**中国矿业大学计算机学院**

**2021 级本科生课程报告**

课程名称： 最优化理论与方法

报告题目： 《三次插值法原理及实现》

报告时间： 2024.1.10

学生姓名： 杨学通

学 号： 08213129

专 业： 人工智能2021-1班

任课教师： 赵佳琦

**课程报告成绩评定及评分依据**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 评分依据 | 分值 | 实际得分 |
| 1 | 课程报告描述了最优化技术或算法的基本原理 | 20 |  |
| 2 | 课程报告使用python或C++程序设计语言（附源代码） | 40 |  |
| 3 | 给出至少2个测试用例，并且给出可视化的运行结果及分析 | 30 |  |
| 4 | 课程报告给出了个人学习体会 | 10 |  |
| 总分 | | |  |
| 教师评语：  教师签名：  年 月 日 | | | |

**摘 要**

三次插值算法是一种用于在给定数据点之间进行插值的方法。三次插值算法利用三次多项式来逼近相邻数据点之间的曲线，以实现平滑的插值。通过三次多项式的性质，可以确保插值曲线在相邻数据点处的一阶导数和二阶导数连续，从而实现平滑插值。在实际应用中，需要注意插值误差的问题，并根据具体情况选择合适的插值方法和参数。 三次插值算法在信号处理、图像处理、数值分析等领域都有广泛的应用。例如，在数字信号处理中，可以利用三次插值算法对离散采样的信号进行插值，从而实现信号的平滑重构。

**关键词**：三次插值算法；平滑插值；插值误差；插值方法；参数

目录

[1问题描述 1](#_Toc155817813)

[2算法原理及流程 1](#_Toc155817814)

[3运行代码 2](#_Toc155817815)

[4运行截图 3](#_Toc155817816)

[5总结与分析 4](#_Toc155817817)

# 

# 1问题描述

三次插值法是函数逼近法的一种。其基本思想是利用三次函数逼近目标函数。三次插值法具有以下优点：

平滑性：三次插值法能够生成平滑的插值曲线，因为使用的是三次多项式来逼近数据点之间的曲线。这意味着在相邻数据点处的一阶导数和二阶导数都是连续的，从而得到的插值曲线具有良好的平滑性。

精度：在适当的情况下，三次插值法能够提供较高的插值精度，即使在数据点较稀疏的情况下也能够较好地逼近原始数据。这使得三次插值法在需要高精度插值的应用中具有优势。

理论基础：三次插值法有坚实的数学理论基础，可以通过拉格朗日插值或者样条插值等方法得到具体的插值多项式。这使得三次插值法在数值分析领域有较好的可靠性和稳定性。

本次实验要求python实现三次插值法，并通过测试样例，验证三次插值法的正确性，体会三次插值法进行一维搜索的过程。

# 2算法原理及流程

Step1:给定初始点，计算

要求满足条件：，

给定允许误差

Step2:计算

Step3:若，则停止计算，得点；否则转Step4

Step4:计算

若，则停止计算，得点

若，则令，，，转Step2

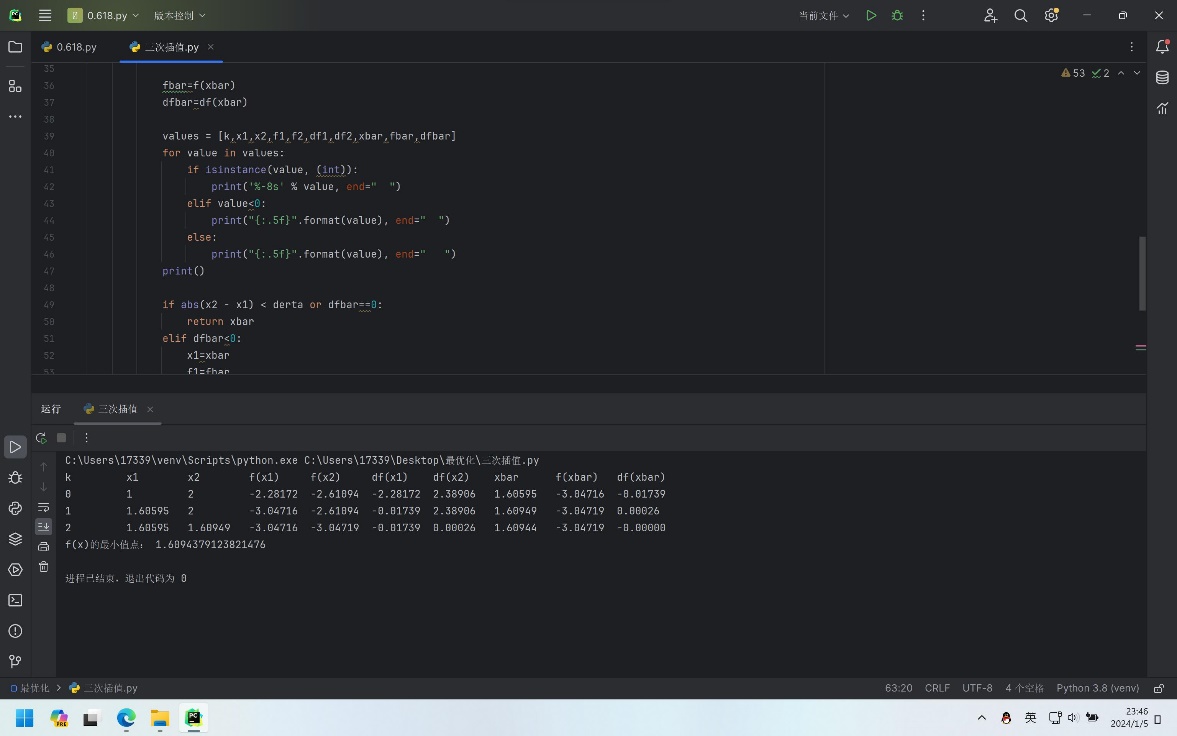
若，则令，，，转Step2

# 3运行代码

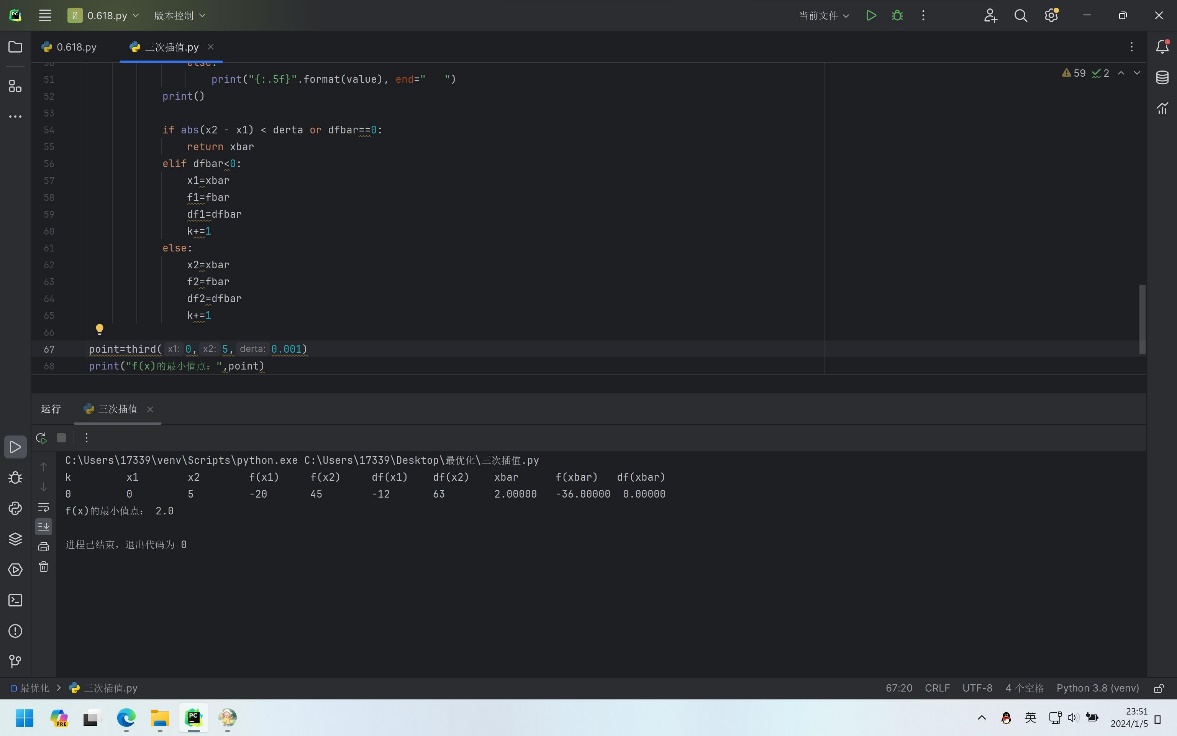
import math  
  
def f1(x):  
 return math.e\*\*x-5\*x  
  
def df1(x):  
 return math.e\*\*x-5  
  
def f(x):  
 return x\*\*3-12\*x-20  
  
def df(x):  
 return 3\*x\*\*2-12  
def third(x1,x2,derta):  
  
 keys = ["k", "x1", "x2", "f(x1)", "f(x2)", "df(x1)", "df(x2)", "xbar", "f(xbar)", "df(xbar)"]  
 for key in keys:  
 print('%-8s' % key, end=" ")  
 print()  
 f1=f(x1)  
 f2=f(x2)  
 df1=df(x1)  
 df2=df(x2)  
 k=0  
  
 if(x1>=x2 or df1>=0 or df2<=0):  
 return  
 else:  
 while 1:  
 s=3\*(f2-f1)/(x2-x1)  
 z=s-df1-df2  
 w=z\*\*2-df1\*df2  
 w=math.sqrt(w)  
  
 *#求x一拔* xbar=df2+w+z  
 xbar=xbar/(df2-df1+2\*w)  
 xbar=1-xbar  
 xbar=(x2-x1)\*xbar+x1  
  
 fbar=f(xbar)  
 dfbar=df(xbar)  
  
 values = [k,x1,x2,f1,f2,df1,df2,xbar,fbar,dfbar]  
 for value in values:  
 if isinstance(value, (int)):  
 print('%-8s' % value, end=" ")  
 elif value<0:  
 print("{:.5f}".format(value), end=" ")  
 else:  
 print("{:.5f}".format(value), end=" ")  
 print()  
  
 if abs(x2 - x1) < derta or dfbar==0:  
 return xbar  
 elif dfbar<0:  
 x1=xbar  
 f1=fbar  
 df1=dfbar  
 k+=1  
 else:  
 x2=xbar  
 f2=fbar  
 df2=dfbar  
 k+=1  
  
point=third(0,5,0.001)  
print("f(x)的最小值点：",point)

# 4运行截图

实例1：，初始点，



实例2：，初始点，



# 5总结与分析

通过本次实验，我用python实现了三次插值法。通过使用不同的函数测试，取得很好的效果。虽然三次插值法有较好的平滑性，较高的精度，坚实的理论基础，但不可否认的是三次插值法的计算复杂度很高，需要计算插值多项式的系数，特别是在数据点较多的情况下，计算复杂度较高，可能导致较大的计算开销。在某些情况下，三次插值法可能会导致插值曲线出现振荡现象。所以在选用此方法时要权衡各方面的因素，尽量节省资源开销得到更低的时间复杂度。