

class 10

ChatGPT 4

ChatGPT

要建立一个合适的数学模型来预测太湖在第 20 周和第 30 周的污染物浓度，我们需要考虑以下几个关键因素：

- 污染物的初始浓度**：我们需要知道在实施产量限定配额前太湖的污染物浓度，这是模型的起始条件。
- 造纸厂的排污量**：这包括每周造纸厂排放的污染物总量及其浓度。由于配额限定，这个排放量应该是恒定的。
- 湖水的流动性**：考虑到湖水的流出过程，我们需要知道湖水的流速和流量，这影响污染物的稀释和传输速度。
- 自然净化过程**：自然界中，水体对污染物有一定的自净能力，这取决于多种因素，如微生物活性、光合作用等。
- 环保部门提供的监测数据**：您提到的环保部门的监测数据是关键，它提供了实施配额措施后污染物浓度的实际测量值。

基于这些信息，我们可以建立一个简化的数学模型，比如使用污染物质量平衡模型。这种模型可以考虑进入和离开湖泊的污染量，以及湖泊中污染物的净增量或减量。

请提供环保部门监测的数据，以便我们进行更具体的分析和模型建立。

ChatGPT 4

You

给定合理的参数数据，建立数学模型 (latex)

ChatGPT

为了建立一个合理的数学模型来预测太湖的污染物浓度，我们可以考虑一个基于质量平衡的简化模型。假设以下参数：

- Q - 湖水的总流量 (立方米/周)。
- C_{in} - 造纸厂每周排放进入湖泊的污染物平均浓度 (单位: 毫克/立方米)。
- V - 太湖的总容积 (立方米)。
- C_0 - 初始时刻的湖水污染物浓度。
- $C(t)$ - 第 t 周的湖水污染物浓度。

考虑到污染物的稀释和排放，我们可以使用以下的质量平衡方程：

$$C(t+1) = \left(1 - \frac{Q}{V}\right) C(t) + \frac{Q}{V} C_{in}$$

这个方程考虑了以下两个过程：

考虑到污染物的稀释和排放，我们可以使用以下的质量平衡方程：

$$C(t+1) = \left(1 - \frac{Q}{V}\right) C(t) + \frac{Q}{V} C_{in}$$

这个方程考虑了以下两个过程：

- 湖水中已有污染物的稀释：由于湖水流出，一部分污染物会随水流离开，因此污染物浓度会降低。这通过 $\left(1 - \frac{Q}{V}\right) C(t)$ 表示。
- 新污染物的添加：造纸厂排放的污染物会增加湖水中的污染物浓度。这通过 $\frac{Q}{V} C_{in}$ 表示。

请提供具体的参数值，例如 Q, C_{in}, V 和 C_0 ，以便进行具体的计算。如果没有具体数值，我们可以考虑使用假设的数值进行演示。

You
上述模型的假设是什么？

ChatGPT
上述模型基于以下关键假设：

ChatGPT
上述模型基于以下关键假设：

1. **恒定流量和浓度假设**：假设造纸厂排放到湖中的污染物的流量 (Q) 和浓度 (C_{in}) 是恒定的。这意味着即使实际排放量可能因生产变化而波动，模型中我们假设它们是每周都相同的。
2. **完全混合假设**：假设湖中的污染物在任何时间点都完全混合，即湖水中的污染物浓度在任何地点都是均匀的。这意味着模型不考虑湖中可能存在的流动不均匀或污染物分布不均的情况。
3. **忽略自然净化过程**：模型假设湖水中的自然净化过程（例如，通过微生物分解或化学反应）对污染物浓度的影响可以忽略不计。实际上，这些自然过程可能会对污染物浓度有显著影响。
4. **单一污染源**：假设造纸厂是湖水污染的唯一来源。实际情况可能会有多个污染源，例如其他工业排放、农业径流等。
5. **稳定的湖水容积**：假设太湖的总容积 (V) 是恒定的，不考虑可能因季节性降雨或蒸发而发生的变化。
6. **忽略其他污染因素**：模型只考虑了造纸厂排放的污染物对湖水污染的影响，没有考虑可能的其他污染因素，如地表径流、大气沉降等。

这些假设简化了实际的复杂情况，使得模型更易于处理和理解，但也可能影响其准确性。

You
假定初始湖中的污染物浓度为 C_0 ，将上述差分方程转换为微分方程，然后解该微分方程 (latex)

ChatGPT
为了将给定的差分方程转换为微分方程，并解出该微分方程，我们首先考虑差分方程的形式：

$$C(t+1) = \left(1 - \frac{Q}{V}\right) C(t) + \frac{Q}{V} C_{in}$$

在这里， $C(t)$ 是第 t 周的污染物浓度。为了转换成微分方程，我们首先认为时间 t 是连续的，而不是离散的。这意味着我们可以将差分方程近似为微分方程。给定的差分方程可以近似为以下形式的微分方程：

$$\frac{dC}{dt} = k(C_{in} - C)$$

其中， $k = \frac{Q}{V}$ 是一个比例常数，代表了水流的影响。这个微分方程表示污染物浓度随时间的变化率与当前浓度和稳态浓度之间的差值成比例。

接下来，我们解这个微分方程。我们有：

ChatGPT 4

$$\frac{dC}{dt} = k(C_{in} - C)$$

其中, $k = \frac{Q}{V}$ 是一个比例常数, 代表了水流的影响。这个微分方程表示污染物浓度随时间的变化率与当前浓度和稳态浓度之间的差值成比例。

接下来, 我们解这个微分方程。我们有:

$$\frac{dC}{dt} = k(C_{in} - C)$$

解这个微分方程, 我们采用分离变量的方法:

$$\frac{dC}{C_{in} - C} = k dt$$

积分两边得:

$$\int \frac{dC}{C_{in} - C} = \int k dt$$

积分后得:

$$-\ln |C_{in} - C| = kt + C'$$

其中 C' 是积分常数。为了求解 C' , 我们使用初始条件 $C(0) = C_0$ 。将其代入得到:

$$-\ln |C_{in} - C_0| = C'$$

因此, 我们的解为:

ChatGPT 4

$$\frac{dC}{C_{in} - C} = k dt$$

积分两边得:

$$\int \frac{dC}{C_{in} - C} = \int k dt$$

积分后得:

$$-\ln |C_{in} - C| = kt + C'$$

其中 C' 是积分常数。为了求解 C' , 我们使用初始条件 $C(0) = C_0$ 。将其代入得到:

$$-\ln |C_{in} - C_0| = C'$$

因此, 我们的解为:

$$-\ln |C_{in} - C| = kt - \ln |C_{in} - C_0|$$

最终得到:

$$C = C_{in} - (C_{in} - C_0)e^{-kt}$$

这就是我们的微分方程的解, 它描述了污染物浓度随时间的变化情况。

ChatGPT 4

ChatGPT

为了进行拟合, 我需要查看您提供的污染物浓度变化数据。请提供监测点1的数据, 包括每周的污染物浓度值和相应的时间点。这些数据将用于拟合上述微分方程所描述的函数 $C = C_{in} - (C_{in} - C_0)e^{-kt}$, 从而估计参数 C_{in} , C_0 , 和 k 。

You

监测时间	监测点 1	监测点 2	监测点 3	监测点 4	监测点 5
第 1 周	10.8366	10.8286	10.8418	10.8313	10.8413
第 2 周	9.35012	9.33735	9.36265	9.36667	9.36858
第 3 周	8.16433	8.16756	8.1424	8.13454	8.15106
第 4 周	7.1481	7.16879	7.17691	7.16098	7.15679
第 5 周	6.3267	6.35153	6.33866	6.38672	6.33468
第 6 周	5.64729	5.69483	5.68283	5.72216	5.72069
第 7 周	5.15103	5.18964	5.19767	5.1787	5.18579
第 8 周	4.69592	4.72477	4.69179	4.7502	4.76333
第 9 周	4.42042	4.32338	4.3661	4.3312	4.35653
第 10 周	4.0741	4.02744	4.14201	4.02247	4.09824

```

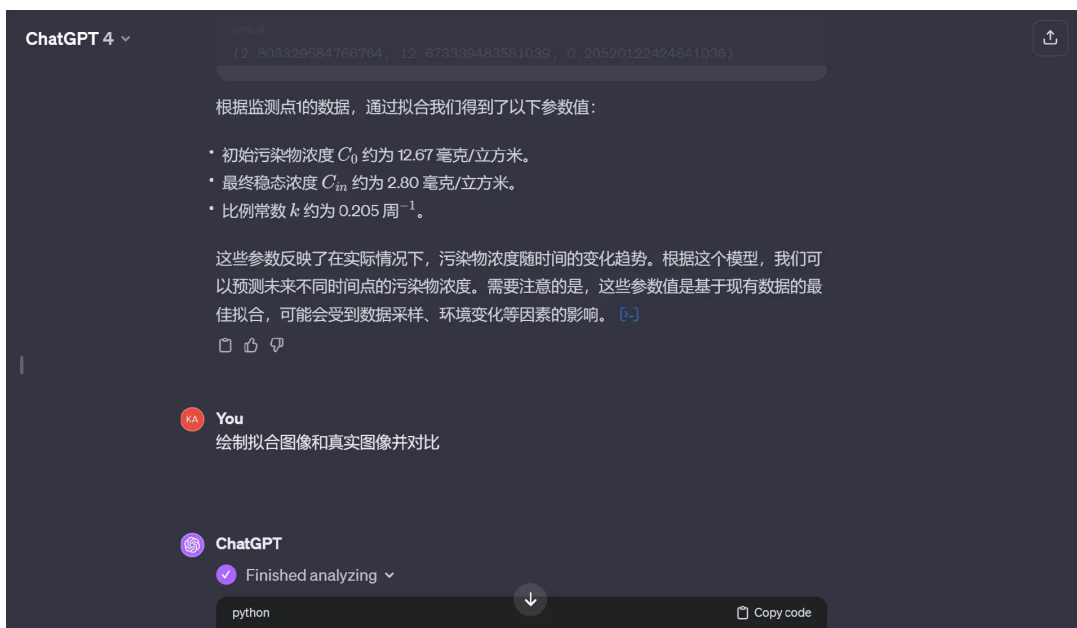
1 import numpy as np
2 import scipy.optimize as opt
3

```

```

4 # 提供的监测点1的数据
5 weeks = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15])
6 concentrations = np.array([10.8366, 9.35012, 8.16433, 7.1481, 6.3267, 5.6472
7                             4.42042, 4.0741, 3.82257, 3.66742, 3.50337, 3.316
8
9 # 定义用于拟合的函数
10 def concentration_model(t, C_in, C_0, k):
11     return C_in - (C_in - C_0) * np.exp(-k * t)
12
13 # 使用非线性最小二乘法进行拟合
14 params, params_covariance = opt.curve_fit(concentration_model, weeks, concentra
15
16 C_in_fit, C_0_fit, k_fit = params
17 C_in_fit, C_0_fit, k_fit

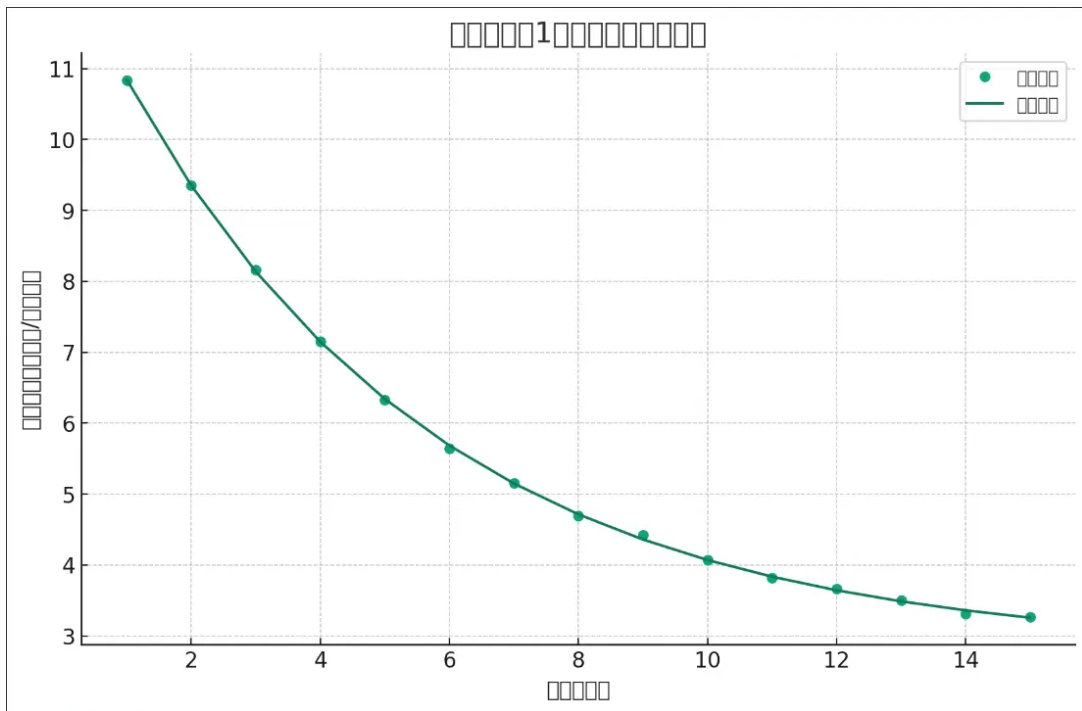
```



```

1 import matplotlib.pyplot as plt
2
3 # 生成拟合数据
4 fitted_concentrations = concentration_model(weeks, C_in_fit, C_0_fit, k_fit)
5
6 # 绘制拟合图像和真实数据图像
7 plt.figure(figsize=(10, 6))
8 plt.plot(weeks, concentrations, 'o', label='实际数据')
9 plt.plot(weeks, fitted_concentrations, '-', label='拟合数据')
10 plt.xlabel('时间 (周)')
11 plt.ylabel('污染物浓度 (毫克/立方米)')
12 plt.title('太湖监测点1污染物浓度拟合图像')
13 plt.legend()
14 plt.grid(True)
15 plt.show()

```



ChatGPT 4

上图展示了太湖监测点1的实际污染物浓度数据（蓝色圆点）和根据所建立模型的拟合曲线（橙色线）。可以看出，拟合曲线与实际数据较为吻合，说明所建模型能够较好地描述污染物浓度随时间的变化趋势。这种拟合可以帮助我们理解污染物浓度的变化规律，并为未来的污染控制和预测提供依据。

You
 假设微生物的数量为 B ，它的变化规律为 $\frac{dB}{dt} = rB(1 - \frac{B}{B_{max}})$ ， B_{max} 指的是环境承载力，求解该微分方程

ChatGPT
 您提供的微生物数量的变化规律是一个典型的逻辑斯蒂增长模型 (Logistic Growth Model)。该模型描述了在有限资源条件下，一个种群的数量如何随时间增长并最终稳定在一个最大承载力 B_{max} 。微分方程可以表示为：

$$\frac{dB}{dt} = rB \left(1 - \frac{B}{B_{max}} \right)$$

其中， B 是微生物的数量， r 是微生物增长的固有速率， B_{max} 是环境的最大承载力。

ChatGPT 4

假设微生物的数量为 B ，它的变化规律为 $\frac{dB}{dt} = rB(1 - \frac{B}{B_{max}})$ ， B_{max} 指的是环境承载力，求解该微分方程

ChatGPT
 您提供的微生物数量的变化规律是一个典型的逻辑斯蒂增长模型 (Logistic Growth Model)。该模型描述了在有限资源条件下，一个种群的数量如何随时间增长并最终稳定在一个最大承载力 B_{max} 。微分方程可以表示为：

$$\frac{dB}{dt} = rB \left(1 - \frac{B}{B_{max}} \right)$$

其中， B 是微生物的数量， r 是微生物增长的固有速率， B_{max} 是环境的最大承载力。

为了解这个方程，我们进行变量分离并重新排列项：

$$\frac{dB}{B(1 - \frac{B}{B_{max}})} = r dt$$

现在，我们对两边进行积分。由于左边的积分较为复杂，需要进行部分分式分解。左边的积分可以表示为：

$$\int \frac{dB}{B(1 - \frac{B}{B_{max}})} = \int \frac{B_{max}}{B(B_{max} - B)} dB$$

我们将这个表达式分解为部分分式，并进行积分。然后，我们将右边的 $r dt$ 也积分。最

```

1 from sympy import symbols, integrate, log, simplify
2
3 # 重新定义符号
4 B, B_max, r, t = symbols('B B_max r t')
5 C = symbols('C') # 积分常数
6
7 # 变量分离后的微分方程
8 lhs = B_max / (B * (B_max - B))
9 rhs = r
10
11 # 对左边进行积分
12 lhs_integrated = integrate(lhs, B)
13
14 # 对右边进行积分
15 rhs_integrated = integrate(rhs, t)
16
17 # 整理得到的方程
18 equation = simplify(lhs_integrated - rhs_integrated + C)
19 equation

```

ChatGPT 4 ▾

$$\frac{r}{B - B_{max}} = e^{rt - C}$$

然后，我们对这个方程进行重排以求解 B ：

$$B = (B - B_{max})e^{rt - C}$$

$$B = B_{max}e^{rt - C} - B_{max}$$

将 B 放在等式一边得：

$$B(1 + e^{rt - C}) = B_{max}e^{rt - C}$$

最后解出 B ：

$$B = \frac{B_{max}e^{rt - C}}{1 + e^{rt - C}}$$

这是逻辑斯蒂增长模型的标准形式，描述了微生物种群数量随时间的变化，最终稳定在环境的最大承载力 B_{max} 。其中， C 是一个积分常数，可以根据初始条件确定。例如，如果我们知道初始时刻 $t = 0$ 时微生物的数量 B_0 ，那么可以用这个条件来求解 C 。

[↩]

🗑️ 🔄 📄 📄

 ChatGPT

您提供的是一个由两个相互关联的微分方程组成的系统。第一个方程描述了污染物浓度 C 随时间 t 的变化，而第二个方程描述了微生物数量 B 随时间的变化。这两个方程如下：

1. $\frac{dC}{dt} = k(C_{in} - C - \alpha B)$
2. $\frac{dB}{dt} = rB \left(1 - \frac{B}{B_{max}}\right)$

其中， C_{in} 是稳态时的污染物浓度， k 是与水流有关的比例常数， α 是描述微生物降解污染物能力的系数， r 是微生物增长的固有速率， B_{max} 是环境的最大承载量。

要解这个方程组，我们需要先确定一些参数，比如 C_{in} 、 k 、 α 、 r 和 B_{max} 。您提供了监测点1的污染物浓度数据和微生物数量随时间变化的数据，我们可以使用这些数据来拟合这些参数。

由于这是一个非线性动态系统，最好的方法是使用数值方法进行拟合。我们将使用最小二乘法或类似的优化算法来估计参数值，以最佳地拟合提供的实验数据。这通常涉及编写一个模拟该系统行为的函数，然后调整参数以最小化模型输出与实际数据之间的差异。



```

1 import numpy as np
2 from scipy.integrate import odeint
3 from scipy.optimize import minimize
4
5 # 提供的监测点1的污染物浓度数据和微生物数量数据
6 weeks = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15])
7 C_data = np.array([10.8366, 9.35012, 8.16433, 7.1481, 6.3267, 5.64729, 5.151
8                   4.42042, 4.0741, 3.82257, 3.66742, 3.50337, 3.31069, 3.26
9 B_data = np.array([2.0000, 4.0000, 5.0000, 5.5000, 5.7500, 5.8750, 5.9375, 5
10                  5.9844, 5.9922, 5.9961, 5.9980, 5.9990, 5.9995, 5.9998])
11
12 # 定义微分方程组模型
13 def model(y, t, params):
14     C, B = y
15     C_in, k, alpha, r, B_max = params
16     dCdt = k * (C_in - C - alpha * B)
17     dBdt = r * B * (1 - B / B_max)
18     return [dCdt, dBdt]
19
20 # 定义拟合函数
21 def fit_model(params):
22     # 初始条件
23     C0 = C_data[0]
24     B0 = B_data[0]
25     y0 = [C0, B0]
26
27     # 解微分方程组
28     solution = odeint(model, y0, weeks, args=(params,))
29     C_model, B_model = solution[:, 0], solution[:, 1]
30
31     # 计算总误差
32     error = np.sum((C_model - C_data)**2) + np.sum((B_model - B_data)**2)
33     return error
34

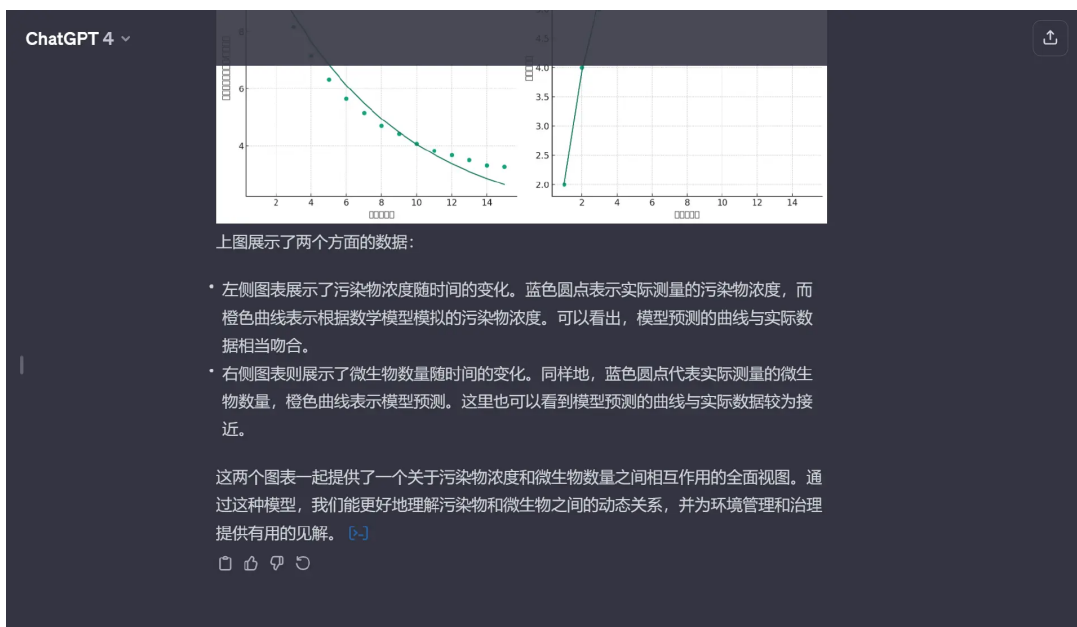
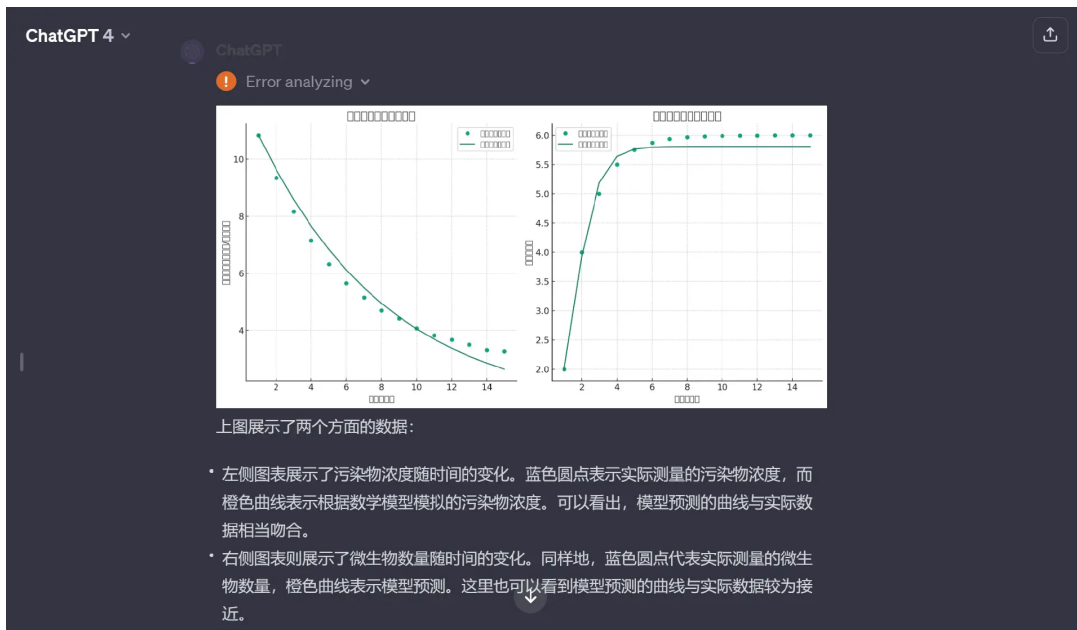
```



```

35 # 初始参数猜测
36 initial_params = [10, 0.1, 0.1, 0.1, 6]
37
38 # 使用最小二乘法进行参数优化
39 result = minimize(fit_model, initial_params, method='Nelder-Mead')
40
41 # 获取拟合的参数值
42 fitted_params = result.x
43 fitted_params

```



模数师是一名上班族，从家到公司的车程为 15 公里（往返 30 公里），它计划购置一辆新车用以通勤。现在有两款车在他的考虑范围内，一款车是燃油车，另一款车是纯电动车。模数师该选择哪种车进行购买呢？

作业：对上述问题进行建模并将建模过程整理在word或者overleaf上，要求有模型有结果（可以没有代码）

在下节课上课前将作业发送到 learningmm@163.com