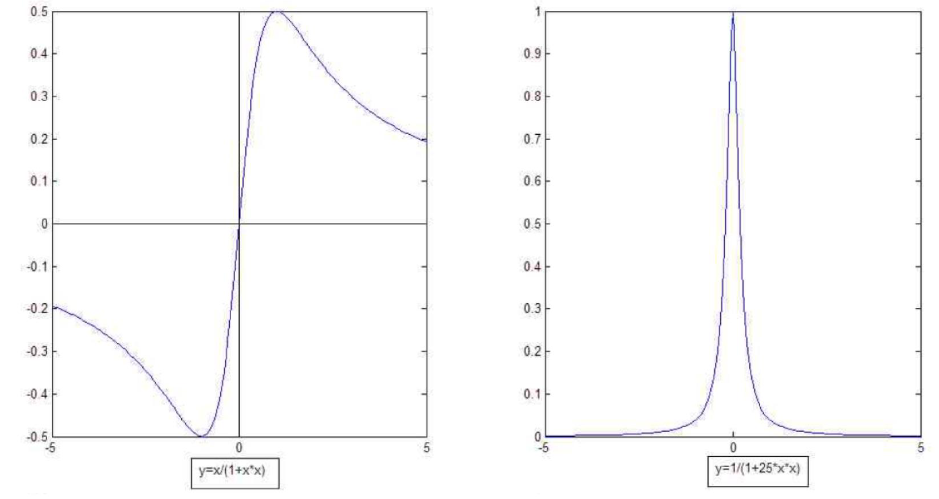
# 第三次上机实验报告

姓名：刘龙 学号：3119009062 班级：软件工程2班

## 实验 物体运动轨迹的插值预测，比较不同的插值方法

【实验目的】

使用拉格朗日插值、分段线性插值、样条插值等方法对物体的运动轨迹进行插值，并绘制插值函数的图形，根据结果，确定一种最好的插值方法

其中物体有以下两种运动轨迹

【实验条件】

计算机配置：64 位操作系统, 基于 x64 的处理器

CPU：Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz 1.99 GHz

内存大小：8GB

操作系统：Windows

【算法介绍】

1. 插值基本概念

对于给定的坐标： ，求得函数 使得 。依据插值基本概念，可以构建n阶多项式

 （1）

将坐标 代入公式（1）可以构建出如下线性方程组

 （2）

通过求解上述线性方程组，求得系数 ，即可得到

1. 拉格朗日插值法

对于 的插值基函数 ，其在 的函数值都为0，于是有

 （3）

又因为 ，所以

 （4）

于是得到插值基函数为：

 （5）

最后得到拉格朗日插值函数为

 （6）

1. 牛顿法
2. 差商定义

对于n+1个互不相同的坐标 ，一阶差商如下定义

 （7）

二阶差商如下定义

 （8）

于是可以得到n阶差商定义

 （9）

1. 插值函数

 （10）

【代码】

拉格朗日插值法

#!/usr/bin/env python

# -\*- coding: utf-8 -\*-

import matplotlib.pyplot as plt

def lagrange(X, Y, xx):

"""拉格朗日插值法"""

result = 0.0

for i in range(len(Y)):

f\_temp = Y[i]

for j in range(len(Y)):

if i != j:

f\_temp \*= (xx - X[j]) / (X[i] - X[j])

result += f\_temp

return result

def plot\_image(X, Y, xq, yq, num):

# 绘图

plt.title("Lagrange\_interpolation") # 打印标题

plt.plot(X, Y, 's', label="original values") # 蓝色点表示原来的值

plt.plot(xq, yq, 'r', label="interpolation values") # 插值曲线

words = "被插点数：{}".format(num)

plt.text(3.5, -10, words, fontproperties='SimHei')

plt.xlabel('x')

plt.ylabel('y')

plt.legend(loc=4) # 指定lgend的位置

plt.show()

牛顿法

import matplotlib.pyplot as plt

import numpy as np

def get\_diff\_table(X, Y):

""" 得到差商表"""

n = len(X)

A = np.zeros([n, n])

for i in range(0, n):

A[i][0] = Y[i] # 零阶均差

for j in range(1, n):

for i in range(j, n):

A[i][j] = (A[i][j - 1] - A[i - 1][j - 1]) / (X[i] - X[i - j])

return A

def newton\_interpolation(X, Y, xx):

""" 计算x点的插值

1. f(x) = f(x0) + f[x0,x1](x-x0)+…… 牛顿插值函数

2. f[x0,x1] = (f(x0)-f(x1))/(x0-x1) chashang 差商

3. (x-x0)(x-x1)…… x\_diff 牛顿插值中，和差商相乘的那一项

"""

fx = Y[0] # f(x0)

chashang = np.zeros((len(X), len(X))) # 差商表

x\_diff = 1.0

# 将第0列赋值

for j in range(0, len(X)):

chashang[j, 0] = Y[j] # 零阶差商 第零列差商

# 计算插值

for j in range(1, len(X)): # 列

# 计算x的多项式 x\_diff

x\_diff = x\_diff \* (xx - X[j - 1])

# 从第1列到第n列计算差商

for i in range(j, len(X)): # 行

chashang[i, j] = (chashang[i, j - 1] - chashang[i - 1, j - 1]) / (X[i] - X[i - j])

fx = fx + x\_diff \* chashang[j, j]

return fx

def plot\_image2(X, Y, xq, yq, num):

# 绘图

plt.title("Newton\_interpolation") # 打印标题

plt.plot(X, Y, 's', label="original values") # 蓝色点表示原来的值

plt.plot(xq, yq, 'r', label="interpolation values") # 插值曲线

words = "被插点数：{}".format(num)

plt.text(3.5, -10, words, fontproperties='SimHei')

plt.xlabel('x')

plt.ylabel('y')

plt.legend(loc=4) # 指定lgend的位置

plt.show()

分段插值法

import matplotlib.pyplot as plt

import numpy as np

# 分段线性插值，需要两个列表

def segmented\_linear\_interpolate(xlist, ylist, x):

"""

n = # of intervals, which is derived from len of xlist

len of xlist is always one item biger than # of intervals

"""

# we have to make sure that items in xlist is in order

data = dict(zip(xlist, ylist))

# 按照key排序，也就是xlist

data = sorted(data.items(), key=lambda item: item[0])

data = dict(data)

xlist = list(data.keys())

ylist = list(data.values())

n = len(xlist) - 1

if n == 0:

raise ValueError("n should be greater or equal to 1")

# print("segmented interpolate, n =",n)

# 需要把新来的元素判断一下在哪个区间

i = -1

for t in xlist:

if x >= t:

i += 1

if i == -1 or i > len(xlist) - 1:

raise ValueError("x should be between %f and %f" % (xlist[0], xlist[-1]))

if i == len(xlist) - 1:

return ylist[i]

return (x - xlist[i + 1]) / (xlist[i] - xlist[i + 1]) \* ylist[i] + (x - xlist[i]) / (xlist[i + 1] - xlist[i]) \* \

ylist[i + 1]

def plot\_image3(x\_points, y\_points):

# 分段线性插值

fx = lambda t: segmented\_linear\_interpolate(x\_points, y\_points, t)

x = np.linspace(-5, 5, 1000)

y = list(map(fx, x))

# 画图

plt.figure("segmented interpolation")

plt.scatter(x\_points, y\_points, color="orange")

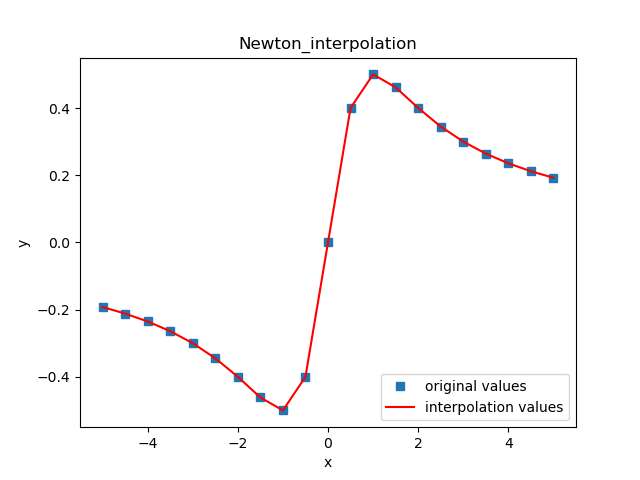
plt.plot(x, y)

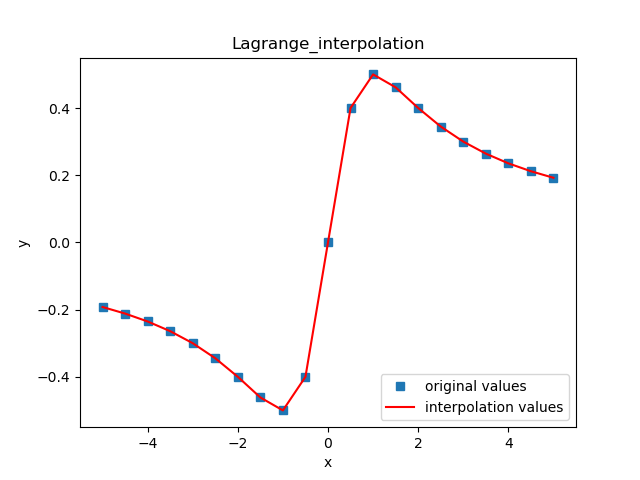
plt.legend(["segmented interpolation", "scattered points"])

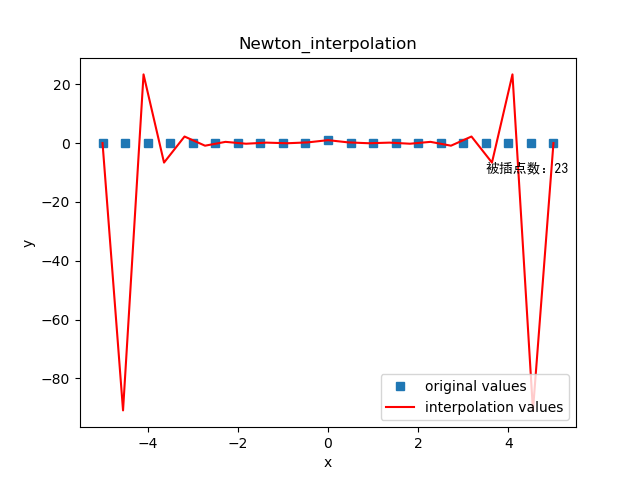
plt.show()

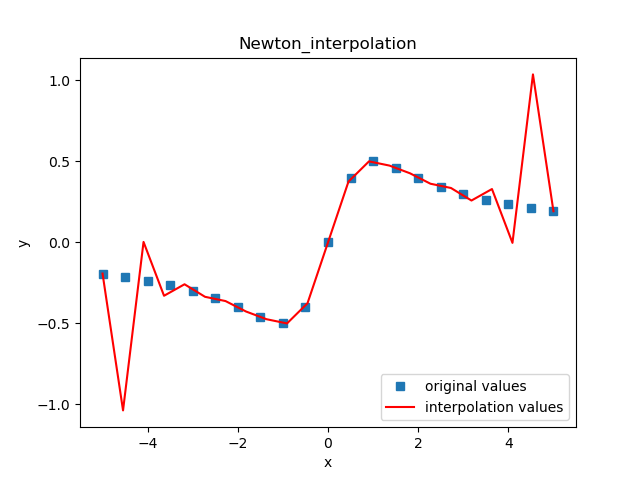
【实验结果分析】

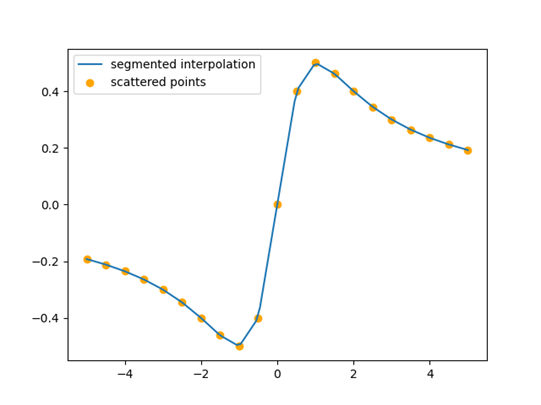
1. 分析对物体1的运动轨迹插值拟合情况
2. 拉格朗日方法和牛顿法

经过调整被插的点数，发现上述两种方法在被插点数为21时，拟合情况最优，如图一所示

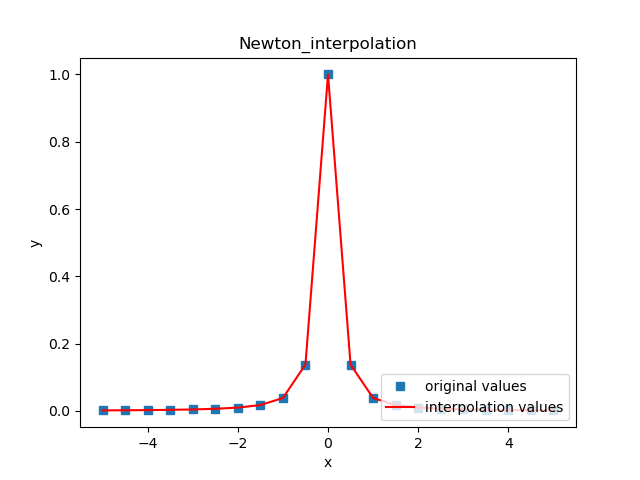
图一 拉格朗日法和牛顿法对物体1运动轨迹拟合

但是，随着插值点个数的增加，插值多项式的次数随之增加，估计值和真实值之间的误差变大，出现了龙格现象，如图二所示。

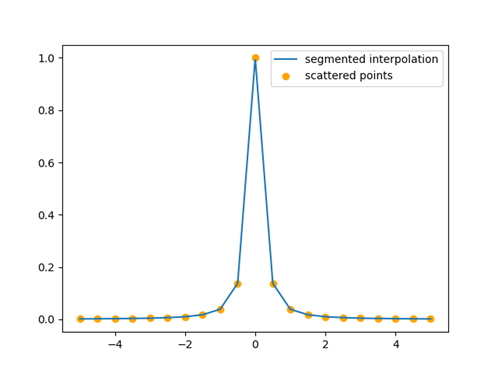
图二 龙格现象产生（此时被插点数为23）

1. 使用分段插值法，将区间[-5,5]分成了1000段，得到了最佳拟合情况，如图三所示

图三 分段插值法

1. 分析对物体2运动轨迹的拟合情况
2. 由于牛顿法和拉格朗日法结果具有等价性，所以只分析牛顿法，被插值数为21时，具有最好的拟合效果，如图四所示，同时当被插值节点过大时，会产生龙格现象

图四 牛顿法拟合结果

1. 使用分段插值法，得到的最优拟合结果
2. 分析x=-3.75和0.25的误差，如下所示

在-3.75处真实值为-0.24896265560165975

拉格朗日方法估计值为-0.38561617038911894

牛顿方法方法估计值为-0.3856161703891152

分段插值方法估计值为-0.24975

在0.25处真实值为0.23529411764705882

拉格朗日方法估计值为0.23003459704029092

牛顿方法方法估计值为0.23003459704029056

分段插值方法估计值为0.2

在-3.75处真实值为0.0028363765289842226

拉格朗日方法估计值为-12.647851822954427

牛顿方法方法估计值为-12.647851822954419

分段插值方法估计值为0.0029

在0.25处真实值为0.3902439024390244

拉格朗日方法估计值为0.7043957943715097

牛顿方法方法估计值为0.7043957943715103

分段插值方法估计值为0.56895

综合上述分析，对于物体1和物体2的运动轨迹函数的拟合，使用牛顿法和分段插值法拟合效果都不错，但是使用牛顿法更好拟合物体一，使用分段插值法更好拟合物体2