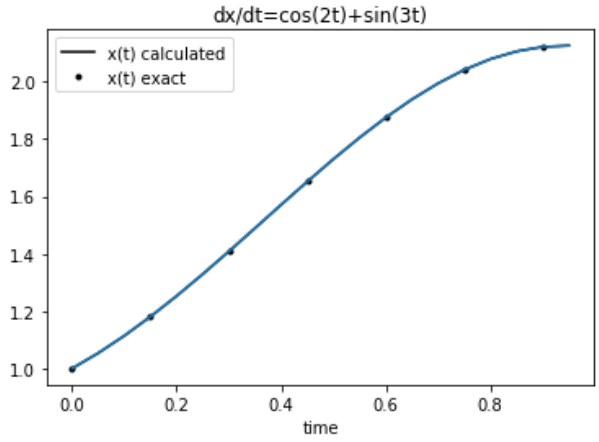
Ex 6

#Solve for x, dx/dt=cos2t+sin3t,initialcond x0=1

```
import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt

deffunct(x,t):
dxdt= cos(2*t)+sin(3*t)
    return dxdt
#initial condition
x0=1
t=np.arange(0.0,1.0,0.05)
x=odeint(funct,x0,t)
x1=0.5*np.sin(2*t)-(1.0/3.0)*np.cos(3*t)+4.0/3.0
#time  plot
plt.plot(t,x,'k', label='x(t) calculated')
plt.plot(t,x1,'k.',markevery=3,label='x(t) exact')
plt.plot(t,x)
plt.xlabel('time')
#plt.ylabel('funct')
plt.legend()
plt.title('dx/dt=cos(2t)+sin(3t)')
plt.show()
```

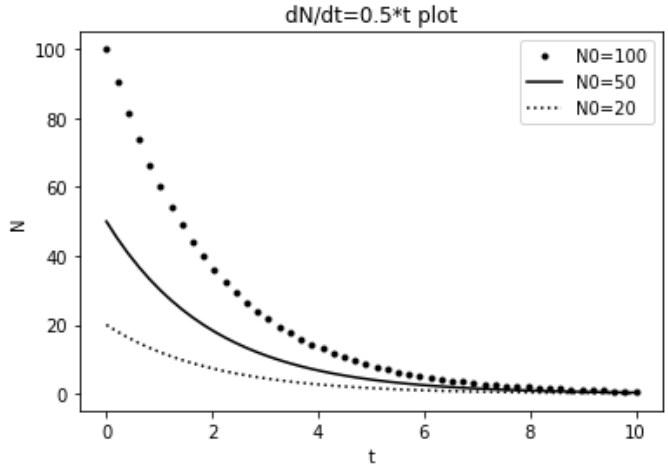


Ex 7

#Solve for N, dN/dt=-0.5t,initialcond N0=100,50,20

```
import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt

deffunct(N,t):
dNdt=-0.5*N
   return dNdt
#initial condition not mentioned separately, included in function calculation N1, N2, N3
#time plot
t = np.linspace(0,10)
#solve ODE
N1=odeint(funct,100,t)
plt.plot(t,N1,'k.',label=r'N0=100')
N2=odeint(funct,50,t)
plt.plot(t,N2,'k-',label=r'N0=50')
N3=odeint(funct,20,t)
plt.plot(t,N3,'k:',label=r'N0=20')
plt.xlabel('t')
plt.ylabel('N')
```

```
plt.legend()
plt.title('dN/dt=0.5*t plot')
plt.show()
```



Ex 8

```
# solve d2x/dt2+2bdx/dt+kx=0,initial cond. at t=0, x=0 and dx/dt=1
# b=.5,k=2
```

Theory:

$$\frac{d^2x}{dt^2} + 2b\frac{dx}{dt} + kx = 0$$

Step :1 Break it into 2 equations. Put , $\frac{dx}{dt} = y$. Therefore, $\frac{dy}{dt} + 2by + kx = 0$

Step:2 Get 2 equations as

$$\frac{dx}{dt} = y$$

$$\frac{dy}{dt} + 2by + kx = 0$$

Step:3 Get initial conds x0=0 and y0=1

Step:4 Solve for x and y as function of t i.esolve for a function of x, y and ti.efind f(x,y,t)?

To write PYTHON code

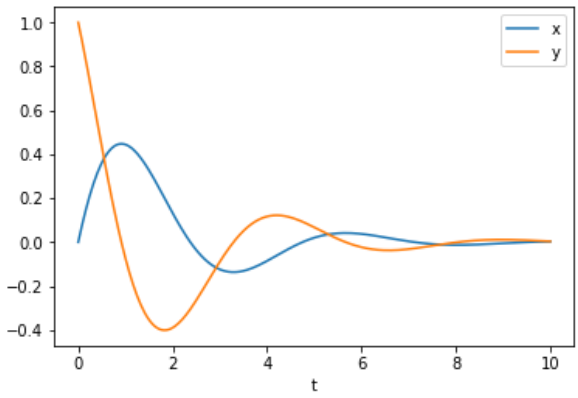
Pack x and y in X such that 0th element of X is x and 1th element is y

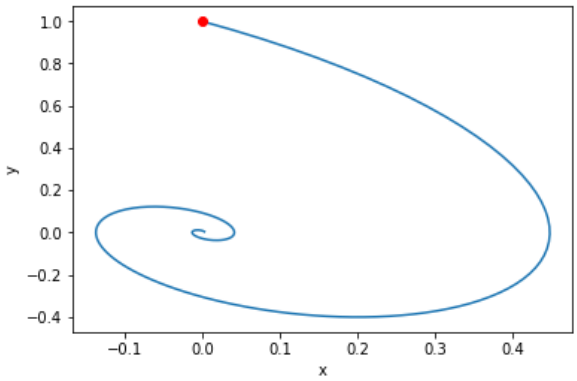
Now f(x,y,t) is just f(X,t). We will find f(X,t) only.

PYTHON CODE using ODEINT

```
# solve d2x/dt2+2bdx/dt+kx=0,initial cond. at t=0, x=0 and dx/dt=1
# b=.5,k=2
from scipy.integrate import odeint
import numpy as np



deffunct(X, t):
    x = X[0]
    y = X[1]
dxdt = y
dydt = -2*b*y-k*x
    return [dxdt, dydt]


b=0.5
k=2
X0 = [0, 1]
t = np.linspace(0, 40, 250)

sol = odeint(funct, X0, t)


import matplotlib.pyplot as plt
x = sol[:, 0]
y = sol[:, 1]
```

```
plt.plot(t,x, t, y)
plt.xlabel('t')
plt.legend(('x', 'y'))
#plt.savefig('images/funct-1.png')

# phase portrait
plt.figure()
plt.plot(x,y)
plt.plot(x[0], y[0], 'ro')
plt.xlabel('x')
plt.ylabel('y')
#plt.savefig('images/funct-2.png')
```





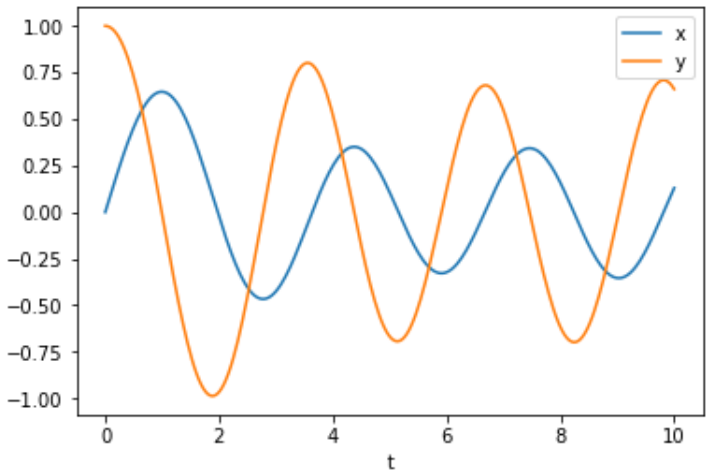
EX 9

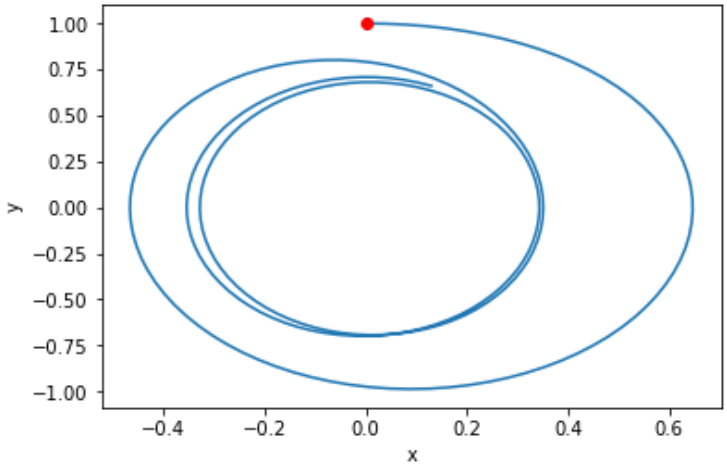
solve d2x/dt2+2bdx/dt+kx=Fcos2t,initial cond. at t=0, x=0 and dx/dt=0 ,

b=.5,k=2,F=1

```
from scipy.integrate import odeint
import numpy as np

deffunct(X, t):
    x = X[0]
    y = X[1]
dxdt = y
dydt = -2*b*y-k*x+F*cos(2*t)
    return [dxdt, dydt]

b=0.5
k=2
F=1

X0 = [0, 1]
t = np.linspace(0, 10, 250)

sol = odeint(funct, X0, t)

import matplotlib.pyplot as plt
x = sol[:, 0]
y = sol[:, 1]

plt.plot(t,x, t, y)
plt.xlabel('t')
plt.legend(('x', 'y'))
#plt.savefig('images/funct-1.png')
```

```
# phase portrait
plt.figure()
plt.plot(x,y)
plt.plot(x[0], y[0], 'ro')
plt.xlabel('x')
plt.ylabel('y')
#plt.savefig('images/funct-2.png')
```





INSTRUCTION

COMPLETE LNB WITHIN A WEEK FROM THE DATE MENTIONED ABOVE. NEXT SHEET WILL BE UPLOADED A WEEK LATER.