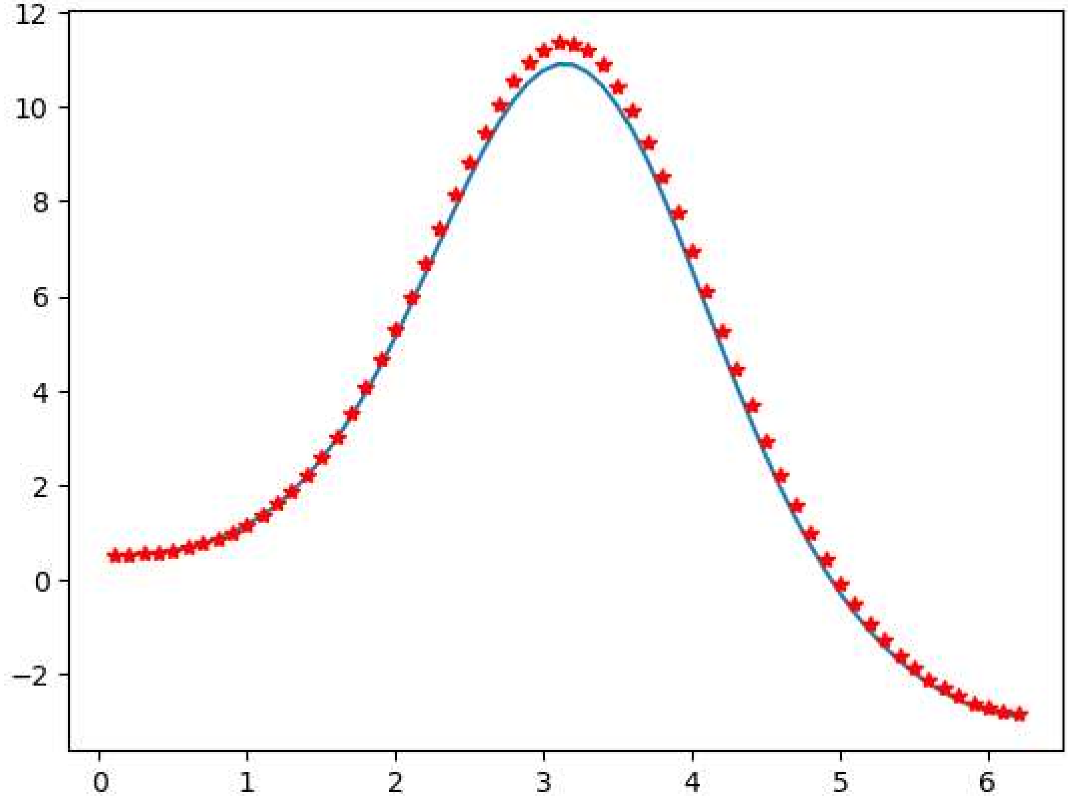
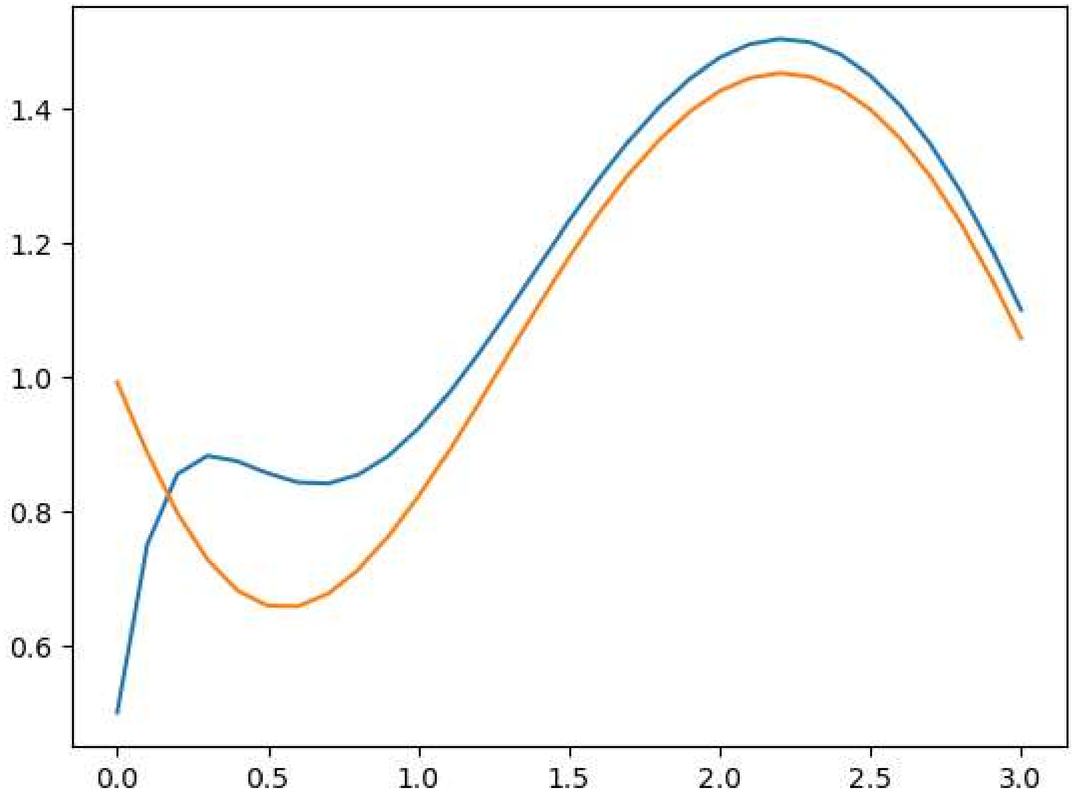
In [23]:
```
#TA 3
#Q1

def f(x,y):
    z=(x+y)*np.sin(x)
    return z

y0=0.5
h=0.1
x0=0

#RK2 Method
def rk2(x,y,h):
        k1=h*f(x,y0)
        k2=h*f(x+h/2,y+k1/2)
        y2=y+k2
        return y2

#RK4 Method
def rk4(x,y,h):
        k1=h*f(x,y)
        k2=h*f(x+h/2,y+k1/2)
        k3=h*f(x+h/2,y+k2/2)
        k4=h*f(x+h,y+k3)
        y4=y+(k1+2*k2+2*k3+k4)/6
        return y4

X=[]
Yr2=[]
Yr4=[]
y2=y0
y4=y0
x=x0
#Loop
while x<6.1:
    Y2=rk2(x,y2,h)
    Y4=rk4(x,y4,h)
    x=x+h
    y2=Y2
    y4=Y4
    X.append(x)
    Yr2.append(y2)
    Yr4.append(y4)

import matplotlib.pyplot as plt
plt.plot(X,Yr2)
plt.plot(X,Yr4,'r*')

!jt -r
```

```
Reset css and font defaults in:
C:\Users\91782\.jupyter\custom &
C:\Users\91782\AppData\Roaming\jupyter\nbextensions
```


12
10
8
6
4
2
0
−2
0
1
2
3
4
5
6

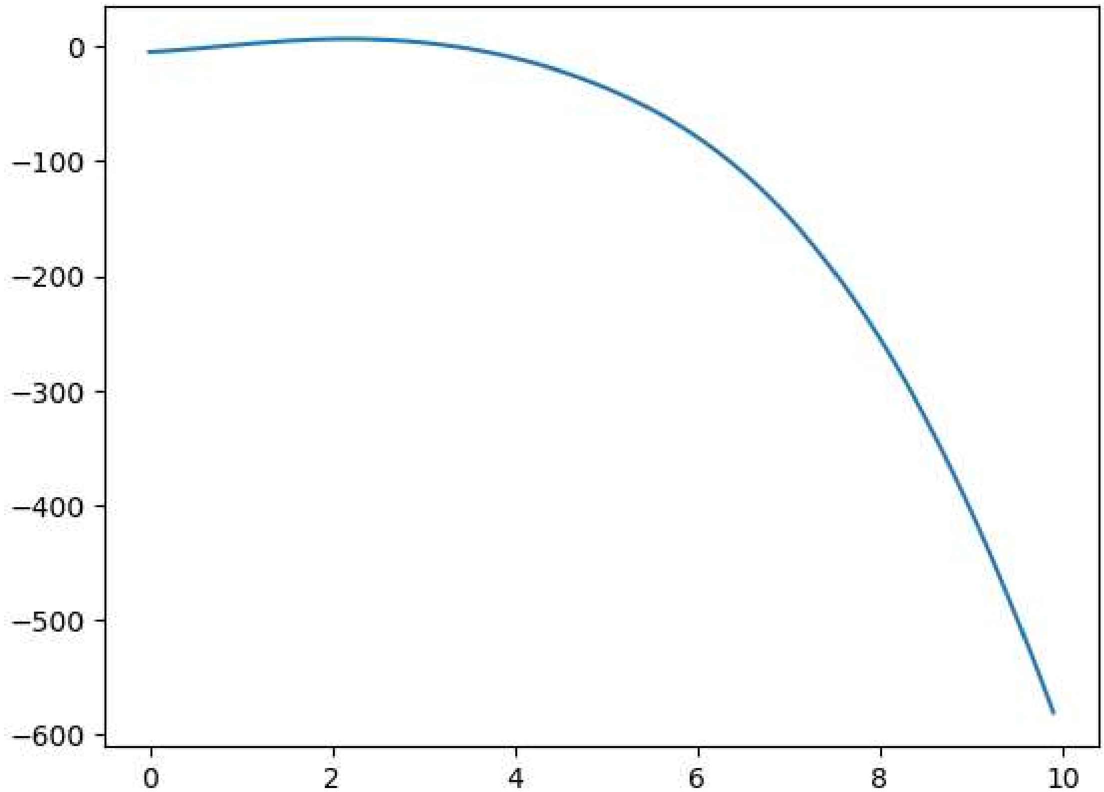
In [5]:
```
#Q2

def f(x,y,z):
    dydx=z
    dzdx=10*np.sin(x)-5*z-6*y
    return dydx,dzdx

x=0
y=0
z=5
h=0.1

import numpy as np

#Euler
Ye=[]
Ze=[]
for x in np.arange(0,3.1,h):
    k1,k2=f(x,y,z)
    y=y+h*k1
    z=z+h*k2
    Ye.append(y)
    Ze.append(z)

#RK4
Y4=[]
Z4=[]
for x in np.arange(0,3.1,h):
    k1y,k1z=f(x,y,z)
    k2y,k2z=f(x+h/2,y+h*k1y/2,z+h*k1z/2)
    k3y,k3z=f(x+h/2,y+h*k2y/2,z+h*k2z/2)
    k4y,k4z=f(x+h/2,y+h*k3y,z+h*k3z)
    y=y+h*(k1y+2*k2y+2*k3y+k4y)/6
    z=z+h*(k1z+2*k2z+2*k3z+k4z)/6
    Y4.append(y)
    Z4.append(z)

X=np.arange(0,3.1,h)

import matplotlib.pyplot as plt
plt.plot(X,Ye)
plt.plot(X,Y4)
```

Out[5]:
```
[<matplotlib.lines.Line2D at 0x1a2e33da810>]
```


1.4
1.2
1.0
0.8
0.6
0.0
0.5
1.0
1.5
2.0
2.5
3.0

In [9]:
```
#Q3
import numpy as np
import matplotlib.pyplot as plt

x=np.arange(0,10,0.1)
y=5*np.sin(x)-x**3+(132/32)*x**2-(28/32)*x-(147/32)

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

```
[-4.59375000e+00 -4.14183292e+00 -3.61840335e+00 -3.03439897e+00
 -2.40065829e+00 -1.72787231e+00 -1.02653763e+00 -3.06911564e-01
  4.21030454e-01  1.14763455e+00  1.86360492e+00  2.56003680e+00
  3.22844543e+00  3.86079093e+00  4.44949865e+00  4.98747493e+00
  5.46811802e+00  5.88532405e+00  6.23348815e+00  6.50750044e+00
  6.70273713e+00  6.81504683e+00  6.84073202e+00  6.77652606e+00
  6.61956590e+00  6.36736072e+00  6.01775686e+00  5.56889940e+00
  5.01919075e+00  4.36724665e+00  3.61185004e+00  2.75190331e+00
  1.78637928e+00  7.14271529e-01 -4.65455510e-01 -1.75391614e+00
 -3.15235222e+00 -4.66218070e+00 -6.28503945e+00 -8.02283080e+00
 -9.87776248e+00 -1.18523856e+01 -1.39496289e+01 -1.61728297e+01
 -1.85257604e+01 -2.10126506e+01 -2.36382050e+01 -2.64076163e+01
 -2.93265730e+01 -3.24012631e+01 -3.56383714e+01 -3.90450734e+01
 -4.26290233e+01 -4.63983372e+01 -5.03615724e+01 -5.45277016e+01
 -5.89060832e+01 -6.35064277e+01 -6.83387609e+01 -7.34133833e+01
 -7.87408275e+01 -8.43318125e+01 -9.01971970e+01 -9.63479305e+01
 -1.02795004e+02 -1.09549400e+02 -1.16622043e+02 -1.24023750e+02
 -1.31765183e+02 -1.39856801e+02 -1.48308817e+02 -1.57131155e+02
 -1.66333411e+02 -1.75924817e+02 -1.85914210e+02 -1.96310000e+02
 -2.07120152e+02 -2.18352159e+02 -2.30013033e+02 -2.42109293e+02
 -2.54646959e+02 -2.67631551e+02 -2.81068097e+02 -2.94961141e+02
 -3.09314755e+02 -3.24132564e+02 -3.39417765e+02 -3.55173154e+02
 -3.71401164e+02 -3.88103896e+02 -4.05283158e+02 -4.22940508e+02
 -4.41077300e+02 -4.59694728e+02 -4.78793873e+02 -4.98375756e+02
 -5.18441384e+02 -5.38991803e+02 -5.60028146e+02 -5.81551679e+02]
```



In [27]:
```
#Lab Sheet4
#Q1

def f(x):
    y=x**3-5*x+3
    return y

a=1
b=2
tol=0.001

#Bisection Method
def bm(a,b):
    if f(a)*f(b)>=0:
        return None
    for i in range (100):
        c=(a+b)/2
        if f(c)==0 or abs(b-a)<tol:
            return c
        if f(c)*f(a)<0:
            b=c         #Upar b ko replace karna success hua to neeche change
        else:
            a=c
    return (a+b)/2

root=bm(a,b)
print(root)
```

```
1.83447265625
```

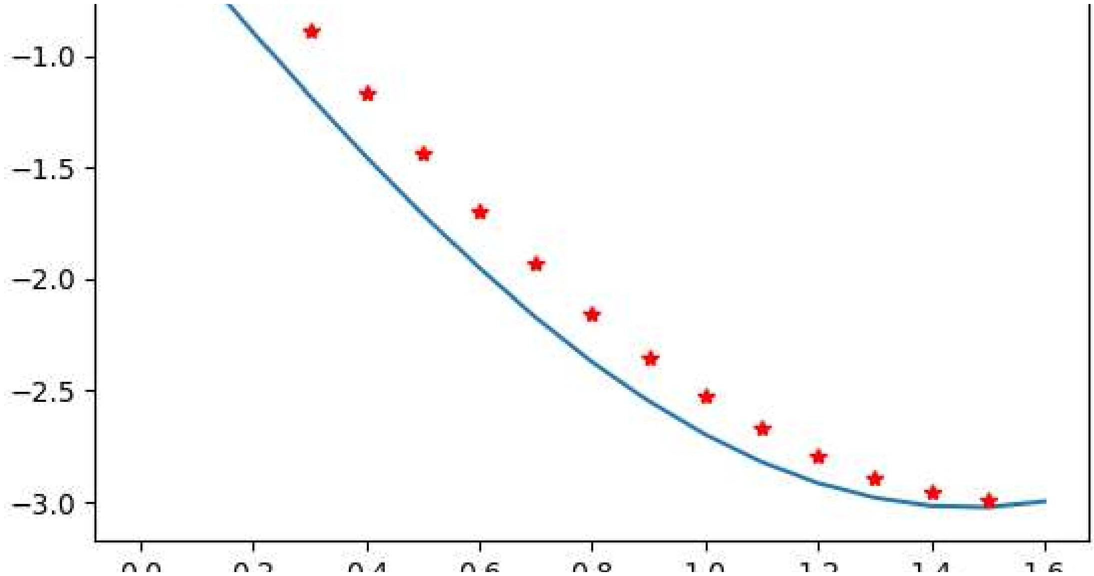
In [26]:
```
#Q2
#Newton Method
def f(x):
    y=x**3-x-4
    return y

def df(x):
    dy=3*x**2-1
    return dy

tol=0.001

def nm(f,df,x0):
    x0=0.3
    for i in range(100):
        x=x0-(f(x0)/df(x0))
        if abs(x-x0)<tol:   #abs imp hai
            break
        x0=x
    return x

x=nm(f,df,x0)
print(x)
```

```
1.7963219032716558
```

In [35]:
```
#Q3

def f(x):
    y=x**3-x-4
    return y

def df(x):
    dy=3*x**2-1
    return dy

#Newton Method
def nm(x0):
    x0=0.5
    tol=0.001
    for i in range(100):
        x=x0-(f(x0)/df(x0))
        if abs(x-x0)<tol:
            break
        x0=x
    return x
Xnm=nm(x0)

#Secant Method
def sm(a,b,f):
    tol=0.001
    a=2
    b=1
    if f(a)*f(b)>=0:
        return None
    for i in range(100):
        c=(a+b)/2
        if f(c)==0 or abs(b-a)<tol:
            return c
        if f(c)*f(a)<0:
            b=c
        else:
            a=c
        return (a+b)/2
Xsm=sm(a,b,f)

print(Xnm,"NM")
print(Xsm,"SM")
```

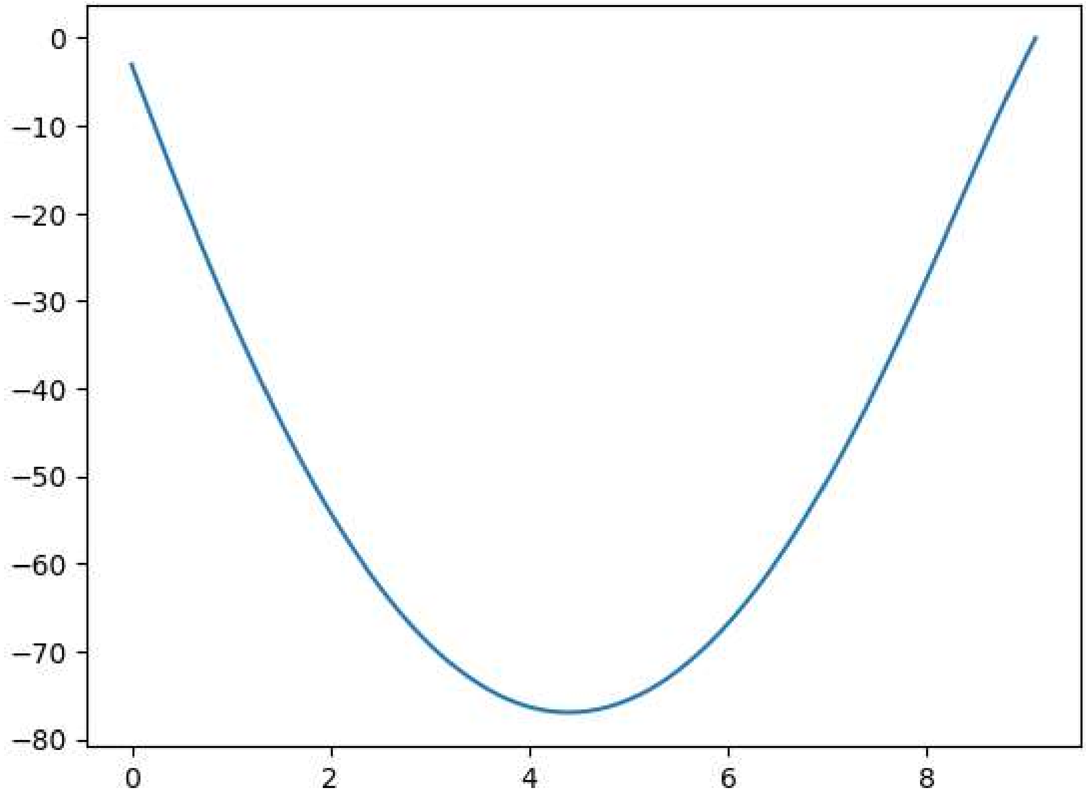
```
1.7963219349961688 NM
1.75 SM
```

In [11]:
```
#Lab Sheet 5
#Q1

import numpy as np

def f(x,y,z):
    dydx=z
    dzdx=-y
    return dydx,dzdx

#BC
y=0
x=0
yL=-3
xL=np.pi/2
h=0.1

X=[]
while x<np.pi/2+0.11:
    X.append(x)
    x=x+h

#RK2 Method
def rk2(x,y,z,h):
    Y=[]
    for x in X:
        k1y,k1z=f(x,y,z)
        k2y,k2z=f(x+h/2,y+h*k1y/2,z+h*k1z/2)
        y=y+h*k2y
        z=z+h*k2z   #Ye imp hai bcz bhale hi z ka kaam na ho but loop me kaam
        Y.append(y)
    return Y

#Gusses
za=1
zb=-6

ya=rk2(x,y,za,h)[-1]
yb=rk2(x,y,zb,h)[-1]

zdes=(zb-za)/(yb-ya)*(yL-ya)+za

y2=rk2(x,y,zdes,h)

d=np.arange(0,np.pi/2,0.1)
yn=-3*np.sin(d)


import matplotlib.pyplot as plt
plt.plot(X,y2)
plt.plot(d,yn,'r*')
```


−1.0
−1.5
−2.0
−2.5
−3.0

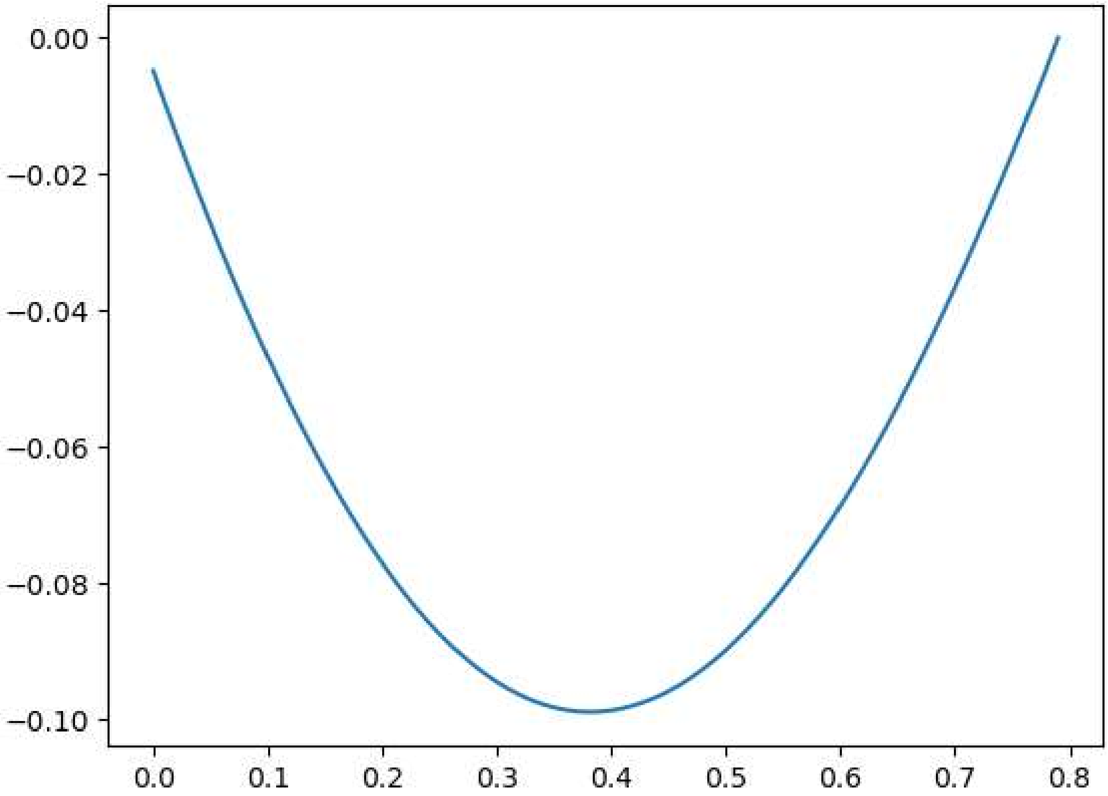
In [12]:
```
#Q2

import numpy as np

def f(x,y,z):
    dydx=z
    dzdx=8*x*(9-x)+2*y
    return dydx,dzdx

#BC
y=0
x=0
yL=0
xL=9
h=0.1

X=[]
while x<xL+h:
    X.append(x)
    x=x+h

#RK2 Method
def rk2(x,y,z,h):
    Y=[]
    for x in X:
        k1y,k1z=f(x,y,z)
        k2y,k2z=f(x+h/2,y+h*k1y/2,z+h*k1z/2)
        y=y+h*k2y
        z=z+h*k2z   #Ye imp hai bcz bhale hi z ka kaam na ho but Loop me kaam
        Y.append(y)
    return Y

#Gusses
za=1
zb=-6

ya=rk2(x,y,za,h)[-1]
yb=rk2(x,y,zb,h)[-1]

zdes=(zb-za)/(yb-ya)*(yL-ya)+za

y2=rk2(x,y,zdes,h)

import matplotlib.pyplot as plt
plt.plot(X,y2)
```

Out[12]:
```
[<matplotlib.lines.Line2D at 0x24a082e4590>]
```


0
−10
−20
−30
−40
−50
−60
−70
−80
0
2
4
6
8

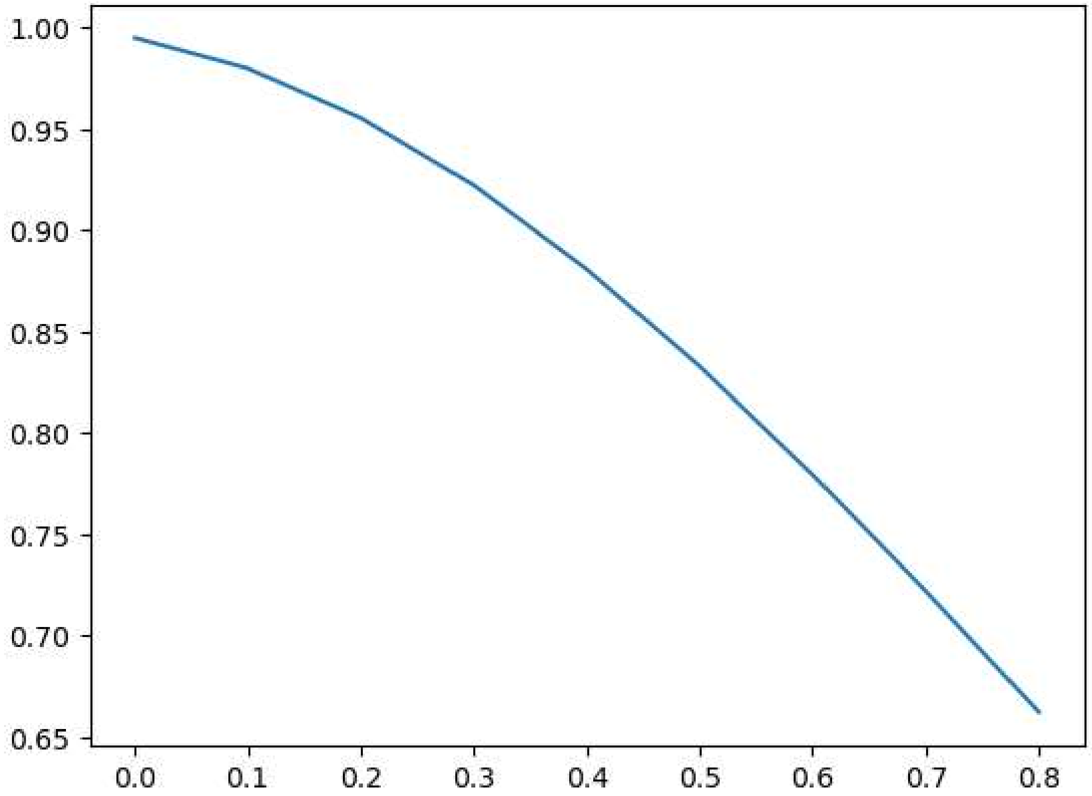
In [18]:
```
#Q3

import numpy as np
import matplotlib.pyplot as plt

#defining function
def f(x,y,z):
    dydx=z
    dzdx=np.cos(x)-4*y
    return dydx,dzdx

#Boundary Conditions
y=0  #My advice, isko y0 karke mat karna
x=0
yL=0
xL=np.pi/4

h=0.01  #Step Size

#Guess
za=-1
zb=6

X=[]
while x<xL+h:
    X.append(x)
    x=x+h

#RK4 Method
def rk4(x,y,z,h):
    Y=[]
    for x in X:
        k1y,k1z=f(x,y,z)
        k2y,k2z=f(x+h/2,y+h*k1y/2,z+h*k1z/2)
        k3y,k3z=f(x+h/2,y+h*k2y/2,z+h*k2z/2)
        k4y,k4z=f(x+h,y+h*k1y,z+h*k1z)
        y=y+h*(k1y+2*k2y+2*k3y+k4y)/6
        z=z+h*(k1z+2*k2z+2*k3z+k4z)/6    #ye baar baar bhool rahe ho
        Y.append(y)
    return Y

#Shooting Method
ya=rk4(x,y,za,h)[-1]
yb=rk4(x,y,zb,h)[-1]

zdes=(zb-za)/(yb-ya)*(yL-ya)+za

y4=rk4(x,y,zdes,h)

plt.plot(X,y4)
```

Out[18]:
```
[<matplotlib.lines.Line2D at 0x24a084fe350>]
```

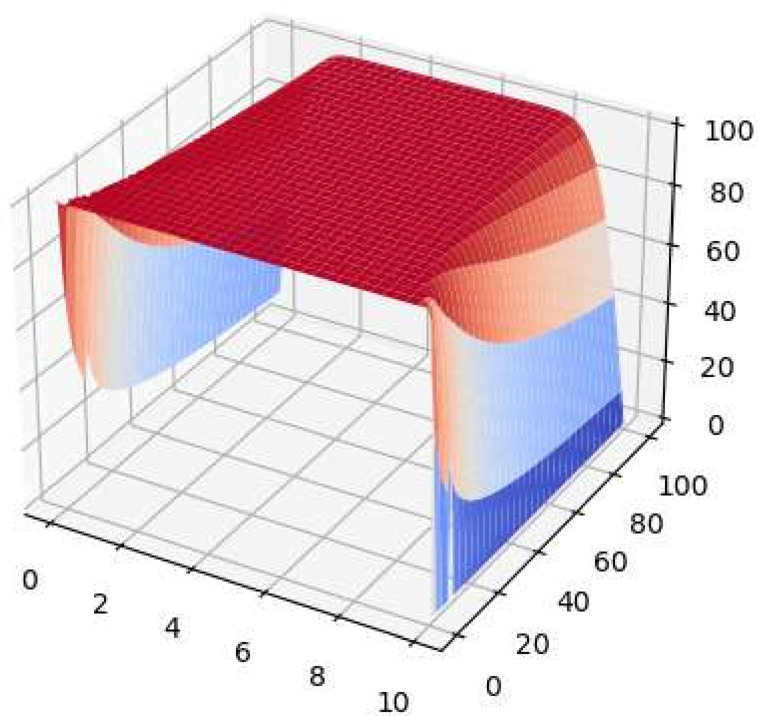

0.00
−0.02
−0.04
−0.06
−0.08
−0.10
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8

In [28]:
```
#Q4

import matplotlib.pyplot as plt

#defining function
def f(x,y,z):
    dydx=z
    dzdx=x*(z**2)-y**2
    return dydx,dzdx

#Boundary Conditions
y=1  #My advice, isko y0 karke mat karna
x=0
z=0

h=0.1  #Step Size

X=[]
while x<xL+h:
    X.append(x)
    x=x+h

#RK4 Method
def rk4(x,y,z,h):
    Y=[]
    for x in X:
        k1y,k1z=f(x,y,z)
        k2y,k2z=f(x+h/2,y+h*k1y/2,z+h*k1z/2)
        k3y,k3z=f(x+h/2,y+h*k2y/2,z+h*k2z/2)
        k4y,k4z=f(x+h,y+h*k1y,z+h*k1z)
        y=y+h*(k1y+2*k2y+2*k3y+k4y)/6
        z=z+h*(k1z+2*k2z+2*k3z+k4z)/6    #ye baar baar bhool rahe ho
        Y.append(y)
    return Y

Y=rk4(x,y,z,h)
y2=rk4(x,y,z,h)[-1]
plt.plot(X,Y)
print(y2)
```

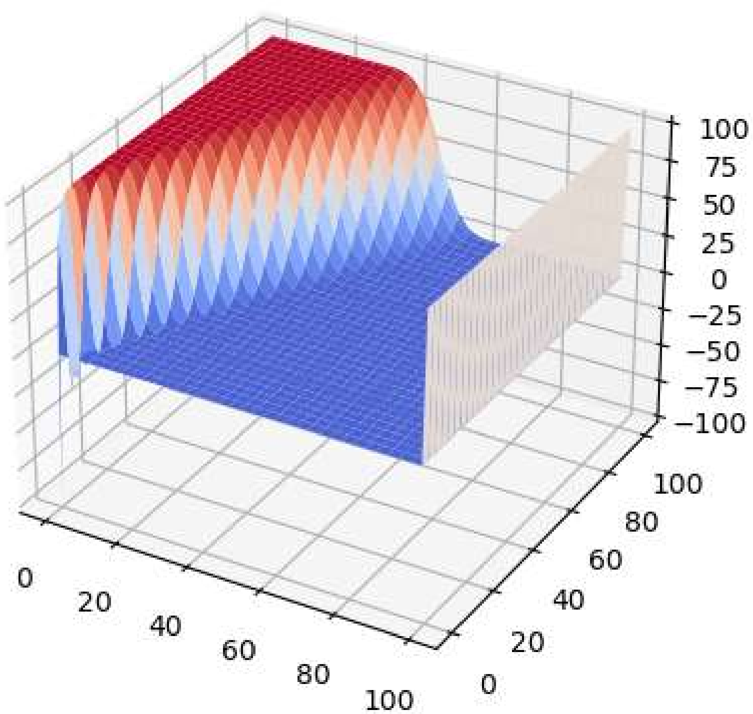
```
0.6622874863368237
```


1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8

In [44]:
```
#Lab Sheet 7
#Q1 (Heat Equation)

import numpy as np
L=10 #cm
x=np.linspace(0,10,101)
t=np.linspace(0,100,101)
u=np.zeros((101,101))

u[:,0]=0  #u(x,t)=T
u[0,:]=100
u[L,:]=50

a=0.25

for n in range(100):
    for j in range(100):
        u[n+1,j]=u[n,j]+a*(u[n,j+1]-2*u[n,j]+u[n,j-1])  #Explicit

import matplotlib.pyplot as plt
from matplotlib import cm

X,T=np.meshgrid(x,t)
fig=plt.figure()
ax=plt.axes(projection='3d')
ax.plot_surface(X,T,u,cmap=cm.coolwarm)
plt.show()
```

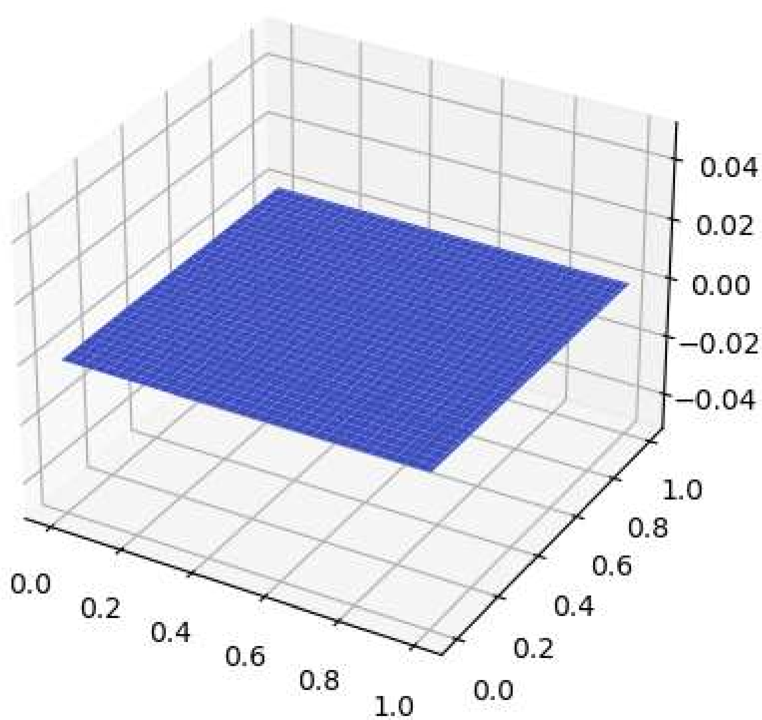


In [45]:
```
#Q2(Laplace Equation)

L=100 #cm
x=np.linspace(0,L,101)
y=np.linspace(0,L,101)
u=np.zeros((101,101))

u[0,:]=0
u[L,:]=0
u[:,0]=-100
u[:,L]=100

a=0.5

for n in range(100):
    for j in range(100):
        u[n+1,j]=u[n,j]-a*(u[n,j]-u[n,j-1])  #Upwind

import matplotlib.pyplot as plt
from matplotlib import cm
X,Y=np.meshgrid(x,y)
fig=plt.figure()
ax=plt.axes(projection='3d')
ax.plot_surface(X,Y,u,cmap=cm.coolwarm)
plt.show()
```



In [48]:

```
#Lab Sheet 6
#Q1

L=1 #cm
import numpy as np
x=np.linspace(0,L,101)
y=np.linspace(0,L,101)
u=np.zeros((101,101))

u[:,0]=np.sin(3*np.pi*x/2)
v=1
dx=0.01
dt=dx/2
a=v*dx/dt

for n in range(100):
    for j in range(100):
        u[n+1,j]=u[n,j]-a*(u[n,j]-u[n,j-1])  #Upwind

import matplotlib.pyplot as plt
from matplotlib import cm
X,Y=np.meshgrid(x,y)
fig=plt.figure()
ax=plt.axes(projection='3d')
ax.plot_surface(X,Y,u,cmap=cm.coolwarm)
plt.show()
```



In [ ]: