaspati
        dRem = 19 - (((lon - 253.33333334) * 19) / 40) # Brihaspati
    elif 293.33333334 <= lon < 333.33333334:
        dElapsed = ((lon - 293.33333334) * 12) / 40 # Rahu
        dRem = 12 - (((lon - 293.33333334) * 12) / 40) # Rahu

    if nakshatraID  in [1,2,26,27]:
        cName = gName[0]
    elif nakshatraID in [3,4,5]:
        cName = gName[1]
    elif nakshatraID in [6,7,8,9]:
        cName = gName[2]
    elif nakshatraID in [10,11,12]:
        cName = gName[3]
    elif nakshatraID in [13,14,15,16]:
        cName = gName[4]
    elif nakshatraID in [17,18,19]:
        cName = gName[5]
    elif nakshatraID in [20,21,22]:
        cName = gName[6]
    elif nakshatraID in [23,24,25]:
        cName =  gName[7]
    return [dElapsed,dRem, cName]

def nakshatraName(nakshatraID):
    nName = [
    "Aswini","Bharani","Krittika","Rohini","Mrigashira","Ardra", \
    "Punarvasu","Pushya","Aslesha","Makha","Purva Phalguni","Uttara Phalguni", \
    "Hasta","Chitra","Swati","Visakha","Anuradha","Jyestha","Moola","Purva Asadha", \
    "Uttara Asadha","Sravana","Dhanistha","Satabhisaj","Purva Bhadrapada", \
    "Uttara Bhadrapada","Revati"]
    return nName[nakshatraID - 1]

def vimshottariDasha( lunarLongitude):
    gName = ["Ketu","Sukra","Rabi","Chandra","Mangal","Rahu","Brihaspati","Sani","Budha"]
    vimshottariSpan = [17,7,20,6,10,7,18,16,19]
    dElapsed, dRem = 0.0, 0.0
    nakshatraID = ceil(lunarLongitude / 13.33)
    if 0 <= lunarLongitude <360:
        N = nakshatraID
        dElapsed = (lunarLongitude - (N - 1) * 13.33333) * vimshottariSpan[N % 9] / 13.33333
        dRem = vimshottariSpan[N % 9] - (lunarLongitude - (N - 1) * 13.33333) * vimshottariSpan[N % 9] / 13.33333
    return [dElapsed, dRem, gName[(nakshatraID - 1) % 9]]

def yoginiDasha(lunarLongitude):
    yoginiSpan = [5,6,7,8,1,2,3,4,5]
    gName =␣
 ↪["Ulka","Siddha","Sangkata","Mangala","Pingala","Dhanya","Bhramari","Bhadrika"]
    dElapsed, dRem = 0.0, 0.0
    nakshatraID = ceil(lunarLongitude / 13.33)
    if 0 <= lunarLongitude <360:
        N = nakshatraID
        dElapsed = (( lunarLongitude - (N - 1) * 13.33333)) * yoginiSpan[(N %␣
 ↪8)] / 13.33333
        dRem = yoginiSpan[N % 8] - ((( lunarLongitude - (N - 1) * 13.33333)) *␣
 ↪yoginiSpan[N % 8] / 13.33333)
    return [dElapsed, dRem, gName[(nakshatraID - 1) % 8]]

def astottariMahaDasha(birthDateTime, currentDateTime, astottariDasha,␣
 ↪astottariDashaGraha):
    gName=["Sukra","Rabi","Chandra","Mangal","Budha","Sani","Brihaspati","Rahu"]
    astottariSpan = [21,6,15,8,17,10,19,12]
    bhukta, bhogya = 0.0, 0.0
    dashaID, flag = 0, 0
    span = currentDateTime - birthDateTime
    currentAge = span / 360
    index = gName.index(astottariDashaGraha)
    while currentAge > 0:
        if flag == 0:
            bhukta = astottariDasha + currentAge
            bhogya = astottariSpan[index] - (astottariDasha + currentAge)
            currentAge = currentAge - (astottariSpan[index] - astottariDasha)
            dashaID = index
            index = (index + 1) % 8
            flag = 1
        else:
            bhukta = currentAge
            bhogya = astottariSpan[index] - currentAge
            currentAge = currentAge - astottariSpan[index]
            dashaID = index
            index = (index + 1) % 8
    return  [dashaID, gName[dashaID], bhukta, bhogya]

def astottariAntarDasha(dashaID, bhukta):
    gName = ["Sukra", "Rabi", "Chandra", "Mangal", "Budha", "Sani",␣
 ↪"Brihaspati", "Rahu"]
    astottariSpan = [21,6,15,8,17,10,19,12]
    flag = 0
    index = dashaID
    mahaDasha = dashaID
    currentAge = bhukta
    while currentAge > 0:
        if flag == 0:
            bhogya = (astottariSpan[mahaDasha] * astottariSpan[index] / 108) -␣
 ↪currentAge
            currentAge = currentAge - (astottariSpan[mahaDasha] *␣
 ↪astottariSpan[index] / 108)
            dashaID = index
            index = (index + 1) % 8
            flag = 1
        else:
            bhukta = currentAge
            bhogya = (astottariSpan[mahaDasha] * astottariSpan[index] / 108) -␣
 ↪currentAge
            currentAge = currentAge - (astottariSpan[mahaDasha] *␣
 ↪astottariSpan[index] / 108)
            dashaID = index
            index = (index + 1) % 8
    return [dashaID, gName[dashaID], bhukta, bhogya]

def vimshottariMahaDasha(birthDateTime, currentDateTime, lunarLongitude,␣
 ↪vimDashaBhukta):
    gName = ["Ketu","Sukra", "Rabi", "Chandra", "Mangal", "Rahu", "Brihaspati",␣
 ↪\
            "Sani", "Budha", "Ketu", "Sukra", "Rabi", "Chandra", "Mangal", \
            "Rahu", "Brihaspati", "Sani", "Budha", "Ketu", "Sukra" "Rabi",␣
 ↪"Chandra", \
            "Mangal", "Rahu", "Brihaspati", "Sani", "Budha"]
    vimsottariSpan =␣
 ↪[7,20,6,10,7,18,16,19,17,7,20,6,10,7,18,16,19,17,7,20,6,10,7,18,16,19,17]
    vimSpan = {"Sani": 19,"Budha": 17,"Ketu": 7,"Sukra": 20,"Rabi": 6,"Chandra":
 ↪ 10,"Mangal":7,"Rahu":18,"Brihaspati":16
    }
    flag = 0
    span = currentDateTime - birthDateTime
    currentAge = span / 360
    nakshatraID = ceil(lunarLongitude/13.33)
    index = nakshatraID - 1
    while currentAge > 0:
        if flag == 0:
            bhukta = vimDashaBhukta + currentAge
            bhogya = vimsottariSpan[index] - (vimDashaBhukta + currentAge)
            currentAge = currentAge - (vimsottariSpan[index] - vimDashaBhukta)
            dashaID = index
            index = (index + 1) % 27
            flag = 1
        else:
            bhukta = currentAge
            bhogya = vimsottariSpan[index] - currentAge
            currentAge = currentAge - vimsottariSpan[index]
            dashaID = index
            index = (index + 1) % 27
    return [dashaID, gName[dashaID], bhukta, bhogya]

def vimshottariAntarDasha(dashaID, bhukta):
    gName = ["Ketu", "Sukra", "Rabi", "Chandra", "Mangal", "Rahu",␣
 ↪"Brihaspati", "Sani", "Budha"]
    vimSpan = [7,20,6,10,7,18,16,19,17]
    flag = 0
    index = dashaID % 9
    mahaDasha = dashaID % 9
    currentAge = bhukta
    while currentAge > 0:
        if flag == 0:
            bhogya = (vimSpan[mahaDasha] * vimSpan[index] / 120) - currentAge
            currentAge = currentAge - (vimSpan[mahaDasha] * vimSpan[index] /␣
 ↪120)
            dashaID = index
            index = (index + 1) % 9
            flag = 1
        else:
            bhukta = currentAge
            bhogya = (vimSpan[mahaDasha] * vimSpan[index] / 120) - currentAge
            currentAge = currentAge - (vimSpan[mahaDasha] * vimSpan[index] /␣
 ↪120)
            dashaID = index
            index = (index + 1) % 9
    return [dashaID, gName[dashaID], bhukta, bhogya]


def yoginiMahaDasha(birthDateTime, currentDateTime, lunarLongitude,␣
 ↪yogiDashaBhukta):
    gName = ["Ulka","Siddha","Sangkata", \
           ␣
 ↪"Mangala","Pingala","Dhanya","Bhramari","Bhadrika","Ulka","Siddha","Sangkata",␣
 ↪\
           ␣
 ↪"Mangala","Pingala","Dhanya","Bhramari","Bhadrika","Ulka","Siddha","Sangkata",␣
 ↪\
           ␣
 ↪"Mangala","Pingala","Dhanya","Bhramari","Bhadrika","Ulka","Siddha","Sangkata"]
    yoginiSpan = [6,7,8,1,2,3,4,5,6,7,8,1,2,3,4,5,6,7,8,1,2,3,4,5,6,7,8]
    yogName = {"Mangala": 1,"Pingala": 2,"Dhanya": 3,"Bhramari": 4,"Bhadrika":
 ↪5,"Ulka": 6,"Siddha": 7,"Sangkata": 8
    }
    flag = 0
    span = currentDateTime - birthDateTime
    currentAge = span / 360
    nakshatraID = ceil(lunarLongitude/13.33)
    index = nakshatraID - 1

    while currentAge > 0:
        if flag == 0:
            bhukta = yogiDashaBhukta + currentAge
            bhogya = yoginiSpan[index] - (yogiDashaBhukta + currentAge)
            currentAge = currentAge - (yoginiSpan[index] - yogiDashaBhukta)
            dashaID = index
            index = (index + 1) % 8
            flag = 1
        else:
            bhukta = currentAge
            bhogya = yoginiSpan[index] - currentAge
            currentAge = currentAge - yoginiSpan[index]
            dashaID = index
            index = (index + 1) % 8
    return [dashaID, gName[dashaID], bhukta, bhogya]

def yoginiAntarDasha(dashaID, bhukta):
    gName = ["Ulka", "Siddha", "Sangkata", "Mangala", "Pingla", "Dhanya",␣
 ↪"Bhramari", "Bhadrika"]
    yoginiSpan = [6,7,8,1,2,3,4,5]
    flag = 0
    index = dashaID % 8
    mahaDasha = dashaID % 8
    currentAge = bhukta
    while currentAge > 0:
        if flag == 0:
            bhogya = (yoginiSpan[mahaDasha] * yoginiSpan[index] / 36) -␣
 ↪currentAge
            currentAge = currentAge - (yoginiSpan[mahaDasha] *␣
 ↪yoginiSpan[index] / 36)
            dashaID = index
            index = (index + 1) % 8
            flag = 1
        else:
            bhukta = currentAge
            bhogya = (yoginiSpan[mahaDasha] * yoginiSpan[index] / 36) -␣
 ↪currentAge
            currentAge = currentAge - (yoginiSpan[mahaDasha] *␣
 ↪yoginiSpan[index] / 36)
            dashaID = index
            index = (index + 1) % 8
    return [dashaID, gName[dashaID], bhukta, bhogya]

def yMDD(yr):
    rMN = modf(yr)[0]*12
    rDY = modf(rMN)[0]*30
    rDD = modf(rDY)[0]*60
    return [floor(yr), floor(rMN), floor(rDY), rDD]
```

[ ]: