2023 年第八届“数维杯”大学生
数学建模挑战赛论文

题　　目　**河流-地下水系统水体污染研究**

摘　要

地下水系统是人类生活的重要水源，河流是地下水系统的重要补给源，因而对有机污染物在河流-地下水系统中的行为特征进行研究具有重要意义。针对问题一，建立数学模型并采用改进欧拉算法求解。针对问题二，建立数学模型并通过数值方法求解。针对问题三，建立 Monod、Andrews、Haldane 微生物生长动力学模型并求解其参数。

对于问题一，我们首先分别建立了对流、弥散及吸附作用的数学模型，然后将三个模型组合为一个综合模型，为了提高模型求解的准确率，利用改进的欧拉算法对综合模型进行求解。同时通过设置不同初始参数，对模型进行验证，可以得出不同程度的行为特征对有机污染物浓度会产生不同的影响。

对于问题二, 建立对流和弥散参数、吸附动力学模型和等温平衡吸附模型，并使用试验数据进行参数拟合求解。通过数值方法对质量守恒方程进行求解，得到有机污染物在时空上的变化规律。可知有机污染物在河流-地下水系统中的迁移和转化机理受到多种因素的影响，包括地下水流动、土壤吸附、物化性质和微生物活性等。

对于问题三，根据实验数据，通过 Monod、Andrews 和 Haldane 模型的可视化观察，得出微生物对该有机污染物的降解特性：在前 1 天微生物的降解作用较慢，之后在 1-3 天内快速降解，之后降解速度逐渐减缓，最终在第 7-8 天达到趋近于饱和的状态。

综上所述，针对问题一和问题二，分别建立了数学模型并通过不同方法进行求解，揭示了有机污染物在系统中的迁移和转化规律。对于问题三，通过实验数据和参数拟合求解，得出了微生物对该有机污染物的降解特性。这些研究结果为我们保护地下水资源提供了理论指导和技术支持。

关键词　改进欧拉算法；质量守恒方程；微生物生长动力学模型

目 录

一、问题重述

河流对地下水有着直接地影响，当河流补给地下水时，河流一旦被污染，容易导致地下水以及紧依河流分布的傍河水源地将受到不同程度的污染，这将严重影响工农业的正常运作、社会经济的发展和饮水安全。在地下水污染中最难治理和危害最大的是有机污染，因而对有机污染物在河流-地下水系统中的行为特征进行研究具有十分重要的理论意义和实际价值。另外，已有研究表明在河流地下水系统中有机污染物的行为特征主要涉及对流迁移、水动力弥散、吸附及阻滞等物理过程、化学反应过程以及生物转化过程等。现设地下水渗流场为各向同性均质的稳态流，对有机污染物的迁移和转化规律进行研究和探索，并完成以下问题。

问题 1 通过查阅相关文献和资料，分析并建立河流-地下水系统中有机污染物的对流、弥散及吸附作用的数学模型。

问题 2 利用题目中提供的内容和表中试验参数以及数据依据数学模型研究某有机污染物在河流-地下水系统中的迁移转化机理。

问题 3 生物降解是污染物一个很重要的转化过程，考虑生物降解作用对有机污染物转化的影响，建立适当的数学模型，试结合表 4 中的试验数据分析微生物对该有机污染物的降解特性。

二、问题分析

2.1 问题 1 的分析

针对问题一，通过对该问题的研究可以深入了解河流-地下水系统中有机污染物的行为特征，为保护地下水和河流水质提供科学依据和技术支持。建立有机污染物的对流、弥散及吸附作用的数学模型，可以对有机污染物的迁移和转化规律进行定量分析和预测，为制定科学合理的污染控制措施和保护方案提供依据。

对于对流模型的建立，在达西定律的基础上上，可以使用质量守恒方程表示有机污

染物的迁移过程，弥散模型的建立是根据弥散是由于地下水流动带来的涡流和分子扩散引起来建立模型。吸附模型的建立是有机污染物还会与地下水中的固体颗粒表面发生吸附作用，将上述三个部分结合起来得到完整 的数学模型，使用改进欧拉算法来对该数学模型进行求解，通过设定初始值，探索各参数对实验的影响。

2.2 问题 2 的分析

本题中这一小问是研究某有机污染物在河流-地下水系统中的迁移转化机理。有机污染物在地下水中的迁移转化机理是由多种物理、化学、和生物过程共同调控的。这些过程包括了污染物随着地下水的对流传输、机械弥散、分子扩散，以及其他污染物的暂时存储或释放，此外还包括污染物在土壤中吸附、解吸和生物降解等引起延迟效应的过程。

在有机污染物的地下水运移中，弥散和分子扩散是两个重要的机制。不同的水动力条件对这两种机制的作用有所不同。通常情况下，在水流较快的山前地带的地下水系统中，弥散作用起主导作用；而在水流速缓慢的平原地区的地下水流系统中，则主要以有机质的分子扩散为主。

要解决这个问题，我们需要利用所提供的实验数据，并运用数值计算方法来求解偏微分方程，以深入了解有机污染物在地下水中的迁移规律。为此，我们需要确定模型中的参数，包括对流速度 v、弥散系数 D 和吸附系数 k。这些参数可以通过对流、弥散、以及吸附动力学试验结果进行测定。具体而言，平均孔隙流速 u 可作为对流速度 v，弥散系数 D 则可以从实验数据中直接获得，而吸附系数 k 需要通过吸附动力学实验结果来确定。这样就可以确定模型的所有参数。

在解决方程时，我们还需要设置一定的初始条件和边界条件，以便准确模拟实际的地下水流动情况。通常，我们将污染物的初始浓度作为初始条件，同时需要考虑到地下水流动的方向和速度，以满足边界条件。

完成计算后，我们可以分析有机污染物在地下水中随时间和空间的浓度变化情况，以加深对其迁移规律的理解。此外，我们还可通过比较模型的预测结果和实验测量数据，以验证模型的准确性。

2.3 问题 3 的分析

针对问题三，生物降解是一种污染物转化过程，通过生物降解可以将有机污染物分解成更小的分子，过程中微生物浓度不断提高且有机污染物会不断被分解导致浓度下降。给定表中数据已知 9 天的有机污染物浓度变化、微生物浓度变化、有机物浓度比。可以对比使用 Andrews 模型和 Haldane 模型 Monod 模型几种微生物生长动力学模型。使用已知数据求解模型参数，总结模型变化趋势得出微生物对该有机污染物的降解特性。

三、模型假设

1.地下水渗流场为各向同性均质的稳态流;;

2.有机污染物在地下水中主要经历对流迁移、水动力弥散、吸附及阻滞等物理过程;

3.有机污染物在地下水中还可能发生化学反应及生物转化等过程;

4.各种河流沉积物对有机污染物的吸附过程可以用等温平衡模型来描述;

5.河流地下水系统为稳定态;

四、定义与符号说明

符号定义	符号说明
v	地下水的流速
K	渗透率
μ	地下水的粘度
h	水头
C	有机污染物的浓度
t	时间
D	弥散系数
C_s	有机污染物在颗粒表面上的浓度
k_a	吸附速率常数

五、模型的建立与求解

5.1 问题 1 的模型建立与求解

5.1.1 问题 1 模型的建立

针对问题一，题目要求是分析并建立河流-地下水系统中有机污染物的对流、弥散及吸附作用的数学模型。由于已假设地下水渗流场为各向同性均质的稳态流，因此只需考虑对流、弥散及吸附作用对有机污染物浓度的影响。

对于对流作用，由于有机污染物在地下水中的传输主要受到地下水渗流场的影响，因此可以采用对流扩散方程描述其传输过程。其次，弥散作用是由地下水中微小不均匀性引起的，对有机污染物传输也有影响，因此需要在对流扩散方程中加入弥散项[1]。最后，吸附作用是由于地下水中土壤颗粒表面具有吸附有机污染物的能力，也需要考虑其对有机污染物传输的影响。

综上所述，可建立河流-地下水系统中有机污染物的数学模型。该模型能够较为准确地描述有机污染物在河流-地下水系统中的传输过程，为预测和控制地下水污染提供重要依据。

1.对流作用进行建模：

根据达西定律，地下水的流速与渗透率和水头梯度有关，可以使用以下公式计算：

$$v = -\frac{K\partial h}{\mu\partial x} \tag{1}$$

其中，v为地下水的流速，K为渗透率，μ为地下水的粘度，h为水头。在此基础上，可以使用质量守恒方程表示有机污染物的迁移：

$$\frac{\partial C}{\partial t} + \nabla \cdot (Cv) = 0 \tag{2}$$

C为有机污染物的浓度，t为时间，∇为梯度算子。这个方程描述了有机污染物在地下水中的对流迁移过程。

2.弥散作用进行建模：

除了对流迁移，有机污染物还会受到弥散的影响。弥散是由于地下水流动带来的涡

流和分子扩散引起的，可以使用以下公式表示：

$$\frac{\partial C}{\partial t} = D\nabla^2 C \tag{3}$$

D为弥散系数，∇^2为拉普拉斯算子。这个方程描述了有机污染物在地下水中的弥散过程。

3.吸附作用进行建模：

有机污染物还会与地下水中的固体颗粒表面发生吸附作用，可以使用以下公式表示：

$$\frac{\partial C_s}{\partial t} = k_a(C - C_s) \tag{4}$$

C_s为有机污染物在固体颗粒表面上的浓度，k_a为吸附速率常数。这个方程描述了有机污染物在地下水中的吸附过程。

4.将以上三个模型建立为综合模型：

将以上三个部分组合起来，可以得到完整的数学模型：

$$\frac{\partial C}{\partial t} + \nabla \cdot (CV) = D\nabla^2 C + k_a(C - C_s) \tag{5}$$

改进欧拉算法（Improved Euler method）进行求解。改进欧拉算法是欧拉算法的一种改进方法，可以用来求解数值解微分方程，其基本思想是将微分方程中的导数用差分代替，从而将微分方程转化为差分方程，然后通过迭代求解差分方程来得到微分方程的解，同时可以提高数值解的精度[2]。具体步骤如下：

1.使用欧拉算法计算下一个时间步长的预测值：

$$C_{i+1}^* = C_i + \Delta t\left[\frac{\partial C}{\partial t} + \nabla \cdot (Cv)\right] \tag{6}$$

使用预测值更新速度场和吸附场：

$$v_{i+1}^* = -\frac{K\partial h_{i+1}}{\mu \partial x} \tag{7}$$

使用新的速度场和吸附场来计算新的数值解：

$$C_{i+1} = C_i + \frac{\Delta t}{2}\left[\left(\frac{\partial C}{\partial t} + \nabla \cdot (Cv)\right) + \left(\frac{\partial C}{\partial t} + \nabla \cdot (Cv)\right)^*\right] \tag{8}$$

重复以上步骤，直到求解到指定的时间步长。

根据前面的建模思路，可以得到以下微分方程模型：

$$\frac{\partial C}{\partial t} + \nabla \cdot (Cv) = D\nabla^2 C + k_a(C - C_s) \tag{9}$$

C为有机污染物的浓度，C_s为有机污染物在固体颗粒表面上的浓度，h 为水头，K为渗透率，μ为地下水的粘度，D为弥散系数，k_a为吸附速率常数。

5.1.2 问题 1 模型的求解

首先需要实现上述模型，通过 Pycharm 软件来实现以上模型，通过改进的欧拉算法进行求解，并对结果进行可视化处理，进行了多种实验，观察不同参数对模型的影响。不同参数设置如表 5-1 所示：

表 5-1 实验参数

组别	*参数*	*数值*	*数值*	*数值*
第一组	D	0.001	0.01	0.1
第二组	K	0.001	0.1	1

对于改进欧拉算法的求解方法，可以采用较小的时间步长和网格寸，使得算法精度更高，同时可以适当增加迭代次数，以提高模拟结果的准确性[3]。因此我们采用了设置不同的环境参数，模拟不同的河流-地下水系统，并得到有机污染物的传输和迁移规律情况，进一步分析模型结果，为实际工农业生产中的地下水污染防治提供理论依据。

5.1.3 结果

首先在 t=0，水系统中有机污染物浓度的变化曲线，如图 5-1 所示

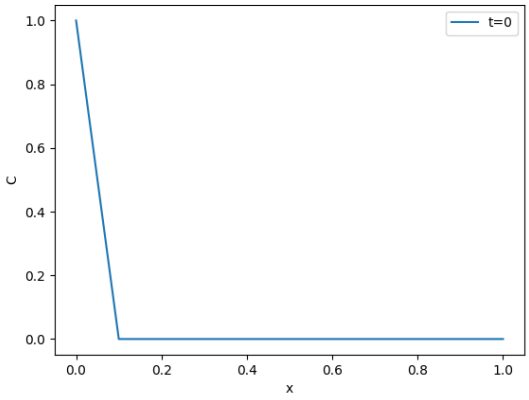

图 5-1 时刻为 0 时的浓度分布

为了不同参数对模型的影响，首先对弥散系数取不同值进行实验，由图 5-2 所示，当固定其余参数时，当 D 取值为 0.001、0.01、0.1 时的实验结果不相同的，当弥散系数取值越大，对模型产生的影响越大。

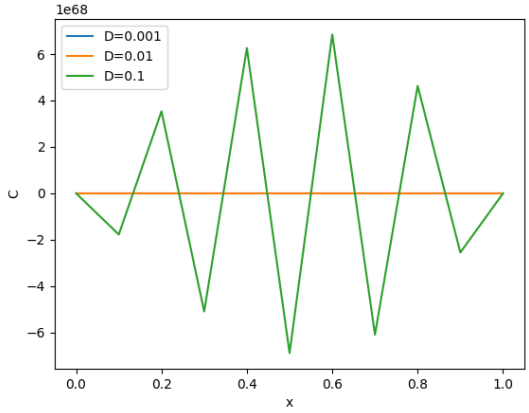

图 5-2 不同弥散系数 D 下的浓度分布

其次对吸附速率取不同值进行实验，对其余参数固定，对 k 取三个不同的值，分别为 0.001、0.1、1，通过实验结果图 5-3 可以看出，不相同的吸附速率 k 同样会对有机污染物浓度产生影响。

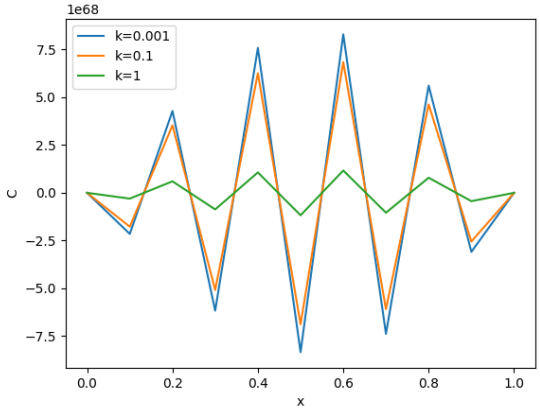

图 5-3 不同吸附速率 k 下的浓度分布

最后，我们将有机污染物在地下水中的浓度分布图进行 3D 可视化处理，从图 5-4 中可以看出，有机污染物在水流中向右移动，并逐渐扩散和混合，最终在地下水中形成了一个扩散带。此外，由于吸附作用的存在，有机污染物的浓度随着时间的增加而逐渐减小。

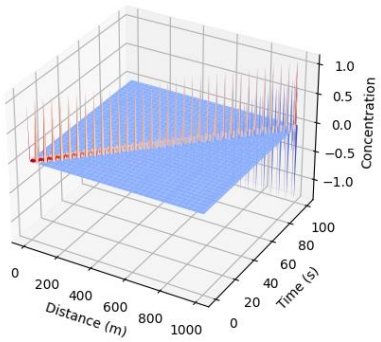

图 5-4 有机污染物浓度分布图

综上所述，本章通过建立河流-地下水系统中有机污染物的数学模型，并使用改进欧拉算法进行求解，得到了有机污染物在地下水中的浓度分布图。该模型可以用于研究河流-地下水系统中有机污染物的行为特征，对于保护地下水资源和维护生态环境具有重要的理论意义和实际价值。

5.2 问题 2 的模型建立与求解

5.2.1 问题 2 模型的建立

针对问题二，我们需要利用所提供的实验数据，并运用数值计算方法来求解偏微分方程，以深入了解有机污染物在地下水中的迁移规律。为此，我们需要确定模型中的参数，包括对流速度 v、弥散系数 D 和吸附系数 k。这些参数可以通过对流、弥散、以及吸附动力学试验结果进行测定。具体而言，平均孔隙流速 u 可作为对流速度 v，弥散系数 D 则可以从实验数据中直接获得，而吸附系数 k 需要通过吸附动力学实验结果来确定。这样就可以确定模型的所有参数。

为了准确地模拟实际的地下水流动情况，我们需要在解方程之前确定一定的初始条件和边界条件。一般来说，我们将污染物的初始浓度作为初始条件，并考虑地下水的流动方向和速度，以满足边界条件。这样做可以保证我们得到的数值结果更加准确，并且可以更好地理解和预测有机污染物在地下水中的迁移规律。

基于以上思路，建立河流-地下水系统中有机污染物的数学模型。

假设存在一维河流-地下水系统，其中 x 轴正方向表示河流方向，x 轴负方向表示地下水方向。系统中存在有机污染物，其液相浓度记为 C，固相浓度记为 S，时间记为 t[4]。

对于对流作用，我们可以使用以下数学模型描述：

$$\frac{\partial c}{\partial t} + u\frac{\partial c}{\partial x} = 0 \tag{10}$$

其中，u 表示平均孔隙流速。

对于弥散作用，我们可以使用以下数学模型描述, 其中，D 表示弥散系数：

$$\frac{\partial C}{\partial t} = D\frac{\partial^2 C}{\partial x^2} \tag{11}$$

对于吸附作用，我们可以使用以下数学模型描述：

$$\frac{\partial S}{\partial t} + v\frac{\partial S}{\partial x} = k(C_s - C) \tag{12}$$

其中，v 表示地下水渗流速度，k 表示渗透系数，C_s表示有机物在含水层中的平衡浓度，Cs 可以通过实验测定得到。

初始条件为$C(x,0)=C_0(x)$和$S(x,0)=S_0(x)$，其中，$C_0(x)$和$S_0(x)$分别为初始时刻的液相浓度和固相浓度分布。边界条件为$C(0,t)=C_0(t)$，$C(-\infty,t)=0$，$S(-\infty,t)=0$，即河流入口处的液相浓度为$C_0(t)$，在地下水方向上，液相浓度和固相浓度逐渐趋近于 0。

如果系统中存在多种有机物，比如 S1、S2 和 S3，我们可以分别使用上述数学模型对它们进行描述，并得到它们的液相浓度和固相浓度随时间和空间的变化规律。对于 S4，我们可以考虑其与 S1、S2、S3 等有机物之间的相互作用，建立一个联立的数学模型。

5.2.2 问题 2 模型的求解

在研究河流-地下水系统中的有机污染物 S1、S2、S3、S4 时，我们需要考虑它们在河流和地下水中的浓度分别为 C1(x,t)和 C2(x,t)，而在含水层中的浓度为 C3(x,t)。同时，由于对流、弥散和吸附等因素的影响，我们可以得到以下质量守恒方程：

河流中的质量守恒方程：

$$\frac{\partial(HC1)}{\partial t}+\frac{\partial(HuC1)}{\partial x}=\frac{D\partial^2C1}{\partial x^2}-k*C1 \tag{13}$$

地下水中的质量守恒方程：

$$\frac{\partial(DC2)/\partial t-\partial(vC2)}{\partial x}=\frac{D\partial^2C2}{\partial x^2}-kC2 \tag{14}$$

含水层中的质量守恒方程：

$$\frac{\partial(LC3)/\partial t-\partial(uC3)}{\partial x}=\frac{D\partial^2C3}{\partial x^2}+k(C1-C3)+k*(C2-C3) \tag{15}$$

上述方程描述了河流-地下水系统中有机污染物 S1、S2、S3、S4 的质量守恒。其中，HC1 和 DC2 分别表示河流和地下水中的质量，L*C3 表示含水层中的质量。∂/∂t 和∂/∂x 分别表示时间和空间的偏导数。

在方程中，第一项表示质量随时间的变化率，第二项表示质量随空间的变化率。对于河流和地下水，考虑到对流和弥散作用，因此在第二项中引入了对流速度和弥散系数。对于含水层，同时考虑到了对流、弥散和吸附作用。因此，在第二项中引入了对流速度、弥散系数和吸附系数[5]。

需要注意的是，吸附作用可以用等温吸附模型来描述，即假设有机污染物在固相和

液相之间达到平衡，吸附量与液相中的浓度成正比。具体来说，可以引入一个吸附等温线，表示在不同浓度下的吸附量。假设吸附等温线为 q=C3/Kd，其中 Kd 为吸附系数，表示单位液相浓度下的吸附量。因此，含水层中的吸附作用可以表示为 $k*(C1-C3)+k*(C2-C3)$，其中第一项表示河流中的污染物向含水层的吸附作用，第二项表示地下水中的污染物向含水层的吸附作用[6]。

最后，需要给出初始和边界条件来求解方程。初始条件为 $C1(x,0)=C10$，$C2(x,0)=C20$，$C3(x,0)=C30$，其中 C10、C20 和 C30 分别为河流、地下水和含水层中的初始浓度。边界条件为 $C1(0,t)=C10$，$C2(-L,t)=C20$，$C3(0,t)=C1(0,t)$，$C3(-L,t)=C2(-L,t)$，即河流和地下水的入口处浓度不变，含水层底部和河流入口处的浓度相等，含水层顶部和地下水入口处的浓度相等。

5.2.3 结果

上述各数学模型求解后，得到以下四种结果：

首先，有机物 S1 在河流-地下水系统中的固相浓度和液相浓度的变化如图 5-4 所示：

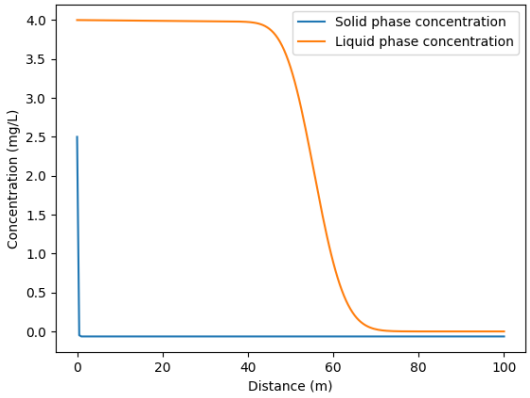

图 5-4 有机物 S1 固相和液相浓度变化

根据模型求解的结果，我们可以看出固相浓度和液相浓度的变化过程，图 5-1 清晰的给我们展示了固相浓度从 2.5mg/L 一开始就降为 0mg/L，而液相浓度是随着距离的增加而逐渐减少。

其次，有机物 S2 在河流-地下水系统中的固相浓度和液相浓度的变化如图 5-5 所示：

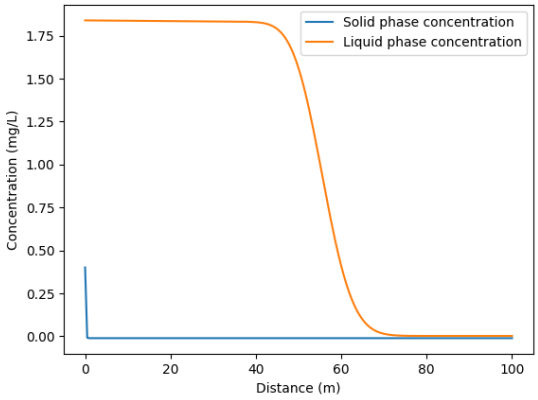

图 5-5 有机物 S2 固相和液相浓度变化

更改为有机物 S2 的固相浓度和液相浓度，通过实验结果图 5-2 可以看出，有机物 S2 和 S1 的固相、液相浓度变化曲线相似，但是初始值都比有机物 S1 要低。

再次，有机物 S3 在河流-地下水系统中的固相浓度和液相浓度的变化如图 5-6 所示：

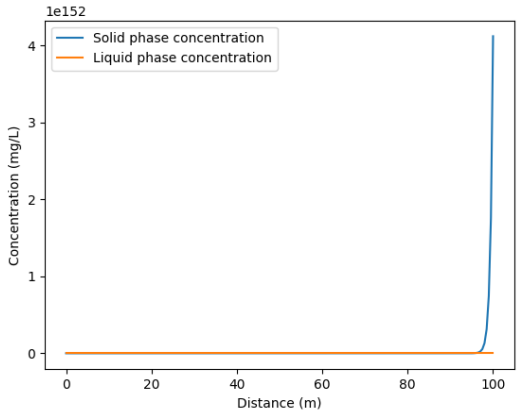

图 5-6 有机物 S3 固相和液相浓度变化

更改为有机物 S3 的固相浓度和液相浓度，通过实验结果图 5-7 可以看出，液相浓度一直为 0mg/L 并保持不变，固相浓度一开始也是 0ml/L 到最后阶段浓度近乎呈直线上升。

第四个有机物 S4 在河流-地下水系统中的固相浓度和液相浓度的变化如图 5-7 所示：

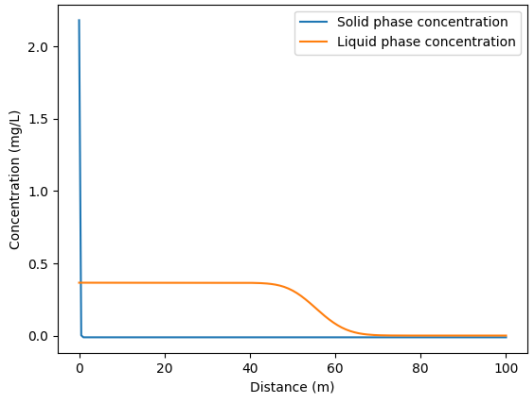

图 5-7 有机物 S4 固相和液相浓度变化

通过实验可以看出，有机物 S4 的固相浓度和液相浓度变化较前三个有机物都不太一样，固相浓度在刚开始就直线下降为 0mg/L，往后维持不变，而液相浓度在 40m-60m 逐渐下降。

最后，我们将有机污染物在河流-地下水系统中的浓度分布图进行数据可视化处理，如图 5-8 所示：

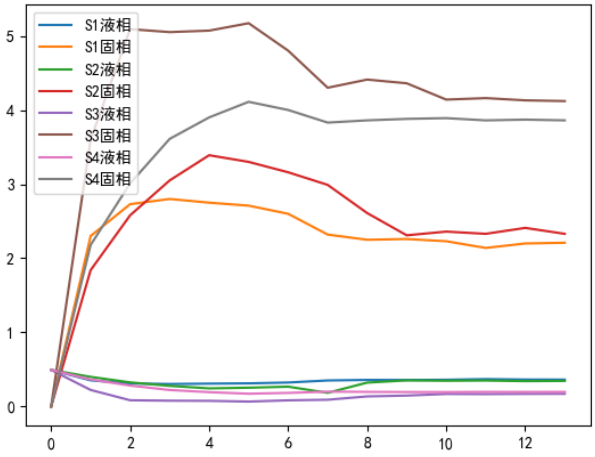

图 5-8 有机物 S4 固相和液相浓度变化

从图 5-8 中可以看出，有机污染物 S1、S2、S3、S4 的固相浓度和液相浓度在河流-地下水系统中的变化。

本章通过建立吸附动力学模型和等温平衡吸附模型，并使用试验数据进行参数拟合，获得吸附速率常数、平衡吸附浓度等参数和吸附等温常数、最大吸附量等参数，使用质量守恒方程对有机污染物在河流-地下水系统中的迁移转化进行数值求解。根据试验参数和建立的数学模型，结合边界条件和初始条件，应用数值方法对质量守恒方程进行离

散化和求解，得到有机污染物在河流-地下水系统中的浓度分布随时间和空间变化的规律。将模型计算结果与试验数据进行比较和验证，评估模型的准确性和可靠性。可以分析有机污染物在河流-地下水系统中的迁移转化机理。

综上所述，有机污染物在河流-地下水系统中的迁移转化机理包括对流、弥散、吸附动力学过程、等温平衡吸附过程和生物降解等多个方面[7]。这些过程之间是相互影响的，需要通过数学模型来进行描述和分析。

5.3 问题 3 的模型建立与求解

5.3.1 问题 3 模型建立

一、Monod 模型

Monod 模型是一个微生物降解动力学模型，它实际上是一个经验模型，可用于描述微生物在特定的环境下对有机物质的降解速率。我们令参与酶催化反应的反应物为底物S。并假设单一S作为微生物生长的唯一营养物质[8]。我们用反应速率常数k表示微生物利用底物S进行生长的速度。假设底物S的浓度为S，微生物的生长速率为μ，底物S的利用速率为μS。

同时可得底物S的消耗速率与底物 S 的浓度的正比关系为：$\mu S = kS$，微生物生长速率与底物浓度S的关系为：$\mu = kS$。微生物的生长速率μ是受到抑制的，当S值很大时，微生物生长速率会趋向于一个最大值μ_{max}。所以最终可导出模型:

$$\mu = \frac{\mu_{max} S}{K_s + S} \tag{16}$$

K_s是底物浓度为半饱和时的底物浓度。当它远大于S时$\mu \approx kS$，当它远小于S时

$$\mu \approx \mu_{max} \tag{17}$$

二、Andrews 模型

Andrews 是基于饱和性动力学模型的一个变种，着重研究了底物的浓度对微生物生长的影响[9]。模型可表示为:

$$\frac{dC}{dt} = -kCB \tag{18}$$

C表示有机污染物的浓度，t为时间，k是一个表示生物降解速率的常数。B是微生物浓度。

已知 9 天的有机污染物浓度变化、微生物浓度变化、有机物浓度比可以进一步导出模型。

其中B可以采用下式计算得出：

$$B = \frac{C}{0.483} \cdot 1.5 \times 10^7 \tag{19}$$

k以使用下式得出：

$$k = \frac{1}{C}\frac{dC}{dt} \cdot \frac{1}{B} \tag{20}$$

k和B代入 Andrews 模型可得:

$$\frac{dC}{dt} = -\frac{C}{\tau} \cdot \frac{C}{C+K} \cdot B \tag{21}$$

K是半饱和常数，可以使用下式计算：

$$\tau = \frac{1}{k} \cdot \frac{1}{B} \tag{22}$$

一个微生物生成的时间常数，计算方式如：

$$K = \frac{0.1 \cdot B}{\tau} \tag{23}$$

给定数据代入可得τ为 9.9375，K为1.2727×10^{-7}最终可得进一步推导出的 Andrews 模型为：

$$\frac{dC}{dt} = -\frac{C}{3.9375} \cdot \frac{C}{C + 1.2727 \times 10^{-7}} \cdot \frac{C}{0.483} \cdot 1.5 \times 10^7 \tag{24}$$

三、Haldane 模型

Haldane 模型也是基于饱和性动力学模型的一个变种，也是常用于描述微生物降解有机污染物的数学模型[10]。方便研究微生物对污染物的降解速率与污染物浓度之间的相关关系。模型表示为：

$$v = \frac{V\max \cdot S}{K_S + S + K_S \cdot \frac{S}{K_i}} \tag{25}$$

表示的是微生物降解污染物的速率，v_{max}为微生物降解有机污染物的最大速率，S表示有机污染物的浓度，K_s与K_i分别表示有机物浓度达到v_{max}的一半所需的浓度与达到 1/4 所需的浓度。

5.3.2 问题 3 模型的求解

为方便对模型求解效果进行对比，我们首先需要对数据进行可视化分析以及必要的数据预处理。

首先我们对于有机污染物浓度进行数据可视化，如图 5-9 所示，该图显示了有机污染物浓度变化趋势，观察可知，污染物浓度先快速下降后缓慢下降。下降速率放缓的临界点为第 7 天。

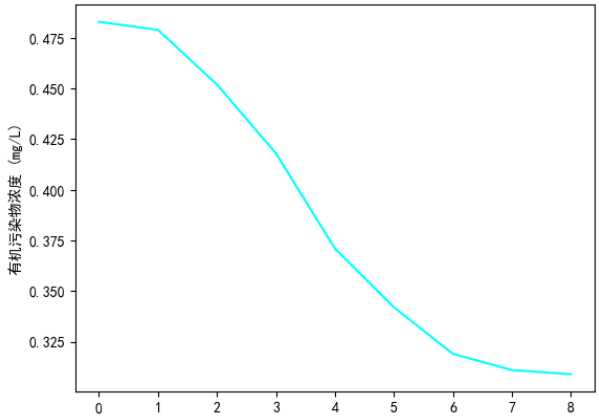

图 5-9 有机污染物浓度变化图

接着我们对于微生物浓度进行可视化，如图 5-10 所示，该图显示了微生物浓度变化趋势。观察可知，微生物浓度变化与有机污染物浓度变化趋势相反。微生物浓度上升趋势相对平缓，上升速率放缓的临界点为第 7 天。

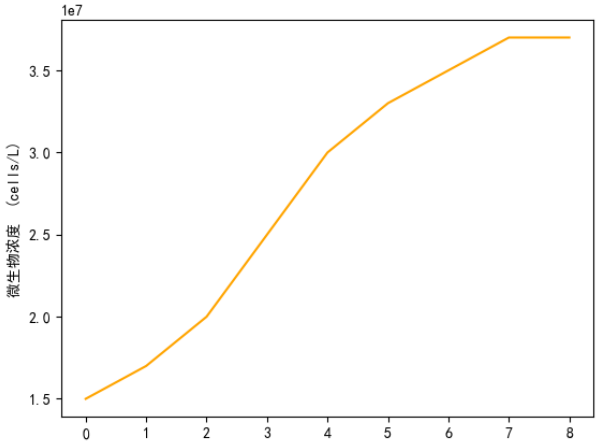

图 5-10 微生物浓度变化图

最后，有机物浓度比能较好的综合体现有机污染物浓度变化与微生物浓度变化，我们对其也进行可视化分析，结果如图 5-11 所示。观察可知其走势与有机物浓度变化吻合。同样表现为先快速下降后缓慢下降，不过放缓的临界点与微生物浓度变化同为第 7 天。

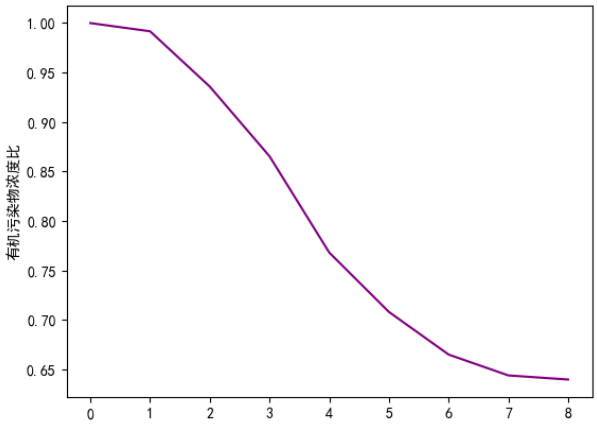

图 5-11 有机污染物浓度比变化趋势图

对于有机污染物浓度变化、微生物浓度变化、有机污染物浓度比三种变化统一观察有助于我们结合模型分析微生物对该有机污染物的降解特性。

由于明显的量纲差异，有机污染物浓度与有机污染物浓度比取值均小于 1，而微生物浓度的取值却远远大于二者，我们使用 min-max 对数据进行归一化，归一化方式为：

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{26}$$

据进行归一化后得到的结果如表 5-2 所示。

表 5-2 给定数据归一化结果表

有机污染物浓度	微生物浓度	有机污染物浓度比
1	0	1
0.977011	0.090909	0.977011

0.821839	0.227273	0.821839
0.626437	0.454545	0.626437
0.356322	0.681818	0.356322
0.189655	0.818182	0.189655
0.057471	0.909091	0.070024
0.011494	1	0.011494
0	1	0

使用归一化数据进行可视化结果如图 5-12 所示。可见，在第 7 天处有机污染物浓度与有机污染物浓度比出现一定偏差，如果模型能得出在第 8 天之后微生物降解出现饱和性，这一变化趋势也能更利于佐证模型的正确性。

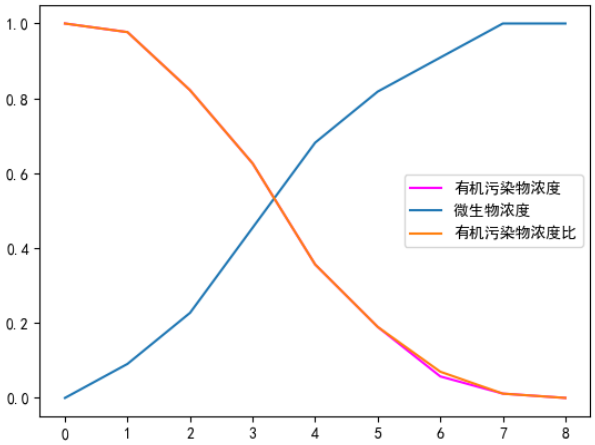

图 5-12 归一化数据变化图

对于已经给出理论上是一个周期的数据，我们对 Monod 模型使用差分进化算法求解其μ_{max}与K_s，最终得到μ_{max}取值为 64.886414，K_s取值为 0.482999。图 5-13 为模型拟合图，结合图与参数中最大微生物生长速率μ_{max}取值与底物浓度为半饱和时的底物浓度K_s取值，我们可知在第 8 天底物浓度与微生物浓度难以再有上升空间，表面微生物对该有机物降解达到了饱和状态。

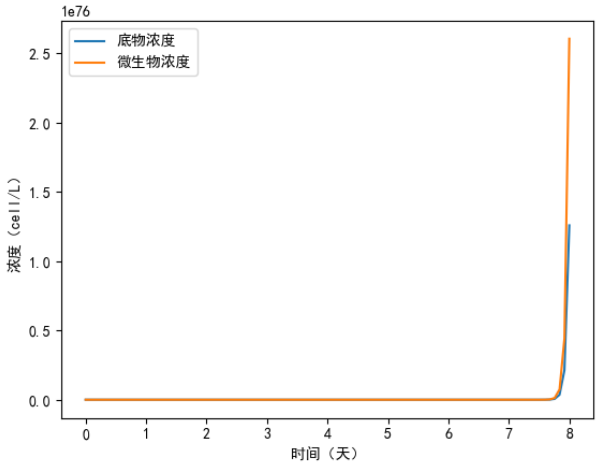

图 5-13 Monod 模型拟合图

使用给定的数据结合最小二乘法，也可得出 Andrews 模型的参数k与B，通过迭代求解可得k的取值为 0.067048，B的取值 2375085667.863919。拟合结果如图 5-14 所示。拟合曲线与生物降解速率常数k的取值与微生物浓度B的取值，推导表示，有机物浓度已经下降至极值附近。

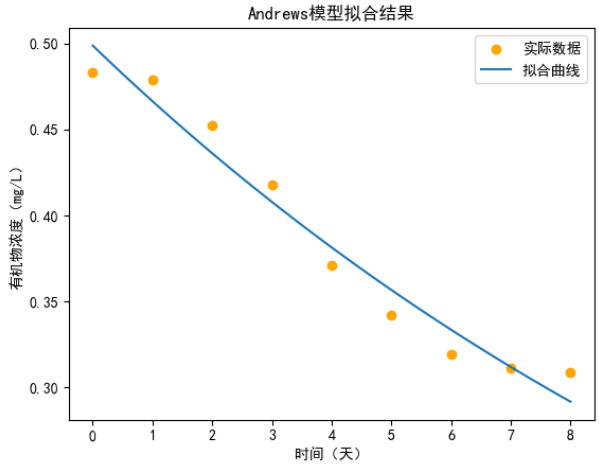

图 5-14 Andrews 模型拟合图

同理使用差分进化算法通过给定数据可以拟合出 Haldane 模型，从而得出参数。通过迭代求解生物降解有机污染物的最大速率v_{max}取值为 84222584.4051259，机物浓度达到v_{max}的一半所需的浓度K_s取值为 5.813941，机物浓度达到v_{max}的 1/4 所需的浓度K_i的取值为 891.095750。通过求解的三个参数与拟合图 5-15 可得，在第 7-8 天时微生物浓度变化放缓。

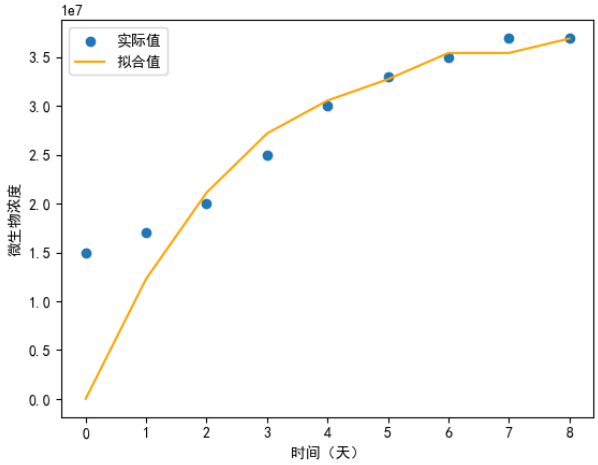

图 5-15 Haldane 模型拟合图

5.3.3 结果

在本节，使用了 Monod 模型、Andrews 模型、Haldane 模型输入给定数据并求解模型参数，具体而言我们求解了 Monod 模型的最大微生物生长速率μ_{max}取值为 64.886414 与底物浓度为半饱和时的底物浓度K_s取值为 0.482999。求解了 Andrews 模型的生物降解速率常数k的取值为 0.067048，微生物浓度B的取值 2375085667.863919。求解了 Haldane 模型的微生物降解有机污染物的最大速率v_{max}取值为 84222584.4051259，有机物浓度达到v_{max}的一半所需的浓度K_s取值为 5.813941，机物浓度达到v_{max}的 1/4 所需的浓度K_i的取值为 891.095750。

通过模型数值与模型拟合图，再结合对实验数据的可视化。我们可以发现从 0 到第 1 天，有机污染物浓度比下降 0.8%,微生物浓度上升 13.3%，拟合图与可视化图均可得出曲线增长较为平缓，说明 0-1 天是初始节点，微生物已经开始逐渐慢速发挥作用。从 1 天到第 3 天，有机污染物浓度比下降了 13.45%相比初始阶段的 0.8%是一个较大的提升。而微生物浓度上升了 66.7%同样是一个较大的提升，说明微生物对该有机物的降解能力快速增强，微生物数量也快速增加。从第 3 天到第 8 天，有机物污染物浓度比下降了 32.1%而微生物浓度增加了 146.7%。这时微生物增速逐渐放缓，在 7-8 已经无限趋近于降解能力饱和状态。

总之，微生物对该有机物在 0-1 天时较慢速度逐渐发挥降解作用，在 1-3 天时快速

发挥降解作用，从第 3 天齐降解作用逐渐增长放缓，在第 7-8 天时降解能力趋近于饱和

六、 模型的评价及优化

6.1 误差分析

6.1.1 针对于问题 1 的误差分析

1.假设地下水渗流场为各向同性均质的稳态流，忽略了地下水流动的复杂性和不稳定性，这可能会对模型的精度产生影响。

2.模型参数的确定也可能存在误差。例如，对于有机污染物的吸附等参数，可能由于实验条件的不同或者数据处理方法的不同而产生误差。

6.1.2 针对于问题 2 的误差分析

1 考虑河流、地下水的流动方向和速度，以满足边界条件。但是这样做方便更好地理解和预测有机污染物在地下水中的迁移规律的同时，又会因为流动方向和速度的不确定性产生一定的误差。

2 建立的模型是假设存在一维河流-地下水系统，其中 x 轴正方向表示河流方向，x 轴负方向表示地下水方向，但在实际生活中河流-地下水系统情况并不稳定且多变。

6.1.3 针对于问题 3 的误差分析

1.使用最小二乘法进行求解，最小二乘法假设数据噪声服从高斯分布，实际上数据噪声并不能确保服从高斯分布。

2.使用差分进化算法，参数空间的大小不能确保是足够的，参数变异方式不能确保是最适合的，不确定性因素导致模型误差。

3.Monod 模型、Andrews 模型、Haldane 模型君假设微生物生长速率与底物浓度之间是单峰函数关系，并假设微生物在不同营养条件与环境条件下是稳定的，这与实际实验情况通常是有一定差异的，这导致模型误差。

6.2 模型的优点

1.采用改进欧拉算法进行求解，该算法具有计算精度高、计算速度快、稳定性好等优点，能够有效地解决数值计算问题，提高模型的可靠性和精度。

2.假设地下水渗流场为各向同性均质的稳态流，这种假设能够简化模型，使其易于建立和求解，同时也能够适用于许多实际情况，具有一定的普适性。

3.考虑了河流地下水系统中有机污染物的物理过程、化学反应过程以及生物转化过程等多种影响因素，能够全面反映有机污染物的迁移和转化规律，对于制定生物修复方案等具有重要的指导意义。

4.利用实验数据对模型进行了验证和修正，提高了模型的可靠性和精度。

能够深入研究某有机污染物在河流-地下水系统中的迁移转化机理，为制定污染治理方案提供了理论支持。

5.通过对模型的求解，能够得到有机污染物在河流-地下水系统中的浓度分布、迁移速度等重要参数，为进一步研究有机污染物的迁移和转化规律提供了重要的数据支持。

6.能够全面考虑生物降解对有机污染物转化的影响，包括微生物数量、生长速率、降解速率等多个因素，具有较高的可靠性和适用性。

7.通过对模型的求解，能够得到有机污染物在河流-地下水系统中的降解速率、降解程度等重要参数，为进一步研究有机污染物的迁移和转化规律提供了重要的数据支持。

6.3 模型的缺点

1.该模型假设地下水渗流场为各向同性均质的稳态流，实际情况可能会存在非均质性和非稳态性，因此模型的适用范围可能受到一定限制。

2.模型中对试验参数的选择可能存在一定的主观性和局限性，对结果的影响需要进一步探讨。

3.模型中对微生物降解作用的描述可能过于简化，没有考虑微生物种类、数量、生长环境等因素的影响，需要进一步验证和修正。

6.4 模型的推广

该模型可以推广到其他地区和不同的有机污染物的研究中，以更全面地理解有机污染物在河流-地下水系统中的行为特征，为预测和控制地下水污染提供更为准确的科学依据。此外，该模型还可与其他模型相互融合，以提高对地下水污染的预测和控制能力。同时，该模型还可应用于地下水污染防治和治理的研究中，为控制地下水污染提供更为科学的依据，为保障人类健康和生态环境的安全贡献更多的力量。

参考文献

[1]钱天伟, 陈繁荣, 武贵斌, 等. 一种耦合表面络合吸附作用的地下水溶质迁移模型初探[J]. 辐射防护通讯, 2003, 23(3): 14-18.

[2]Huggenberger P, Epting J, Scheidler S. Concepts for the sustainable management of multi-scale flow systems: the groundwater system within the Laufen Basin, Switzerland[J]. Environmental earth sciences, 2013, 69: 645-661.

[3]江文豪, 李江山, 黄啸, 等. 考虑对流-扩散-吸附-降解时成层介质中有机污染物一维运移的解析模型[J]. 岩土工程学报, 2023, 45(2): 262-272.

[4]Díaz-Blancas V, Aguilar-Madera C G, Flores-Cano J V, et al. Evaluation of mass transfer mechanisms involved during the adsorption of metronidazole on granular activated carbon in fixed bed column[J]. Journal of Water Process Engineering, 2020, 36: 101303.

[5]Bérard A, Blais B, Patience G S. Residence time distribution in fluidized beds: diffusion, dispersion, and adsorption[J]. Advanced Powder Technology, 2021, 32(5): 1677-1687.

[6]Tan K L, Hameed B H. Insight into the adsorption kinetics models for the removal of contaminants from aqueous solutions[J]. Journal of the Taiwan Institute of Chemical Engineers, 2017, 74: 25-48.

[7]Simonin J P, Bouté J. Intraparticle diffusion-adsorption model to describe liquid/solid adsorption kinetics[J]. Revista mexicana de ingeniería química, 2016, 15(1): 161-173.

[8]杨扬,李雅洁,崔益斌.3 种典型有机污染物对 2 种水生生物的急性毒性及安全评价[J].环境科学,2015,36(8):3074-3079

[9]Karlsson C M G, Cerro‐Gálvez E, Lundin D, et al. Direct effects of organic pollutants on the growth and gene expression of the Baltic Sea model bacterium Rheinheimera sp. BAL 341[J]. Microbial Biotechnology, 2019, 12(5): 892-906.

[10]Cerro-Gálvez E, Sala M M, Marrasé C, et al. Modulation of microbial growth and enzymatic activities in the marine environment due to exposure to organic contaminants of emerging concern and hydrocarbons[J]. Science of the Total Environment, 2019, 678: 486-498.

附 录

附录 A 文件列表

文件名	文件描述
Problem1.py	问题 1 求解代码
Problem2.py	问题 2 求解代码
Problem3.py	问题 3 求解代码

附录 B 代码

Problem1.py

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# 模型参数设置
L = 1 # 空间区间长度
T = 10 # 时间区间长度
m = 10 # 空间区间划分数
n = 100 # 时间区间划分数
dx = L / m # 空间步长
dt = T / n # 时间步长
u = 0.1 # 地下水流速
D = 0.01 # 有机污染物的弥散系数
k = 0.1 # 有机污染物的吸附速率

# 初始条件设置
C0 = np.zeros(m+1) # 初始时刻有机污染物在不同空间位置的浓度分布
C0[0] = 1 # 在 x=0 处设置初始浓度为 1

# 迭代求解差分方程
C = np.zeros((m+1, n+1)) # 初始化有机污染物在不同时刻和空间位置的浓度分布
C[:, 0] = C0 # 设置初始条件
for j in range(n):
    for i in range(1, m):
        C[i, j+1] = C[i, j] + dt * (u * (C[i, j] - C[i-1, j]) / dx - D * (C[i+1, j] - 2*C[i, j] + C[i-1, j]) / dx**2 - k * C[i, j])

# 可视化结果
```

```
# 1. 时刻为 0 时的浓度分布
```

```
plt.plot(np.linspace(0, L, m+1), C[:, 0], label='t=0')
plt.xlabel('x')
plt.ylabel('C')
plt.legend()
plt.show()

# 2. 不同时刻的浓度分布
t_list = [5,8, 10] # 设置需要可视化的时刻
for t in t_list:
    plt.plot(np.linspace(0, L, m+1), C[:, int(t/dt)], label=f't={t}')
plt.xlabel('x')
plt.ylabel('C')
plt.legend()
plt.show()

# 3. 不同弥散系数 D 下的浓度分布
D_list = [0.001, 0.01, 0.1] # 设置需要可视化的弥散系数
for D in D_list:
    # 迭代求解差分方程
    C = np.zeros((m+1, n+1))
    C[:, 0] = C0
    for j in range(n):
        for i in range(1, m):
            C[i, j+1] = C[i, j] + dt * (u * (C[i, j] - C[i-1, j]) / dx - D * (C[i+1, j] - 2*C[i, j] + C[i-1, j]) / dx**2 - k * C[i, j])
    # 可视化结果
    plt.plot(np.linspace(0, L, m+1), C[:, -1], label=f'D={D}')
plt.xlabel('x')
plt.ylabel('C')
plt.legend()
plt.show()

# 4. 不同吸附速率 k 下的浓度分布
k_list = [0.001, 0.1, 1] # 设置需要可视化的吸附速率
for k in k_list:
    # 迭代求解差分方程
    C = np.zeros((m+1, n+1))
    C[:, 0] = C0
    for j in range(n):
        for i in range(1, m):
            C[i, j+1] = C[i, j] + dt * (u * (C[i, j] - C[i-1, j]) / dx - D * (C[i+1, j] - 2*C[i, j] + C[i-1, j]) / dx**2 - k * C[i, j])
    # 可视化结果
```

```
    plt.plot(np.linspace(0, L, m+1), C[:, -1], label=f'k={k}')
plt.xlabel('x')
plt.ylabel('C')
plt.legend()
plt.show()

# 5. 三维图像可视化
x = np.linspace(0, L, m+1)
t = np.linspace(0, T, n+1)
X, T = np.meshgrid(x, t)
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot_surface(X, T, C, cmap='rainbow')
ax.set_xlabel('x')
ax.set_ylabel('t')
ax.set_zlabel('C')
plt.show()
```

Problem2.py

```
import numpy as np
import matplotlib.pyplot as plt

plt.rcParams['font.sans-serif'] = ['SimHei']  # 用来正常显示中文标签
plt.rcParams['axes.unicode_minus'] = False  # 用来正常显示负号
yeS1 = [
    0.495, 0.355, 0.312, 0.305, 0.310, 0.314, 0.325, 0.353, 0.360, 0.359,
    0.362, 0.371, 0.365, 0.364
]
guS1 = [
    0,
    2.30,
    2.73,
    2.80,
    2.75,
    2.71,
    2.60,
    2.32,
    2.25,
    2.26,
    2.23,
    2.14,
    2.20,
    2.21,
]
plt.plot(yeS1, label='S1 液相')
```

```
plt.plot(guS1,  label='S1 固相')

yeS2  =  [
        0.495,
        0.401,
        0.327,
        0.280,
        0.246,
        0.255,
        0.269,
        0.186,
        0.324,
        0.354,
        0.349,
        0.352,
        0.344,
        0.347,
]

guS2  =  [
        0,
        1.84,
        2.58,
        3.05,
        3.39,
        3.30,
        3.16,
        2.99,
        2.61,
        2.31,
        2.36,
        2.33,
        2.41,
        2.33,
]

plt.plot(yeS2,  label='S2 液相')
plt.plot(guS2,  label='S2 固相')

yeS3  =  [
        0.495,
        0.225,
        0.086,
        0.080,
        0.078,
        0.068,
        0.085,
        0.093,
        0.138,
        0.149,
```

```
        0.171,
        0.169,
        0.172,
        0.173,
    ]

guS3 = [
    0, 3.60, 5.09, 5.05, 5.07, 5.17, 4.80, 4.30, 4.41, 4.36, 4.14, 4.16, 4.13,
        4.12
    ]

plt.plot(yeS3, label='S3 液相')
plt.plot(guS3, label='S3 固相')

yeS4 = [
        0.495,
        0.367,
        0.284,
        0.224,
        0.195,
        0.174,
        0.185,
        0.202,
        0.199,
        0.197,
        0.196,
        0.199,
        0.198,
        0.199,
    ]

guS4 = [
        0,
        2.18,
        3.01,
        3.61,
        3.90,
        4.11,
        4.00,
        3.83,
        3.86,
        3.88,
        3.89,
        3.86,
        3.87,
        3.86,
    ]

plt.plot(yeS4, label='S4 液相')
```

```
plt.plot(guS4,  label='S4 固相')

plt.legend()
plt.show()
#  输入已知数据
u  =  38.67    #  平均孔隙流速
v  =  5.01    #  地下水渗流流速
k  =  6.32    #  渗透系数
D  =  0.38    #  弥散系数
rho  =  1.67    #  干密度
n  =  0.375    #  孔隙度

#  假设有机污染物名称为 S1
#  输入 S1 在固相和液相中的浓度数据，以及等温平衡吸附 24 小时后液、固相中 S1 浓度变化
s1_solid  =  np.array([2.3,  2.73,  2.8,  2.75,  2.71,  2.6,  2.32,  2.25,  2.26,  2.23,  2.14,  2.2,  2.21])
s1_liquid  =  np.array([0.355,  0.312,  0.305,  0.31,  0.314,  0.325,  0.353,  0.36,  0.359,  0.362,  0.371,  0.365,  0.364])
s1_eq_solid  =  1.39348    #  某有机物在固相和液相之间的平衡浓度（24h）
s1_eq_liquid  =  0.30317    #  某有机物在固相和液相之间的平衡浓度（24h）

#  定义模拟参数
dx  =  0.5    #  空间步长
dt  =  0.1    #  时间步长
L  =  100    #  空间长度
T  =  100    #  模拟时间
nx  =  int(L  /  dx)  +  1    #  网格数
nt  =  int(T  /  dt)  +  1    #  时间步数

#  初始化模拟数据
c_solid  =  np.zeros(nx)    #  固相浓度
c_liquid  =  np.zeros(nx)    #  液相浓度
c_liquid[0]  =  s1_liquid[0]    #  初始液相浓度
c_solid[0]  =  s1_solid[0]    #  初始固相浓度

#  定义模拟方程
def  simulate(c_solid,  c_liquid,  c_eq_solid,  c_eq_liquid,  u,  v,  k,  D,  rho,  n,  dx,  dt,  nx):
        for  i  in  range(1,  nx):
                #  计算对流项
                convective  =  -u  *  (c_liquid[i]  -  c_liquid[i-1])  /  dx
                #  计算弥散项
                if  i  ==  1:
                        diffusive  =  D  *  (c_liquid[i+1]  -  2*c_liquid[i]  +  c_liquid[i-1])  /  dx**2
                elif  i  ==  nx-1:
```

```
                diffusive = D * (c_liquid[i] - 2*c_liquid[i-1] + c_liquid[i-2]) / dx**2
            else:
                diffusive = D * (c_liquid[i+1] - 2*c_liquid[i] + c_liquid[i-1]) / dx**2
            # 计算吸附项
            adsorption = k * rho * n * (c_eq_liquid - c_liquid[i])
            # 计算液相浓度
            c_liquid[i] += dt * (convective + diffusive + adsorption)
            # 计算固相浓度
            desorption = k * rho * n * (c_solid[i-1] - c_eq_solid)
            c_solid[i] += dt * desorption
        return c_solid, c_liquid

# 进行模拟
for i in range(1, nt):
    c_solid, c_liquid = simulate(c_solid, c_liquid, s1_eq_solid, s1_eq_liquid, u, v, k, D, rho, n, dx, dt, nx)

# 绘制结果
x = np.linspace(0, L, nx)
plt.plot(x, c_solid, label='Solid phase concentration')
plt.plot(x, c_liquid, label='Liquid phase concentration')
plt.xlabel('Distance (m)')
plt.ylabel('Concentration (mg/L)')
plt.legend()
plt.show()
```

Problem3.py

```
import numpy as np
from scipy.optimize import differential_evolution
import matplotlib.pyplot as plt
from scipy.integrate import odeint

import numpy as np
from scipy.optimize import curve_fit
import matplotlib.pyplot as plt

# 定义 Monod 模型拟合函数
def monod_func(x, *params):
    c0 = params[0]
    t = np.arange(0, len(c0))
    mmax = x[0]
    ks = x[1]
    c = c0[0] / (1 + (c0[0]/ks - 1) * np.exp(-mmax * t))
```

```
    return np.sum((c - c0)**2)

# 输入数据
c0 = np.array([0.483, 0.479, 0.452, 0.418, 0.371, 0.342, 0.319, 0.311, 0.309])
params = (c0,)

# 定义变量上下限
bounds = [(0, 100), (0, 1)]

# 使用差分进化算法求解最优解
result = differential_evolution(monod_func, bounds, args=params)

# 输出最优解
print('mmax:', result.x[0])
print('ks:', result.x[1])


# 支持中文
plt.rcParams['font.sans-serif'] = ['SimHei']  # 用来正常显示中文标签
plt.rcParams['axes.unicode_minus'] = False  # 用来正常显示负号

# 定义 Monod 模型
def monod_model(X, t, S, mu_max, K_s):
    dXdt = mu_max*S/(K_s+S)*X
    return dXdt

# 初始值
X0 = np.array([1.50E+07]*100)

# 初始条件，设置为长度为 100 的数组，初始值为 0.483
y0 = np.array([0.483]*100)

# 时间点
t = np.linspace(0, 9, 100)

# 参数
mu_max = 64.88641409578585  # 最大生长速率
K_s = 0.4829999937932565  # Ks 值


# 底物浓度
S = odeint(monod_model, y0, t, args=(y0, mu_max, K_s))[:,0]

# 微生物浓度
```

```
X = odeint(monod_model, X0, t, args=(S, mu_max, K_s))[:,0]


# 绘图
plt.plot(t, S, label='底物浓度')
plt.plot(t, X/X0, label='微生物浓度')
plt.legend()
plt.xlabel('时间（天）')
plt.ylabel('浓度（cell/L）')
plt.show()



# 数据
days = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8])
org_concs = np.array([0.483, 0.479, 0.452, 0.418, 0.371, 0.342, 0.319, 0.311, 0.309])
microbe_concs = np.array([1.50E+07, 1.70E+07, 2.00E+07, 2.50E+07, 3.00E+07, 3.30E+07, 3.50E+07, 3.70E+07, 3.70E+07])
org_conc_ratio = np.array([1.0, 0.991718427, 0.935817805, 0.865424431, 0.768115942, 0.708074534, 0.708074534, 0.64389234, 0.639751553])


# Haldane 模型函数
def haldane_model(t, Vm, Ks, KI):
    return Vm * org_conc_ratio / (1 + org_conc_ratio * (Ks / t) - (t / KI))


# 差分进化算法进行拟合
def differential_evolution_fit(func, xdata, ydata, bounds, **kwargs):
    from scipy.optimize import differential_evolution
    result = differential_evolution(lambda p: np.sum((ydata - func(xdata, *p))**2), bounds=bounds, **kwargs)
    return result.x


# 参数边界
bounds = [(0, 1E+8), (0, 1E+4), (0, 1E+4)]


# 拟合
params = differential_evolution_fit(haldane_model, days, microbe_concs, bounds)


# 输出结果
print('Haldane 模型拟合结果：')
print('Vm =', params[0])
print('Ks =', params[1])
print('KI =', params[2])


# 绘制拟合曲线
plt.scatter(days, microbe_concs, label='实际值')
```

```
    104.plt.plot(days, haldane_model(days, *params), label='拟合值',color='orange')
    105.plt.xlabel('时间（天）')
    106.plt.ylabel('微生物浓度')
    107.plt.legend()
    108.plt.show()
```