

Practicas Mecánica

Aroa Antón

Diciembre 2022

1 Tiro 1

Determinamos si un número es primo o no:

```
fig=plt.figure()
fig.set_dpi(100)
#fig.set_size_inches(7,6.5)

# Datos a modificar en la simulacion
# Muelles acoplados

m=1.0      # Valor de las masa
g=9.8      # gravedad
omega=np.pi/12.0/3600.0    # velocidad rotacion
radi= 6378000 # radio tierra

tf=75.0    #tiempo de simulacion
#mi=1.0/m   # Inversa de la masa
velin =700.0 # velocidad de disparo


lat= 40.0  #latitud
alz=30.0 # angulo de alzada del tiro
pla=180.0 # angulo plano 0 grados Sur eje X, 90 grados Este eje y, 180 grados Sur 270
#angulo de orientacion del plano
#correalz= -0.05*np.pi/180.0
#correpla= 0.145*np.pi/180

correalz= 0.0 #correcciones en angulo de alzada del tiro
correpla= 0.0

ralat= lat*np.pi/180.0 #angulos despues de rotar el sist de referencia
raalz= alz*np.pi/180.0
rapla= pla*np.pi/180.0

omex= -omega*np.cos(ralat) #componentes de la velocidad angular de la tierra
omez= omega*np.sin(ralat) #la componente y es 0
ome2radx= omega**2*radi*np.cos(ralat)*np.sin(ralat) #termino centripeto velocidad angular
ome2radz= omega**2*radi*np.cos(ralat)**2 #la componente y es 0

#ome2radz= omega**2*radi*np.cos(ralat)**2

print ( ome2radz)

#par=[mi,g,omega]
par=[g,omex,omez,ome2radx,ome2radz] #aceleracion, componentes v angular, componentes
#centripetas

# Definiendo tiro solo gravedad
```

```

def tiro(z,t,par):
    z1,z2,z3,z4,z5,z6=z
    dzdt=[z4,z5,z6,
           ome2radx +2*z5*omez,
           2*(z6*omex-z4*omez),
           ome2radz -2*z5*omex -g]
    return dzdt

def tirog(az,at,par):
    az1,az2,az3,az4,az5,az6=az
    dazdt=[az4,az5,az6, 0, 0, -g]
    return dazdt

# Llamada a odeint que resuelve las ecuaciones de movimiento

nt=25000 #numero de intervalos de tiempo
dt=tf/nt

# Valores iniciales
z1_0= 0.0 # x punto disparo
z2_0=0.0 # y punto disparo
z3_0=0.0 # z punto disparo
z4_0= velin*np.cos(raalz+correalz)*np.cos(rapla+correpla) # Velocidad x
z5_0= velin*np.cos(raalz+correalz)*np.sin(rapla+correpla) # Velocidad y
z6_0= velin*np.sin(raalz+correalz) # velocidad z

z0=[z1_0,z2_0,z3_0,z4_0,z5_0,z6_0] #Valores iniciales

az1_0= 0.0 # x punto disparo
az2_0=0.0 # y punto disparo
az3_0=0.0 # z punto disparo
az4_0= velin*np.cos(raalz)*np.cos(rapla) # Velocidad x
az5_0= velin*np.cos(raalz)*np.sin(rapla) # Velocidad y
az6_0= velin*np.sin(raalz) # velocidad z

az0=[az1_0,az2_0,az3_0,az4_0,az5_0,az6_0] #Valores inici

t=np.linspace(0,tf,nt)
at=np.linspace(0,tf,nt)
abserr = 1.0e-8
relerr = 1.0e-6

z=odeint(tiro,z0,t,args=(par,),atol=abserr, rtol=relerr)

az=odeint(tirog,az0,at,args=(par,),atol=abserr, rtol=relerr)

plt.close('all')

for i in range (0,nt):
    if z[i-1,2]*z[i,2] <0 : nfinal=i

print (' Tiro contando rotacion')
print ('n=',nfinal)
print (' xf= ', z[nfinal,0], ' yf= ', z[nfinal,1], ' zf= ', z[nfinal,2])

for i in range (0,nt):
    if az[i-1,2]*az[i,2] <0 : nafinal=i

print (' ')
print ('Tiro puro')
print ('na=',nafinal)
print ('xaf= ', az[nafinal,0], ' yaf= ', az[nafinal,1], ' zaf= ', az[nafinal,2])

```

```

dist =sqrt( (z[nfinal,0] -az[nafinal,0])**2 +(z[nfinal,1] -az[nafinal,1])**2 )

print ('Error en el tiro =', dist)

# Definicion del grafico
# Aqui se grafica la evolucion de los desplazamientos x1 y x2 con el tiempo

#fig, ax1 = plt.subplots()
#ax2 = ax1.twinx()
#ax1.set_xlabel('time (s)')
#ax1.set_ylabel('x1(m)', color='b', fontsize=15)
#ax2.set_ylabel('x2(m)', color='r', fontsize=15)
#ax1.tick_params('y', colors='b')
#ax2.tick_params('y', colors='r')
#ax1.set_xlim(xmin=0.,xmax=60.0) #limites del eje x
#line1, = ax1.plot(t[:,z[:,0]], linewidth=2, color='b')
#line2, = ax2.plot(t[:,z[:,1]], linewidth=2, color='r')

# Posicion

ex=z[:,0]/2000
ey=z[:,1]/20000
ez=z[:,2]/20000

# Trayectoria

fig, ax11 = plt.subplots(2,2)
ax11[0,0].set_xlabel('time (s)')
ax11[0,0].set_ylabel('x(m)', color='b', fontsize=12)
line1, = ax11[0, 0].plot( t[:,z[:,0]] , color='b')
ax11[1,0].set_xlabel('time (s)')
ax11[1,0].set_ylabel('y(m)', color='b', fontsize=12)
line2, = ax11[1, 0].plot( t[:,z[:,1]] , color='r')
ax11[0,1].set_xlabel('time (s)')
ax11[0,1].set_ylabel('z(m)', color='b', fontsize=12)
line3, = ax11[0, 1].plot( t[:,z[:,2]] , color='g')
ax11[1,1].set_xlabel('x (s)')
ax11[1,1].set_ylabel('y( ', color='b', fontsize=12)
line4, = ax11[1, 1].plot( z[:,0],z[:,1]**2, color='y')
plt.show()

#Trayectoria tiro puro

fig, ax41 = plt.subplots(2,2)
ax41[0,0].set_xlabel('time (s)')
ax41[0,0].set_ylabel('x(m)', color='b', fontsize=12)
line1, = ax41[0, 0].plot( at[:,az[:,0]] , color='b')
ax41[1,0].set_xlabel('time (s)')
ax41[1,0].set_ylabel('y(m)', color='b', fontsize=12)
line2, = ax41[1, 0].plot( at[:,az[:,1]] , color='r')
ax41[0,1].set_xlabel('time (s)')
ax41[0,1].set_ylabel('z(m)', color='b', fontsize=12)
line3, = ax41[0, 1].plot( at[:,az[:,2]] , color='g')
ax41[1,1].set_xlabel('x (s)')
ax41[1,1].set_ylabel('y( ', color='b', fontsize=12)
line4, = ax41[1, 1].plot( az[:,0], az[:,1]**2, color='y')
plt.show()

# Diferencias

fig, ax81 = plt.subplots(2,2)
ax81[0,0].set_xlabel('time (s)')
ax81[0,0].set_ylabel('dx(m)', color='b', fontsize=12)
line1, = ax81[0, 0].plot( at[:,z[:,0]] -az[:,0] , color='b')

```

```

ax81[1,0].set_xlabel('time (s)')
ax81[1,0].set_ylabel('dy(m)', color='b', fontsize=12)
line2, = ax81[1, 0].plot( at[:,z[:,1]-az[:,1] , color='r')
ax81[0,1].set_xlabel('time (s)')
ax81[0,1].set_ylabel('dz(m)', color='b', fontsize=12)
line3, = ax81[0, 1].plot( at[:,z[:,2]-az[:,2] , color='g')
ax81[1,1].set_xlabel('x (s)')
ax81[1,1].set_ylabel('y( ', color='b', fontsize=12)
line4, = ax81[1, 1].plot( az[:,0], az[:,1]**2, color='y')
plt.show()

# Velocidades

fig, ax51 = plt.subplots(2,2)
ax51[0,0].set_xlabel('time (s)')
ax51[0,0].set_ylabel('Vx(m)', color='b', fontsize=12)
line1, = ax51[0, 0].plot( t[:,z[:,3] , color='b')
ax51[1,0].set_xlabel('time (s)')
ax51[1,0].set_ylabel('Vy(m)', color='b', fontsize=12)
line2, = ax51[1, 0].plot( t[:,z[:,4] , color='r')
ax51[0,1].set_xlabel('time (s)')
ax51[0,1].set_ylabel('Vz(m)', color='b', fontsize=12)
line3, = ax51[0, 1].plot( t[:,z[:,5] , color='g')

fig, ax61 = plt.subplots(2,2)
ax61[0,0].set_xlabel('time (s)')
ax61[0,0].set_ylabel('x(m)', color='b', fontsize=12)
line1, = ax61[0, 0].plot( t[:, ex , color='b')
ax61[1,0].set_xlabel('time (s)')
ax61[1,0].set_ylabel('y(m)', color='b', fontsize=12)
line2, = ax61[1, 0].plot( t[:, ey , color='r')
ax61[0,1].set_xlabel('time (s)')
ax61[0,1].set_ylabel('z(m)', color='b', fontsize=12)
line3, = ax61[0, 1].plot( t[:, ez , color='g')

#ex=z[:,0]/20000
#ey=z[:,1]/20000
#ez=z[:,2]/20000

aex=az[:,0]/20000
aey=az[:,1]/20000
aez=az[:,2]/20000

#fig= plt.figure()
#ax = Axes3D(fig)
#ax.scatter (ex,ey,ez, color='b')
#ax.scatter (aex,aey,aez, color='r')

#
fig= plt.figure()
ax= fig.gca(projection='3d')
ax.view_init(20,-40)
ax.plot3D(ex,ey,ez)

fig= plt.figure()
ax= fig.gca(projection='3d')
ax.view_init(20,-40)
ax.plot3D(aex,aey,aez)
#ax.plot3D(ex,ey,ez)

```

2 Tiro 2

```
import numpy as np
import math
import random
import matplotlib.pyplot as plt
import matplotlib.patches as patches
import matplotlib.mlab as mlab
from scipy.integrate import odeint
from scipy import signal
import matplotlib.animation as animation
from matplotlib.pylab import *
from mpl_toolkits.axes_grid1 import host_subplot
from mpl_toolkits.mplot3d import Axes3D

#fig=plt.figure()
#fig.set_dpi(100)
#fig.set_size_inches(7,6.5)

# Datos a modificar en la simulacion

g=9.8          # gravedad
omega=np.pi/12.0/3600.0      # velocidad rotacion
#omega=0.0
radi= 6378000  # radio tierra

tf=80.0        #tiempo de simulacion
velin =700.0   # velocidad de disparo

lat= -40.0     #latitud
alz=30.0       # angulo de alzada del tiro
# pla=        # angulo plano 0 grados Sur eje X, 90 grados Este eje y, 180 grados Sur 270 grados
#este

ralat= lat*np.pi/180.0
raalz= alz*np.pi/180.0
#rapla= pla*np.pi/180.0

omex= -omega*np.cos(ralat)
omez= omega*np.sin(ralat)
ome2radx= omega**2*radi*np.cos(ralat)*np.sin(ralat)
#ome2radz= omega**2*omega*radi*np.cos(ralat)**2
ome2radz= omega**2*radi*np.cos(ralat)**2
#ome2radz=0.1

print (ome2radz)

#par=[mi,g,omega]
par=[g,omex,omez,ome2radx,ome2radz]

# Definiendo tiro solo gravedad

def tiro(z,t,par):
    z1,z2,z3,z4,z5,z6=z
    dzdt=[z4,z5,z6,
          ome2radx +2*z5*omez,
          2*(z6*omex-z4*omez),
          ome2radz -2*z5*omex -g]
    return dzdt
```

```

def tirog(az,at,par):
    az1,az2,az3,az4,az5,az6=az
    dazdt=[az4,az5,az6, 0, 0, -g]
    return dazdt

# Llamada a odeint que resuelve las ecuaciones de movimiento

nt=20000 #numero de intervalos de tiempo
dt=tf/nt

rapla=np.linspace(0.0,np.pi,360)
dist=np.linspace(0.0,np.pi,360)

for j in range (0,360):
    # Valores iniciales
    rapla[j] = 2*j*np.pi/360
    z1_0= 0.0 # x punto disparo
    z2_0=0.0 # y punto disparo
    z3_0=0.0 # z punto disparo
    z4_0= velin*np.cos(raalz)*np.cos(rapla[j]) # Velocidad x
    z5_0= velin*np.cos(raalz)*np.sin(rapla[j]) # Velocidad y
    z6_0= velin*np.sin(raalz) # velocidad z
    #
    z0=[z1_0,z2_0,z3_0,z4_0,z5_0,z6_0] #Valores iniciales
    #
    az1_0= 0.0 # x punto disparo
    az2_0=0.0 # y punto disparo
    az3_0=0.0 # z punto disparo
    az4_0= velin*np.cos(raalz)*np.cos(rapla[j]) # Velocidad x
    az5_0= velin*np.cos(raalz)*np.sin(rapla[j]) # Velocidad y
    az6_0= velin*np.sin(raalz) # velocidad z
    #
    az0=[az1_0,az2_0,az3_0,az4_0,az5_0,az6_0] #Valores inici
    #
    t=np.linspace(0,tf,nt)
    at=np.linspace(0,tf,nt)
    abserr = 1.0e-8
    relerr = 1.0e-6
    #
    z=odeint(tiro,z0,t,args=(par,),atol=abserr, rtol=relerr)
    #
    az=odeint(tirog,az0,at,args=(par,),atol=abserr, rtol=relerr)
    #
    for i in range (0,nt):
        if z[i-1,2]*z[i,2] <0 : nfinal=i
    for k in range (0,nt):
        if az[k-1,2]*az[k,2] <0 : nafinal=k
    print (j, rapla[j])
    dxy =np.sqrt((z[nfinal,0]-az[nafinal,0])**2+(z[nfinal,1] -az[nafinal,1])**2 )
    dist[j] = dxy
    print ('Error en el tiro =', dxy)

# Definicion del grafico
# Aqui se grafica la evolucion de los desplazamientos x1 y x2 con el tiempo

fig, ax1 = plt.subplots()
#ax2 = ax1.twinx()
ax1.set_xlabel('Angulo')
ax1.set_ylabel('error', color='b', fontsize=15)
ax1.tick_params('y', colors='b')
line1, = ax1.plot(rapla[:], dist[:], linewidth=2, color='b')

```