常微分方程边值问题

公式推导

$$\begin{cases} \frac{d^2\phi}{dx^2} = -k^2\phi \\ \phi(x=0) = \phi(x=1) = 0 \end{cases}$$

转化为初值问题

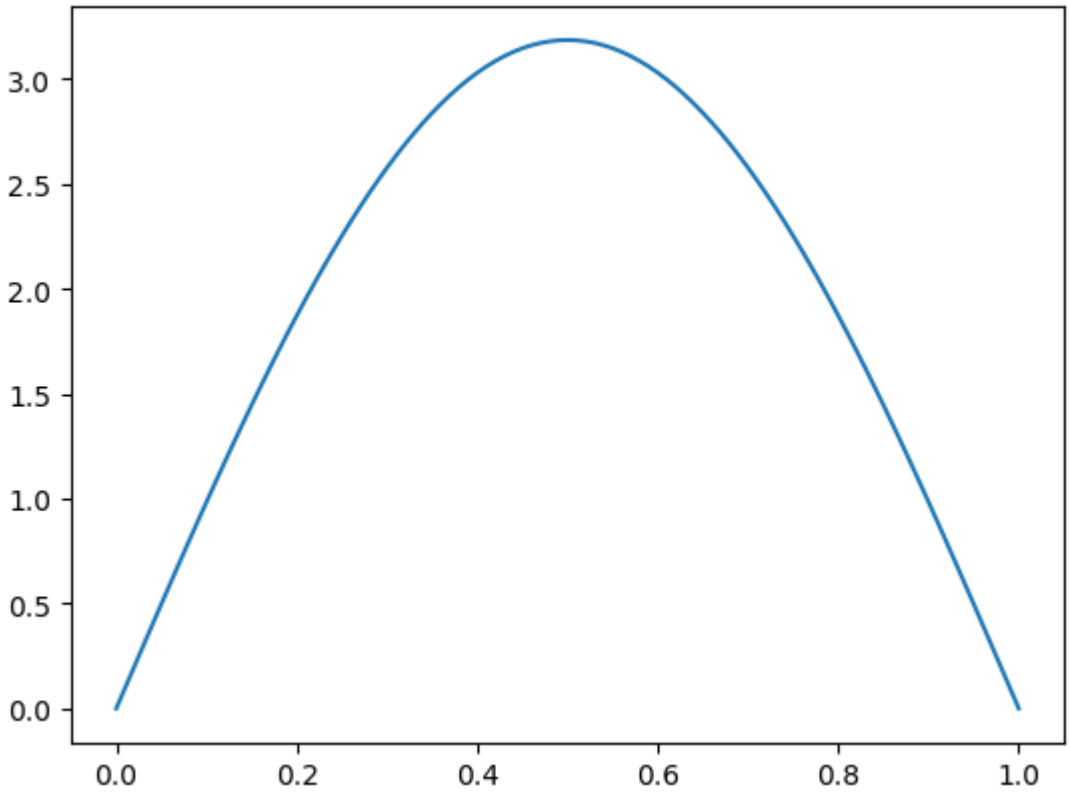
令 $\frac{d\phi}{dx} = y$，有 $\begin{cases} \frac{dy}{dx} = -k^2\phi \\ \frac{d\phi}{dx} = y \\ \phi(x=0) = 0 \\ y(x=0) = \text{任意} \end{cases}$

代码实现

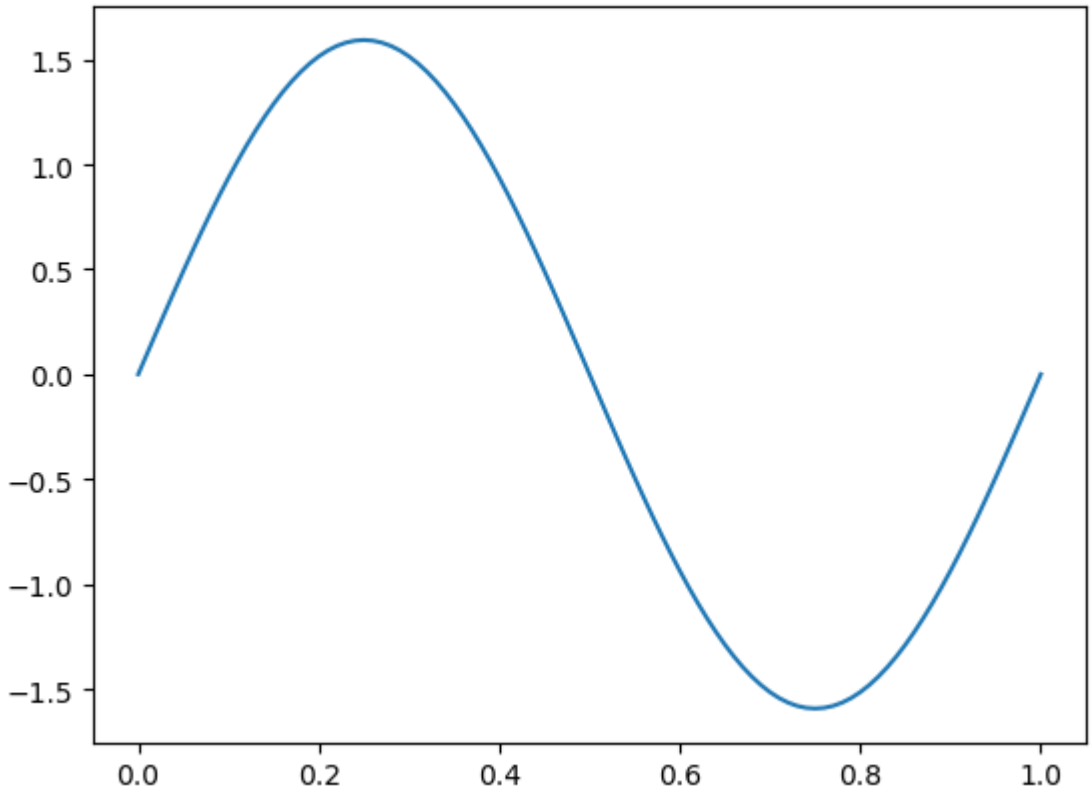
```
import numpy as np
import matplotlib.pyplot as plt
derv_phi = lambda phi, y, x, k:y                                    ###定义函数
derv_y = lambda phi, y, x, k: -k**2*phi
def Runge_Kutta(x, phi, y, k, h):                                   ###龙格库塔方法
    phihalf = phi+derv_phi(phi, y, x, k)*h*0.5
    yhalf = y+derv_y(phi, y, x, k)*h*0.5
    phinext = phi+derv_phi(phihalf, yhalf, x+0.5*h, k)*h
    ynext = y+derv_y(phihalf, yhalf, x+0.5*h, k)*h
    return (phinext, ynext)
def Integral(xx, phi_list, y_list, N, h, k):
    for i in range(N):
        phi_list[i+1], y_list[i+1] = Runge_Kutta(xx[i], phi_list[i], y_list[i], k, h)
    return
xa = 0.0                                                            ###设置参数
xb = 1.0
N = 100
h = (xb-xa)/N
xx = np.linspace(xa, xb, N+1, dtype=np.float64)
phi_list = np.zeros(N+1, dtype=np.float64)
y_list = np.zeros(N+1, dtype=np.float64)
y_list[0] = 10
phi_list[0] = 0
k = 8                                                               ###设置k的搜索起点
addk = 0.1
Integral(xx, phi_list, y_list, N, h, k)                             ###第一次打靶
k = k+addk                                                          ###更改k值
loss = (phi_list[-1])**2                                            ###计算方差
oldphi_list = np.array([phi_list[-1]])                              ###记录历次打靶的终值
while loss>=0.00000000001:
    oldphi = oldphi_list[-1]
    Integral(xx,phi_list,y_list,N,h,k)                              ###循环打靶
    recentphi = phi_list[-1]
    if oldphi*recentphi>0:                                          ###使用搜索法
        k = k+addk
        oldphi_list = np.append(oldphi_list,recentphi)
    else:
        addk = addk/2
        k = k-addk
    loss = (phi_list[-1])**2                                        ###更新方差
    print(k)
plt.figure()                                                        ###画图
plt.plot(xx, phi_list)
plt.show()
```

画图

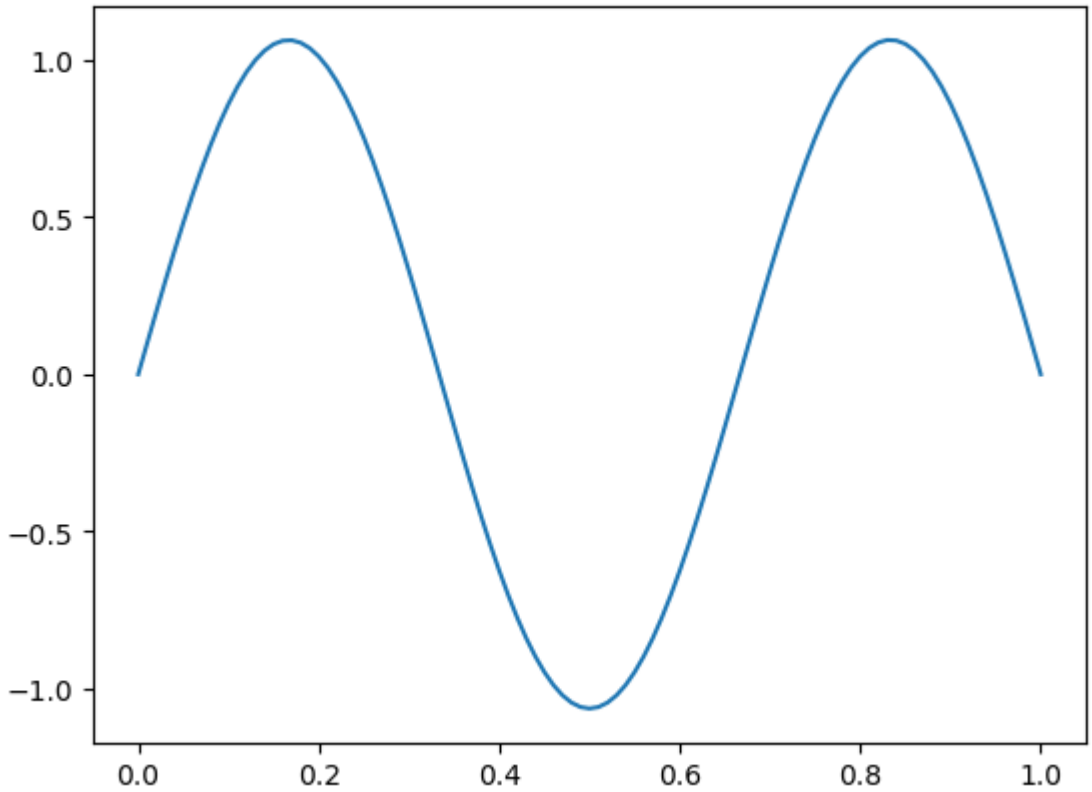
第一个本征值k=π



第二个本征值$k=2\pi$



第三个本征值$k=3\pi$


1.0
0.5
0.0
−0.5
−1.0
0.0
0.2
0.4
0.6
0.8
1.0