Python for Physics
Lab -6

One Dimension Kinematics

The position of a particle moving in a straight line is given by

$$X = 5 + 2t + 4t2 - t3$$

where x is in meter. (a) Find an expression for the Velocity and Acceleration as a function of time. (b) Find the position of the particle at t=1 sec

In [722]:

```
import sympy as sp
sp.init_printing()
t = sp.symbols('t')
Position = 2*t + 4*t**2 + t**3+ 5
velocity = sp.diff(Position,t)
acceleration = sp.diff(Position,t,2)
```

print('The position is : ') Position

In [723]:

```
print('The velocity is : ')
velocity
```

```
The velocity is :
```

Out[723]: $3t^2 + 8t + 2$

Out[723]: $3t^2 + 8t + 2$

```
In [724]: print('The acceleration is : ')
          acceleration
```

The acceleration is :

Out[724]: $2(3t + 4)$

Define function

In [725]:
```
import math
from scipy.misc import derivative

def f(x):
    fn = math.sin(x)
    return fn

derivative(f,45, dx =0.1)
```
Out[725]: 0.5244468898338156

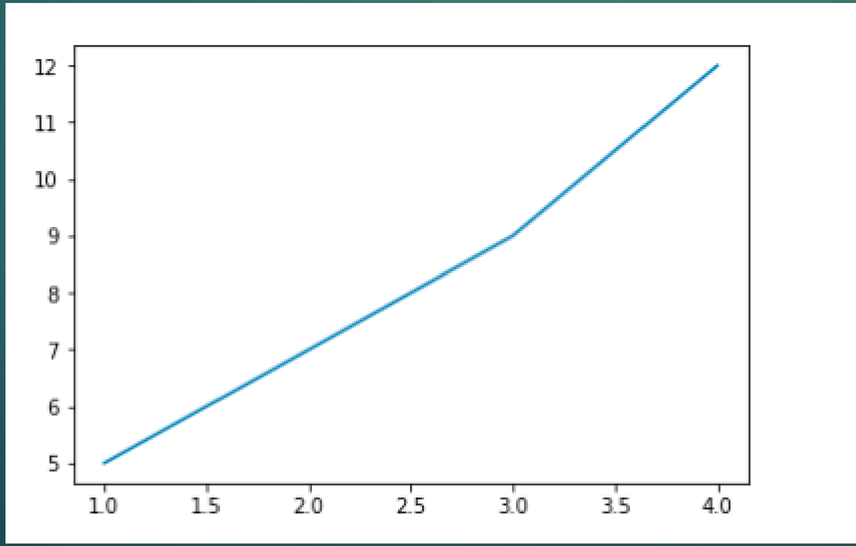
In [726]:
```
math.cos(45)
```
Out[726]: 0.5253219888177297

In [727]:
```
derivative(math.cos,45, dx =1e-2)
```
Out[727]: -0.8508893428794517

In [728]:
```
math.sin(45)
```
Out[728]: 0.8509035245341184

Matplotlib

```
In [706]: import matplotlib.pyplot as plt

          x = np.array([1,2,3,4])
          y = np.array([5,7,9,12])

          plt.plot(x,y)
          plt.show()
```



Free Fall Motion

$g = \frac{F}{m}$

$v = g * t$

$h = \frac{1}{2} * g * t^2$

In [735]:
```
np.arange(1,10,.1)
```

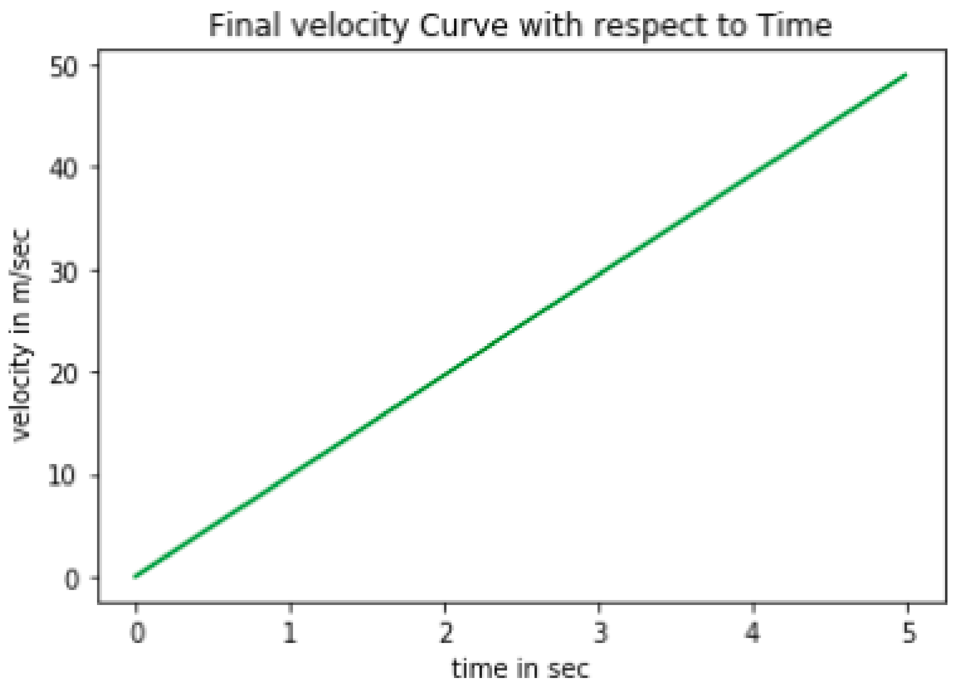
Out[735]:
```
array([1. , 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2. , 2.1, 2.2,
       2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 3. , 3.1, 3.2, 3.3, 3.4, 3.5,
       3.6, 3.7, 3.8, 3.9, 4. , 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8,
       4.9, 5. , 5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.7, 5.8, 5.9, 6. , 6.1,
       6.2, 6.3, 6.4, 6.5, 6.6, 6.7, 6.8, 6.9, 7. , 7.1, 7.2, 7.3, 7.4,
       7.5, 7.6, 7.7, 7.8, 7.9, 8. , 8.1, 8.2, 8.3, 8.4, 8.5, 8.6, 8.7,
       8.8, 8.9, 9. , 9.1, 9.2, 9.3, 9.4, 9.5, 9.6, 9.7, 9.8, 9.9])
```

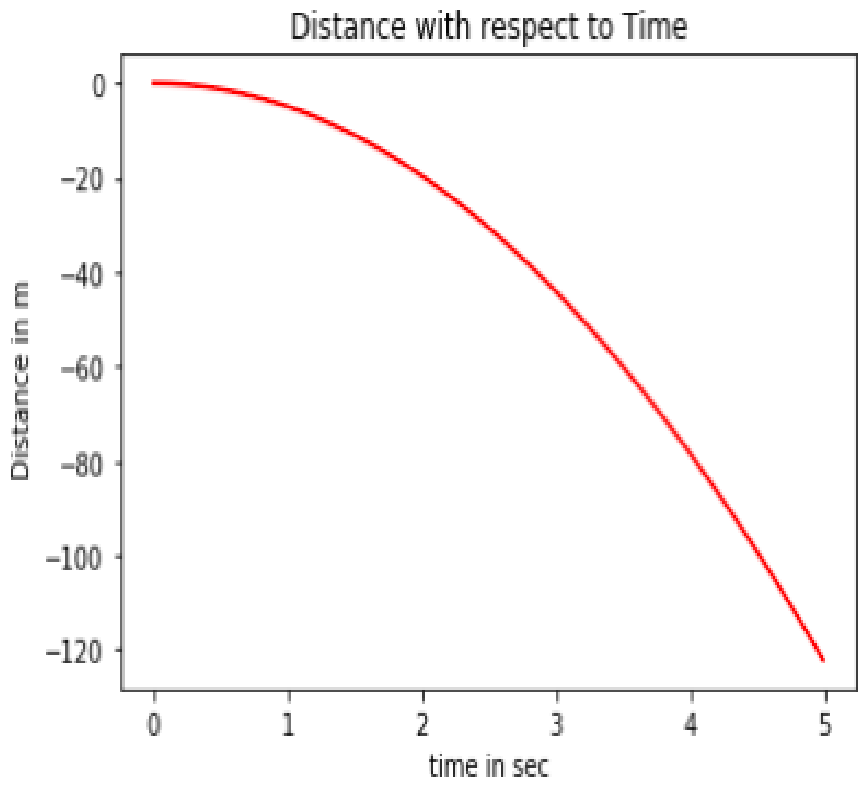
```
In [736]: #Input Variable:
# tfinal = final time (in seconds)
# Output Variables:
# t = array of times at which speed is % computed (in seconds)
# v = array of speeds (meters/second)

g = 9.81  # Acceleration in SI units
tfinal = int(input('Enter final time (in seconds): '))
dt = tfinal/500
t =np.arange(0,tfinal,dt)   # Creates an array of 501 time values
# the final velocity
v = g*t
```

```
# The Distance travel by the object
D = - (0.5*g*t**2)
plt.plot(t,v, 'g')
plt.xlabel('time in sec')
plt.ylabel('velocity in m/sec ')
plt.title('Final velocity Curve with respect to Time')
plt.show()
plt.plot(t,D, 'r')
plt.xlabel('time in sec')
plt.ylabel('Distance in m ')
plt.title('Distance with respect to Time')
plt.show()
```

```
Enter final time (in seconds): 5
```

```
Enter final time (in seconds): 5
```





Projectile Motion without making Custom Functions

The equation of Projectile's trajectory is

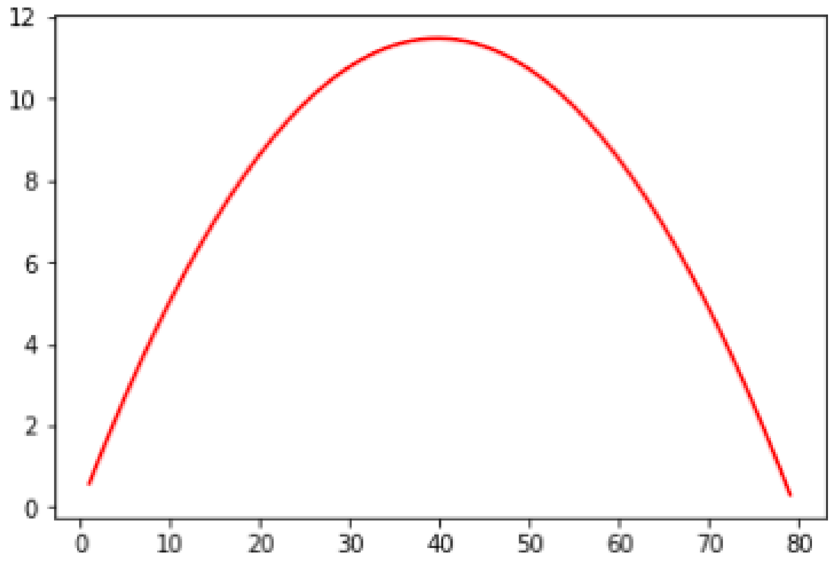
$$y = \tan\theta \cdot x - \frac{g}{2 \cdot u^2 \cdot \cos^2\theta} \cdot x^2$$

In [569]:

```
# Projectile's trajectory
import math
x = np.arange(1,80, 1)
g = 9.8
```

```
v0 = 30
theta = math.radians(30)
# The equation of Projectile's trajectory is :

y = x* math.tan(theta)-(x**2 * g)/(2 * v0**2 * (math.cos(theta)**2))
plt.plot(x,y,'r')
```

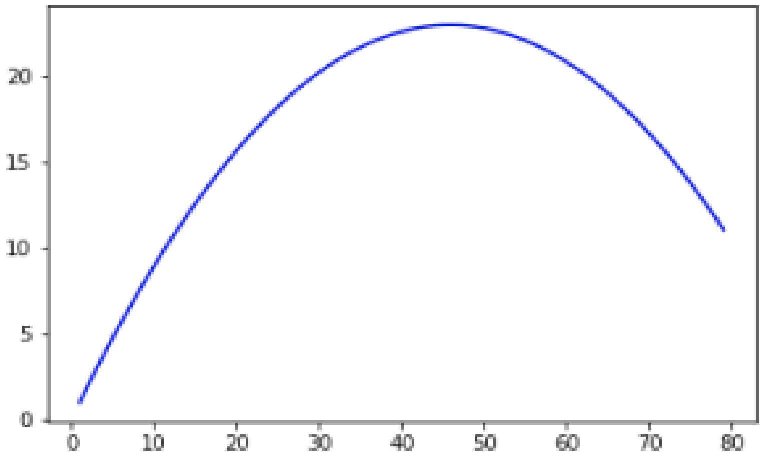
Out[569]: [<matplotlib.lines.Line2D at 0x253ccc27630>]



In [570]:
```
theta1 = math.radians(45)
y1 = x* math.tan(theta1)-(x**2 * g)/(2 * v0**2 * (math.cos(theta1)**2))
plt.plot(x,y1,'b')
```

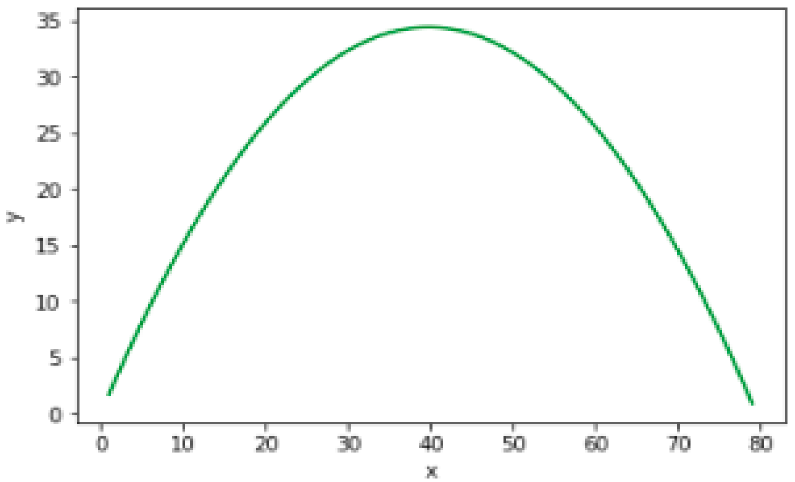
Out[570]: [<matplotlib.lines.Line2D at 0x253ccbd5a90>]

```
In [570]: theta1 = math.radians(45)
          y1 = x* math.tan(theta1)-(x**2 * g)/(2 * v0**2 * (math.cos(theta1)**2))
          plt.plot(x,y1,'b')

Out[570]: [<matplotlib.lines.Line2D at 0x253ccbd5a90>]
```

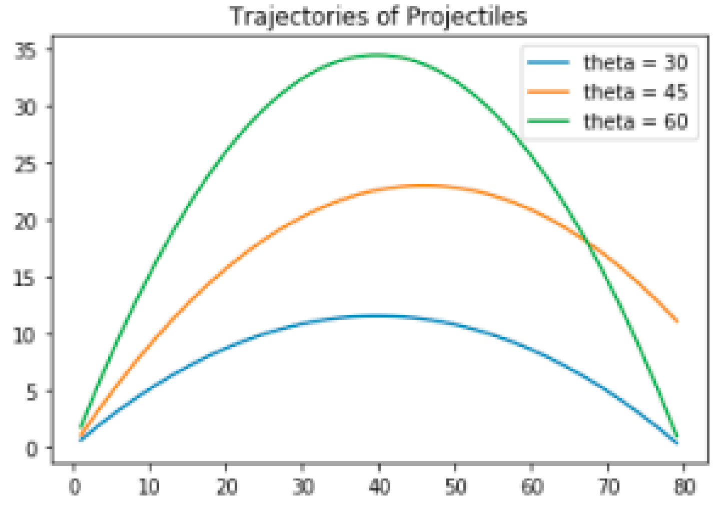


In [572]:
```
theta2 =math.radians(60)
y2 = x * math.tan(theta2)-(x**2 * g)/(2 * v0**2 * (math.cos(theta2)**2
))
plt.plot(x,y2,'g')
plt.xlabel('x')
plt.ylabel('y')
```

Out[572]: `Text(0,0.5,'y')`



```
In [573]: ax = plt.subplot(111)

          ax.plot(x, y, label='theta = 30')
          ax.plot(x, y1, label='theta = 45')
          ax.plot(x, y2, label='theta = 60')
          plt.title('Trajectories of Projectiles')
          ax.legend()
          plt.show()
```

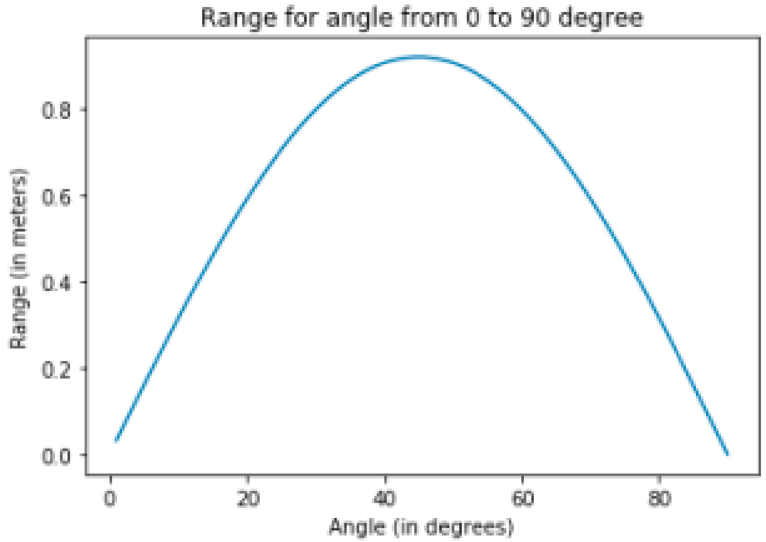


Range and Height of projectile

```
In [574]: # Range of projectile
          import math
          def Range(angle):
              v = 3
              R = v**2 *(np.sin(np.radians(2*angle)))/ g
              return R

          angle = np.arange(1, 90, 0.01)
```

```
plt.plot(angle, Range(angle))
plt.xlabel('Angle (in degrees)')
plt.ylabel('Range (in meters)')
plt.title ('Range for angle from 0 to 90 degree')
```

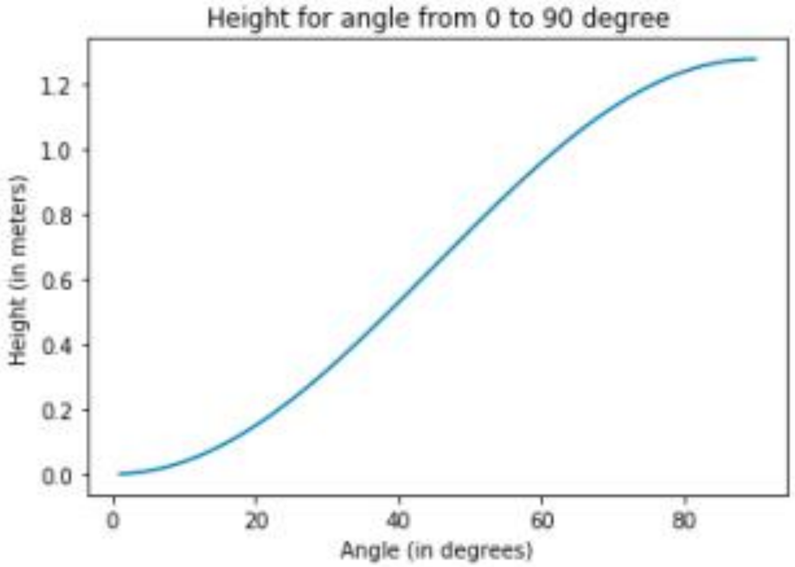
Out[574]: Text(0.5,1,'Range for angle from 0 to 90 degree')



In [575]:

```
# Height of projectile
def Height(angle):
    v = 5
    return v**2 *(np.sin(np.radians(angle))**2)/ (g*2)

angle = np.arange(1, 90, 0.01)
plt.plot(angle,Height(angle))
plt.xlabel('Angle (in degrees)')
plt.ylabel('Height (in meters)')
plt.title ('Height for angle from 0 to 90 degree')
```

Out[575]: Text(0.5,1,'Height for angle from 0 to 90 degree')



Projectile Motion by making Custom Functions

In [576]:
```
# Range of projectile
import math
def Range(angle,in_vel):
    R = in_vel**2 *(np.sin(np.radians(2*angle)))/ g
    return R

# Height of projectile
def Height(angle,in_vel):
    return in_vel**2 *(np.sin(np.radians(angle))**2)/ (g*2)

Height(45,3), Range(45,3)
```

Out[576]: (0.2295918367346939, 0.9183673469387754)