

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。

```

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)
13736

```

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
interpola_fuc = interp1d(time_gap, data_speed, kind='linear')

#新闻隔
new_time_gap = range(0,24*60)
#估计值
estimated_value = interpola_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

```

..... 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
..... interpola_fuc = interp1d(time_gap, data_speed, kind='linear')
.....
..... #新间隔
..... new_time_gap = range(0,24*60)
..... #估计值
..... estimated_value = interpola_fuc(new_time_gap)
.....
..... #通过车辆总数
..... total_num = int(np.sum(estimated_value))
..... print(total_num)
12989

```

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 $f(\eta)$ 与 $T(\eta)$ 的数值解曲线如图 3 所示，图中蓝色的为 $f(\eta)$ 的数值解曲线，绿色为 $T(\eta)$ 的数值解曲线。

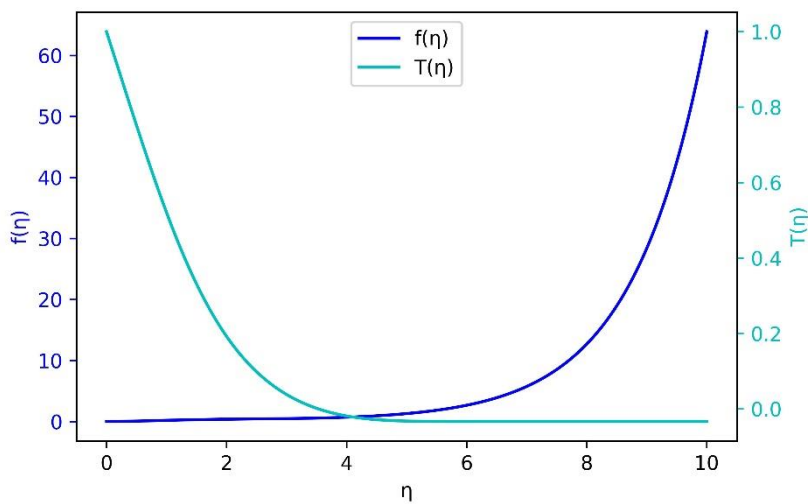


图 3 微分方程组数值解曲线