Runge Kutta

February 2, 2020

[3]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import math
```

[93]:

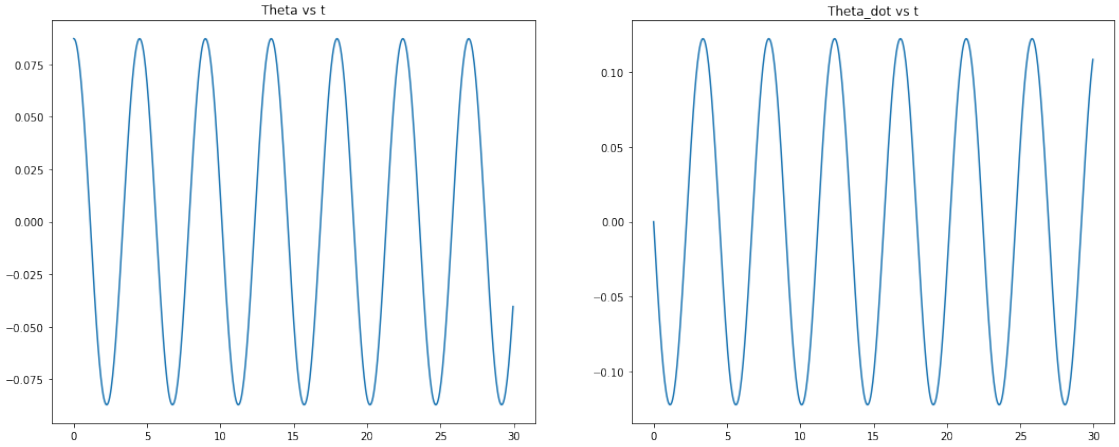
```
def Ftheta(t, theta,theta_dot):
    return theta_dot
def Ftheta_dot(t, theta,theta_dot):
    return g/l*(-1*theta)
def Ftheta_inv(t, theta,theta_dot):
    return g/l*math.sin(theta)
```

0.0.1 For simple Pendulum,

[94]:

```
l=5;m=2;g=9.8
h=0.05 #step-size
tf=30 #final time
t=[0]
th=[0.0873] #Theta
th_d=[0] #Theta_dot
# Count number of iterations using step size h
n = (int)(tf/h)
# Iterate for number of iterations
t0=0
for i in range(1,n):
    "Apply Runge Kutta Formulas to find next value of y"
    k1 = Ftheta(t0,th[-1],th_d[-1])
    k1_d = Ftheta_dot(0,th[-1],th_d[-1])
    k2 = Ftheta(t0+0.5*h, th[-1]+0.5*h*k1,th_d[-1]+0.5*h*k1_d)
    k2_d = Ftheta_dot(t0+0.5*h, th[-1]+0.5*h*k1,th_d[-1]+0.5*h*k1_d)
    k3 = Ftheta(t0+0.5*h, th[-1]+0.5*h*k2,th_d[-1]+0.5*h*k2_d)
    k3_d = Ftheta_dot(t0+0.5*h, th[-1]+0.5*h*k2,th_d[-1]+0.5*h*k2_d)
    k4 = Ftheta(t0+h, th[-1]+h*k3,th_d[-1]+h*k3_d)
    k4_d = Ftheta_dot(t0+h, th[-1]+h*k3,th_d[-1]+h*k3_d)

    # Update next value of theta
    th.append(th[-1] + (1/6)*(k1 + 2 * k2 + 2 * k3 + k4)*h)
    th_d.append(th_d[-1] + (1/6)*(k1_d+2*k2_d+2*k3_d+k4_d)*h)
    # Update next value of t
    t0 = t0 + h
    t.append(t0)
fig,a =  plt.subplots(1,2,figsize=(18,7))
a[0].plot(t,th)
a[0].set_title('Theta vs t')
a[1].plot(t,th_d)
a[1].set_title('Theta_dot vs t')
plt.show()
```

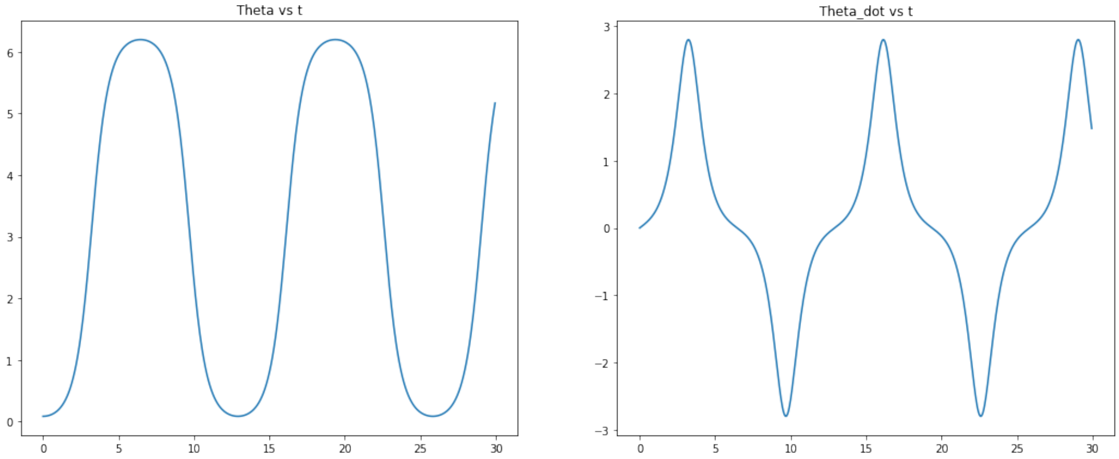


0.0.2 For Inverse Pendulum,

[95]:

```
t=[0]
th=[0.0873] #Theta
th_d=[0] #Theta_dot
# Count number of iterations using step size h
n = (int)(tf/h)
# Iterate for number of iterations
t0=0
for i in range(1,n):
    "Apply Runge Kutta Formulas to find next value of y"
    k1 = Ftheta(t0,th[-1],th_d[-1])
    k1_d = Ftheta_inv(0,th[-1],th_d[-1])
    k2 = Ftheta(t0+0.5*h, th[-1]+0.5*h*k1,th_d[-1]+0.5*h*k1_d)
    k2_d = Ftheta_inv(t0+0.5*h, th[-1]+0.5*h*k1,th_d[-1]+0.5*h*k1_d)
    k3 = Ftheta(t0+0.5*h, th[-1]+0.5*h*k2,th_d[-1]+0.5*h*k2_d)
    k3_d = Ftheta_inv(t0+0.5*h, th[-1]+0.5*h*k2,th_d[-1]+0.5*h*k2_d)
    k4 = Ftheta(t0+h, th[-1]+h*k3,th_d[-1]+h*k3_d)
    k4_d = Ftheta_inv(t0+h, th[-1]+h*k3,th_d[-1]+h*k3_d)

    # Update next value of theta
    th.append(th[-1] + (1/6)*(k1 + 2 * k2 + 2 * k3 + k4)*h)
    th_d.append(th_d[-1] + (1/6)*(k1_d+2*k2_d+2*k3_d+k4_d)*h)
    # Update next value of t
    t0 = t0 + h
    t.append(t0)
fig,a =  plt.subplots(1,2,figsize=(18,7))
a[0].plot(t,th)
a[0].set_title('Theta vs t')
a[1].plot(t,th_d)
a[1].set_title('Theta_dot vs t')
plt.show()
```



Theta vs t

Theta_dot vs t

[68]:
```
A=[[0,1,0],[0,0,1],[-6,-11,-6]]
w,v = np.linalg.eig(A)
print(w)
```

```
[-1. -2. -3.]
```