# 第7、8章作业(插值、拟合和微分方程)

## 1、任务1

### 1.1 理论建模分析

题中给出了24h内分别在21个不同时间点观测1min内通过桥梁的车辆数目，需要利用这已知的21个1min内车辆通过的数量，估计一天中共有多少量车辆通过这座桥梁。针对这个问题可以利用插值或者拟合的方式估计在没有观测时的1min内车辆通过的数量，如此就可以估计一天中多少量车通过桥梁。

### 1.2 程序与结果

**1）多项式拟合实现**

"""

Created on Sat Nov 11 14:38:48 2023

多项式拟合实现估计

@author: 张启元

"""

import numpy as np

#单位时间内通行车数量

data\_speed = [2, 2, 0, 2, 5, 8, 25, 12, 5, 10, 12, 7, 9, 28, 22, 10, 9, 11, 8, 9, 3]

#时间间隔

time\_gap = [0, 120, 240, 300, 360, 420, 480, 540, 630, 690, 750, 840, 960, 1020, 1080, 1140, 1200, 1260, 1320, 1380, 1440]

#4次多项式拟合 , 得到多项式拟合系数

fit\_4 = np.polyfit(time\_gap, data\_speed, 4)

#创建待拟合函数间隔

new\_time\_gap = range(0, 24\*60)

#拟合值

fit\_4\_value = np.polyval(fit\_4,new\_time\_gap)

#通过车辆总数

total\_num = int(np.sum(fit\_4\_value))

print(total\_num)

运行以上程序后，可以得到运行结果如图1所示，即估计一天内通过桥梁的车辆数目有13736辆车。

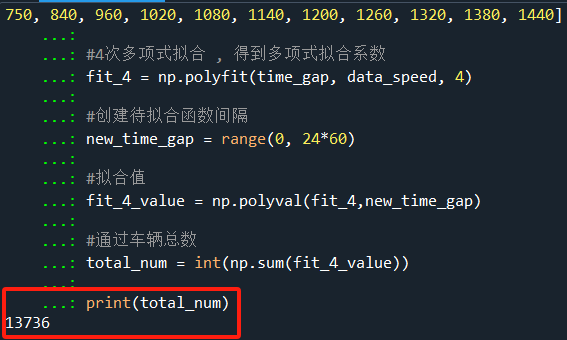


图1 多项式拟合估计结果

**2）线性插值实现**

"""

Created on Sat Nov 11 14:38:48 2023

线性插值实现估计

@author: admin

"""

from scipy.interpolate import interp1d

import numpy as np

#单位时间内通行车数量

data\_speed = [2, 2, 0, 2, 5, 8, 25, 12, 5, 10, 12, 7, 9, 28, 22, 10, 9, 11, 8, 9, 3]

#时间间隔

time\_gap = [0, 120, 240, 300, 360, 420, 480, 540, 630, 690, 750, 840, 960, 1020, 1080, 1140, 1200, 1260, 1320, 1380, 1440]

#线性插值计算插值函数

interpolta\_fuc = interp1d(time\_gap, data\_speed, kind='linear')

#新间隔

new\_time\_gap = range(0,24\*60)

#估计值

estimated\_value = interpolta\_fuc(new\_time\_gap)

#通过车辆总数

total\_num = int(np.sum(estimated\_value))

print(total\_num)

运行以上程序后，可以得到运行结果如图2所示，即估计一天内通过桥梁的车辆数目有12989辆车

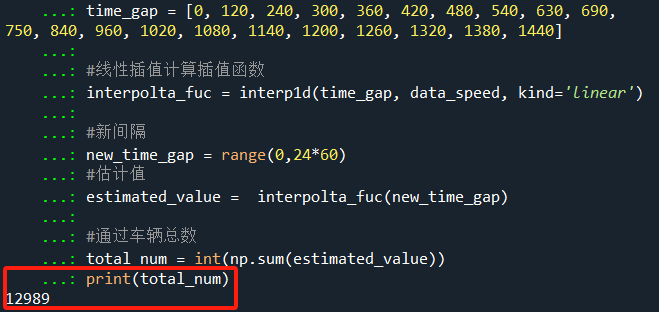


图2 线性插值估计结果

## 2、任务2

### 2.1 理论分析

任务2为求解一个3阶非线性微分方程组的数值解，并绘制出两个因变量在[0, 10]上的数值解曲线。该题可以直接调用 Scipy 库中的 odeint 函数来实现。

### 2.2 程序与运行结果

"""

Created on Sat Nov 11 17:40:34 2023

求解三阶非线性微分方程组，含有两个因变量

@author: admin

"""

import numpy as np

import matplotlib.pyplot as plt

from scipy.integrate import odeint

# 定义含两个因变量的二阶非线性微分方程

def model(y, t):

y1, y2, y3, y4, y5 = y

dy1dt = y2

dy2dt = y3

dy3dt = -3\*y1\*y3+2\*y1\*\*2-y4

dy4dt = y5

dy5dt = -2.1\*y1\*y5

return [dy1dt, dy2dt, dy3dt, dy4dt, dy5dt ]

# 定义初始条件

y0 = [0, 0.0, 0.68, 1, -0.5]

t = np.linspace(0, 10, 1000) # 时间间隔从0到10，将其均匀分成1000个点

# 求解微分方程

y = odeint(model, y0, t)

#创建图表和轴对象

fig, ax1 =plt.subplots()

#绘制第一个数据集（左Y轴）

ax1.plot(t,y[:, 0],'b-',label = 'f(η)')

ax1.set\_xlabel('η')

ax1.set\_ylabel('f(η)',color = 'b')

ax1.tick\_params('y', colors = 'b')

#绘制第二个数据集

ax2 = ax1.twinx()

ax2.plot(t, y[:, 3], 'c', label='T(η)')

ax2.set\_ylabel('T(η)',color = 'c')

ax2.tick\_params('y',colors = 'c')

#添加图例

lines, labels = ax1.get\_legend\_handles\_labels()

lines2, labels2 = ax2.get\_legend\_handles\_labels()

ax2.legend(lines + lines2, labels + labels2, loc='upper center')

plt.savefig('Imag/微分方程组数值解曲线.jpg', dpi=600)

plt.show()

运行程序后，可以得到*f*(*η*)与*T*(*η*)的数值解曲线如图3所示，图中蓝色的为*f*(*η*)的数值解曲线，绿色为*T*(*η*)的数值解曲线。

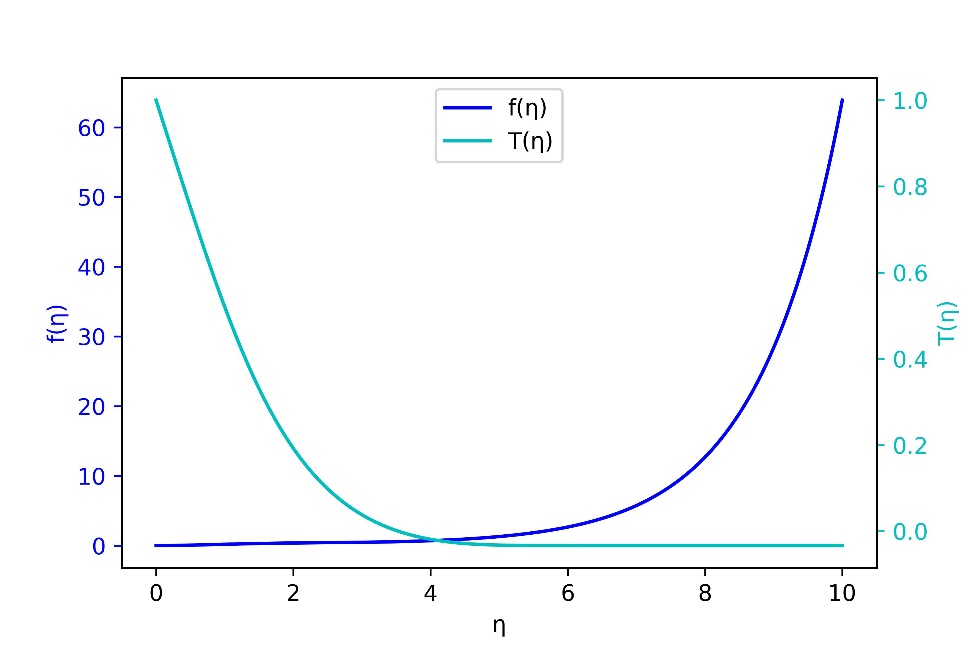


图3 微分方程组数值解曲线