

```

#-----
#
# 双曲線起動（内向き：侵入系） 計算
#
#           Programmer:H.Endo
#       Create Date:2021/03/09
#
#-----

import math

#
# comFunction : 共通関数
#
class comFunction:
    #
    # A : 動的能量計算(je)
    # <Param>
    #   m : 質量(kg)
    #   v : 速度(km/h)
    #
    def A(self,m,v):
        return m*v**2          # A=mv^2

    #
    # v  :  動的能量->速度  換算(km/h)
    # <Param>
    #   A : 動的能量(je)
    #   m : 質量(kg)
    #
    def v(self,A,m):
        return math.sqrt(A/m)  # v=sqrt(A/m)

    #
    # S  :  静的エネルギー計算(je)
    # <Param>
    #   M : 中心天体質量(kg)
    #   m : 周回天体質量(kg)
    #   r : 2天体間距離(km)
    #   n : 1:通常静的エネルギー  2:脱出静的エネルギー
    #
    def S(self,M,m,r,n):
         $\mu = \text{self.}\mu(M,m)$           #  $E_m = m_p \times c^2$ 

```

```

    Sr = n *  $\mu$  / r          # Sr = Em x ac / r
    return Sr

#
#  $\mu$  : 重力定数
# <Param>
# M : 中心天体質量(kg)
# m : 周回天体質量(kg)
#
def  $\mu$ (self,M,m):
    Em = self.Em(m)          # Em = mp x c^2
    ac = self.a(M,m)         # ac = U x (m + mp)
     $\mu$  = Em * ac             #  $\mu$  = Em x ac
    return  $\mu$ 

#
# Em : 質量エネルギー(je)
# <Param>
# M : 質量(kg)
#
def Em(self,m):
    cc = comConst()
    Em = m * cc.c**2         # Em = mp x c^2
    return Em

#
# a : 光速時基準軌道半径(km)
# <Param>
# M : 質量(kg)
# m : 周回天体質量(kg)
#
def a(self,M,m):
    cc = comConst()
    ac = cc.U * (M + m)      # ac = U x (m + mp)
    return ac                # Em = mp x c^2

#
# hToS : 時間を秒に換算
# <Param>
# h : 時間(h)
#
def hToS(self,h):

```

```

    return h/3600

#
# radToDeg : radian を 度に換算
# <Param>
# rad : 時間(h)
#
def radToDeg(self,rad):
    return rad * 180 /math.pi

#
# 定数変数 クラス
#

class comConst:
    c=299792458/1000*3600 #光速(km/h)
    U=7.42426e-31      #宇宙エネルギー一定数(km/kg)

    M = 1.9891e30      #太陽質量(kg)
    m = 1.89813E+27    #木星質量(kg)
    mp = 1000          #宇宙船質量(kg)
    Rj = 7.9992e8
    Re = 149598262     #太陽一地球間距離
    r = 2.25302e6 + 71492 #木星近点
    Rp = 7.9992e8
    ap = 4.1759e9

    rp = 6378.137 + 260.9 # (近点) (km)
    θ = 1.32542          # 侵入角度(rad) θ =75.941°

def listU(self):
    print("")
    print("-----")
    print("<宇宙用定数>")
    print("-----")
    print("")

    print("光速 c = {0} km/h , 宇宙エネルギー一定数 U = {1} km/kg".format(self.c,self.U))

def listM(self):
    print("")
    print("-----")
    print("<軌道用定数>")

```

```

print("-----")
print("")

print("太陽質量      M = %.5e kg"%(self.M))
print("地球質量      m = %.5e kg"%(self.m))
print("宇宙船質量     mp = %.5e kg"%(self.mp))
print("太陽-地球間距離 Re = %.5e km"%(self.Re))
print("木星近点       r = %.5e km"%(self.r))
print("")

```

```

#=====
# earthToJupiter :   地球から木星
#=====

```

```

class earthToJupiter:

```

```

    #地球から木星軌道

```

```

    Sp = 0    #軌道近点静的エネルギー
    Se = 0    #木星付近静的エネルギー
    Sap = 0   #半交軸近点静的エネルギー
    ve = 0    #軌道近点速度
    vp = 0    #木星到着速度

```

```

    def culcVelocity(self):

```

```

        cf = comFunction()
        cc = comConst()

```

```

        self.Se = cf.S(cc.M,cc.mp,cc.Re,1) #軌道近点静的エネルギー
        self.Sp = cf.S(cc.M,cc.mp,cc.Rp,1) #木星付近静的エネルギー
        self.Sap = cf.S(cc.M,cc.mp,cc.ap,1) #半交軸近点静的エネルギー
        self.Ae = 2 * self.Se + self.Sap  #地球脱出動的エネルギー
        self.Ap = 2 * self.Sp + self.Sap  #木星付近動的エネルギー
        self.ve = cf.v(self.Ae,cc.mp)     #近点速度
        self.vp = cf.v(self.Ap,cc.mp)     #木星到着速度

```

```

        return self.vp

```

```

    def listData(self):

```

```

        cf = comFunction()
        cc = comConst()

```

```

print("")
print("-----")
print("<地球->木星 vp>")
print("-----")
print("")
print("太陽-木星間距離 Rp = %.5e km"%(cc.Rp))
print("半交軸 ap = %.5e km"%(cc.ap))
print("地球脱出速度 ve = %.5e km"%(cf.hToS(self.ve)))
print("木星到着速度 vp = %.5e km"%(cf.hToS(self.vp)))
print("")

```

```

#=====
# Perigee : 宇宙船軌道 木星近点
#=====

```

```

class Perigee:
    #近点
    vHr = 0          # 基準軌道方向速度
    vVr = 0          # 中心天体方向速度
    AHr = 0          # 基準軌道方向動的エネルギー
    AVr = 0          # 中心天体逆方向動的エネルギー
    Ar = 0           # 全動的エネルギー(=2Sr + Sa)
    Sr = 0           # 近点静的エネルギー
    vr = 0
    θr = 0

```

```

#
# culVelocity : 速度計算
#
def culcVelocity(self,Sa):
    cf      = comFunction()
    cc      = comConst()
    self.AHr = cf.S(cc.m,cc.mp,cc.r,2)
    self.AVr = Sa
    self.vHr = cf.v(self.AHr,cc.mp)
    self.vVr = cf.v(self.AVr,cc.mp)
    self.θr  = math.atan(self.vVr/self.vHr)
    self.Ar  = 2 * self.AHr + Sa
    self.vr  = cf.v(self.Ar , cc.mp)

```

```

#
# listData : データリスト

```

```

#
def listData(self):
    cf = comFunction()
    cc = comConst()

    print("")
    print("-----")
    print("<侵入(近点) r>")
    print("-----")
    print("")
    print("距離          r = %.5e km"%(cc.r))
    print("速度          vr = %.5e km/h"%(cf.hToS(self.vr)))
    print("角度          θ = %.5e 度"%(cf.radToDeg(self.θ r)))
    print("速度 (基準軌道) vHr = %.5e km/h"%(cf.hToS(self.vHr)))
    print("速度 (中心天体) vVr = %.5e km/h"%(cf.hToS(self.vVr)))
    print("エネ (基準軌道) AHr = %.5e je"%(self.AHr))
    print("エネ (中心天体) AVr = %.5e je"%(self.AVr))
    print("")

```

```

#=====
# SemilIntersectingAxis : 宇宙船軌道 半交軸
#=====

```

```

class SemilIntersectingAxis:
    a = 0          # 脱出 (近点) (km)
    θ a = 0       # 脱出角度(rad) θ =26.97242°
    va = 0        # 脱出速度(km/h)
    vHa = 0       # 基準軌道方向速度
    vVa = 0       # 中心天体方向速度
    AHa = 0       # 基準軌道方向動的エネルギー
    AVa = 0       # 中心天体逆方向動的エネルギー
    Aa = 0        # 動的エネルギー
    Sa = 0        # 生的エネルギー

```

```

def culcVelocity(self,Sa):
    cf = comFunction()
    cc = comConst()

    self.a = cf.μ (cc.M,cc.m) / Sa
    self.AHa = 2 * Sa
    self.AVa = Sa

```

```

self.vHa = cf.v(self.AHa,cc.mp)
self.vVa = cf.v(self.AVa,cc.mp)
self.θ a = math.atan(self.vVa/self.vHa)
self.Aa = self.AHa + self.AVa
self.va = cf.v(self.Aa,cc.mp)

#
# listData : データリスト
#
def listData(self):
    cf = comFunction()

    print("")
    print("-----")
    print("<侵入(半交軸) a>")
    print("-----")
    print("")
    print("<侵入(半交軸) a>")
    print("距離          a = %.5e km"%(self.a))
    print("速度          va = %.5e km/h"%(cf.hToS(self.va)))
    print("角度          θ a = %.5e 度"%(cf.radToDeg(self.θ a)))
    print("エネ (基準軌道) AHa = %.5e je"%(self.AHa))
    print("エネ (中心天体) AVa = %.5e je"%(self.AVa))
    print("速度 (基準軌道) vHa = %.5e km/h"%(cf.hToS(self.vHa)))
    print("速度 (中心天体) vVa = %.5e km/h"%(cf.hToS(self.vVa)))

```

```

#=====
# AreaOfInfluence : 宇宙船軌道 影響圏侵入
#=====

```

```

class AreaOfInfluence:
    #影響圏
    R = 0      # 脱出 (近点) (km)
    θ R = 0    # 侵入角度(rad) θ =26.97242°
    vR = 0     # 侵入速度(km/h)
    vHR = 0    # 基準軌道方向速度
    vVR = 0    # 中心天体方向速度
    AHR = 0    # 基準軌道方向動的エネルギー
    AVR = 0    # 中心天体逆方向動的エネルギー
    AR = 0     # 動的エネルギー
    Sa = 0     # 静的エネルギー

```

```

def culcVelocity(self,vp):
    cf = comFunction()
    cc = comConst()

    self.vR = vp
    self.vVR = vp * math.sin(cc.θ)
    self.vHR = vp * math.cos(cc.θ)
    self.θR = math.atan(self.vVR/self.vHR)

    self.AVR = cc.mp * self.vVR**2 # 中心天体逆方向動のエネルギー
    self.AHR = cc.mp * self.vHR**2 # 基準軌道方向動のエネルギー
    self.Sa = self.AVR
    μ = cf.μ(cc.m,cc.mp) # 重力定数
    self.R = 2 * μ / (self.AHR) # 影響圏距離(a)
    return self.Sa

```

```

def listData(self):
    cf = comFunction()

    print("")
    print("-----")
    print("<侵入(影響圏半径) R>")
    print("-----")
    print("")
    print("距離          R = %.5e km"%(self.R))
    print("速度          vR = %.5e km/h"%(cf.hToS(self.vR)))
    print("角度          θR = %.5e 度"%(cf.radToDeg(self.θR)))
    print("速度 (基準軌道) vHR = %.5e km/h"%(cf.hToS(self.vHR)))
    print("速度 (中心天体) vVR = %.5e km/h"%(cf.hToS(self.vVR)))
    print("エネ (基準軌道) AHR = %.5e je"%(self.AHR))
    print("エネ (中心天体) AVR = %.5e je"%(self.AVR))
    print("")

```

```

#=====
# Main Routine
#=====

```

```

#
# define Class
#

```



```
cc = comConst()      # 定数クラス
ej = earthToJupiter() # 地球から木星クラス
pg = Perigee()       # 近点クラス
ia = SemilIntersectingAxis() # 半交軸クラス
ai = AreaOfInfluence() # 影響圏クラス

#
# Variable
#

vR = 0               # 影響圏侵入速度
Sa = 0               # 半交軸静的エネルギー

#
# Listing
#

# Common Constant

cc.listU()           # 宇宙用定数
cc.listM()           # 軌道用定数

vR = ej.culcVelocity() # 地球から木星到着時速度計算
ej.listData()         # リスト

Sa = ai.culcVelocity(vR) # 影響圏侵入時静的エネルギー計算
ai.listData()

ia.culcVelocity(Sa)   # 半交軸状態
ia.listData()

pg.culcVelocity(Sa)   # 近点状態
pg.listData()
```