**核安全综合保障**

**基于SPH方法的U材料力学性能探究**

|  |  |  |
| --- | --- | --- |
| 姓名 |  | 梁欣 |
| 学号 |  | 201900501005 |

# 练习1：基于SPH方法的导数求解器

## 1.SPH方法基本原理

SPH方法采用光滑核函数，对Dirac delta函数进行近似。与Dirac delta函数可以表示函数类似，光滑核函数可以近似表示函数f(x)：

(1a)

经典的光滑核函数有三次样条、五次样条、钟形函数等，它们分别具有不同的近似精度。本练习中使用的光滑核函数是二维情况下的钟形函数:

(1b)

其中的由下式给出:

(1c)

通过光滑核函数表示的,在进行微分、积分运算的时候，可以转化成关于光滑核函数的一个黎曼和表达式，从而可以利用计算机进行数值模拟:

(1d)

## 2.基于 SPH方法的导数求解器原理

由公式(1d)，得到只有一个自变量x时候，表达式：

(2a)

自变量一维的时候，

(2b)

(2c)

上式中的为，是对求梯度，是对或者求梯度。则:

(2d)

由之前的表达式(1b)，可以计算出:

(2e)

其中，

(2f)

将式(2c),(2d),(2e)，代入式(2a),得到基于SPH方法计算导数的原理式:

(2g)

其中，

(2h)

## 3.可执行源代码

#include  
import matplotlib.pyplot as plt  
from sympy import \*  
import math  
#a,b=(input("请输入下界和上界:").split())  
#d=(input("请输入维度:"))  
#h=(input("请输入步长:"))  
a=-math.pi  
b=math.pi  
d=1  
dx=(b-a)/100  
h=2\*dx  
x=[]  
y=[]  
y1=[]  
y2=[]  
if d==1:  
    a4=5/4/h  
if d==2:  
    a4=5/3.14/h/h  
if d==3:  
    a4=105/16/3.14/h/h/h  
for i in range(0,100):  
    x.append(a+i\*dx)  
    y.append(sin(a+i\*dx))  
    y2.append(cos(a+i\*dx))  
for i in x:  
    sum=0  
    for j in x:  
        if i!=j:  
            r=abs((i-j))/h  
              
            if r>1:  
                dew=0  
            else:  
                dew=a4\*(1-r)\*(1-r)\*(-12\*r)\*(i-j)/sqrt((i-j)\*\*2)/h  
                #dew=(3\*(1-r)\*\*3-((1-r)\*\*2)\*(1+3\*r))\*a4/h\*r/sqrt(r\*\*2)  
                #print(dew)  
            sum=sum+dx\*sin(j)\*dew  
        #print(sum)  
    y1.append(sum)  
      
plt.xlabel("x")  
plt.ylabel("y")  
#plt.plot(x,y)  
plt.plot(x,y1)  
#plt.plot(x,y2)  
plt.show()

图表, 折线图, 直方图

描述已自动生成练习2：三物块弹簧链接动力学计算

## 基本物理模型

计算在给定初始条件(1)、(2)下，此系统在0-10s内的位移-时间图像、能量-时间图像。

## 图示 描述已自动生成

1. v0=3m/s, x0=0m
2. v0=0m/s, x0=1m

## 2. 计算方法

将时间微分dt，根据t时刻的位移得出t时刻的加速度a，利用差分近似的原理求出t+dt时刻的速度v，根据t+dt时刻的位移和速度求a出t+2dt时刻的各个运动参量，依此类推得出全时间的运动参量。

求出各时刻的动能和弹性势能，验证是否满足能量守恒定律，验证结果的正确性。

## 可执行源代码

#include  
import matplotlib.pyplot as plt  
from sympy import \*  
import math  
min=-math.pi  
max=math.pi  
dx=(max-min)/100  
h=2\*dx  
x=[]  
y=[]  
y1=[]  
y2=[]  
a=5/4/h  
for i in range(0,100):  
    x.append(min+i\*dx)  
    y.append(sin(min+i\*dx))  
    y2.append(cos(min+i\*dx))  
for i in x:  
    sum=0  
    for j in x:  
        if i!=j:  
            r=abs((i-j))/h  
            if r>1:  
                dw=0  
            else:  
                dw=a\*(1-r)\*(1-r)\*(-12\*r)\*(i-j)/sqrt((i-j)\*\*2)/h  
            sum=sum+dx\*sin(j)\*dw  
    y1.append(sum)  
plt.xlabel("x")  
plt.ylabel("y")  
plt.plot(x,y1)  
plt.show()

计算结果:

1. v0=3m/s, x0=0m

图形用户界面

描述已自动生成

图形用户界面

描述已自动生成

1. v0=0m/s, x0=1m

图形用户界面

描述已自动生成

图片包含 图形用户界面

描述已自动生成