

河流-地下水系统水体有机污染的分析与模型开发

摘要

本题主要是研究开发一个河流-地下水系统有机污染模型。首先分析河流、地下水之间的相互作用和污染物的迁移过程并建立对流-弥散-吸附数学模型；接着利用题目所给试验参数和结果并结合理论模型研究有机污染物在河流-地下水系统中的迁移转化机理；最后结合表 4 中的试验数据分析微生物对该有机污染物的降解特性并考虑生物降解作用对有机污染物转化的影响，建立相关的数学模型。

对于问题一，本文基于**质量守恒方程**首先结合 **Darcy 定律**推导得出对流模型，该模型描述了有机污染物的浓度随时间的变化率等于有机污染物在流动地下水中的输运通量减少的速率；接着结合 **Fick 第一定律**推导得出弥散模型，该模型描述了有机污染物在地下水中的运动轨迹和浓度分布；然后结合 **Langmuir 等温线理论**推导得出吸附模型，用以描述有机污染物的总量变化。最后综合考虑三种影响作用并联立上述三种分模型建立一个完整的有机污染物在对流-弥散-吸附作用下的数学模型再使用**有限差分法**将模型转化为一个离散的方程组，然后应用数值求解算法进行求解，并得到污染物浓度分布。

对于问题二，考虑到对流-弥散-吸附等作用对有机污染物在河流-地下水系统中的迁移转化机理的影响，结合试验数据分析这些作用对有机污染物的传输的影响特性。本文使用随机性模型描述有机污染物在河流-地下水系统中的迁移转化机理。根据提供的试验数据和使用**有限差分法**的数值方式求解，得到**有机污染物浓度迭代公式**： $C^{n+1} = A^{-1}(B - C^n + F)$ 。从实验拟合的结果来说该模型在数值求解时需要隐式差分格式，选择 Crank-Nicolson 格式，让因为模型的性能提升，具有更好的数值稳定性和精度。符合河流-地下水系统实际情况中复杂影响因素，鲁棒性强。

对于问题三，考虑到生物降解作用对有机污染物转化的影响，结合试验数据分析微生物对该有机污染物的降解特性。本文使用 **Monod 模型**描述微生物对底物的降解过程。根据提供的试验数据和使用 Monod 模型拟合的结果，得到 μ_{max} 和 K_S 。从试验数据中观察，有机物浓度随着时间的增加而逐渐降低，且降低的速率逐渐减缓。这与微生物的降解特性相符，微生物对有机物的降解过程是一个逐渐减速的过程。较大的 μ_{max} 值说明微生物具有较高的降解能力，而较小的 K_S 值说明微生物对低浓度的有机物也能进行有效降解。结果得出微生物对该有机污染物具有较高的降解能力，能够以较快的速度对有机物进行降解。此外，微生物对低浓度的有机物也具有一定的降解能力，显示出对低浓度有机污染物的敏感性。

关键词 Darcy 定律；Fick 第一定律；Langmuir 等温线；有限差分法；Monod 模型

目 录

一、 问题重述	3
1.1 问题背景	3
1.2 数据分析	3
1.3 问题提出	3
二、 问题分析	4
2.1 问题 1 的分析	4
2.2 问题 2 的分析	4
2.3 问题 3 的分析	5
三、 模型假设	5
四、 定义与符号说明	5
五、 模型的建立与求解	6
5.1 问题 1 的模型建立与求解	6
5.2 问题 2 的模型建立与求解	11
5.3 问题 3 的模型建立与求解	20
六、 模型的评价及优化	23
6.1 误差分析	23
6.2 模型的优点	24
6.3 模型的缺点	24
6.4 模型的推广	25
参考文献	25
附录	26

一、问题重述

1.1 问题背景

河流-地下水系统的水体污染研究主要关注在河流和地下水之间的相互作用和污染物的迁移过程。这个系统对于水资源的管理和保护至关重要，因为许多地区的饮用水和农业用水都来自于该系统。通过这些研究，我们可以更好地理解河流-地下水系统中的水体污染问题，为保护水资源、改善水质提供科学依据，制定有效的管理和治理措施。

河流直接影响地下水，河流污染可能导致地下水和附近水源的污染。这对工农业运作、社会经济发展和饮水安全产生严重影响。有机污染是地下水污染中最难治理且最具危害性的类型。因此，研究有机污染物在河流-地下水系统中的行为特征具有重要的理论意义和实际价值。现有研究表明，在河流地下水系统中，有机污染物的行为特征涉及物理过程（对流迁移、水动力弥散、吸附和阻滞）、化学反应过程和生物转化过程等。本研究设定地下水渗流场为各向同性均质的稳态流，旨在研究和探索有机污染物的迁移和转化规律，并回答相关问题。

1.2 数据分析

本题总共给出四个附件表格数据，在此做一个简要分析。

表 1 显示了通过试验测得河流-地下水系统中某有机污染物的对流、弥散有关参数。

表 2 显示了四种不同河流沉积物对初始浓度为 0.5mg/L 左右的某有机污染物吸附体系的吸附动力学过程及不同吸附时间测得固、液相某有机物的浓度。

表 3 显示了地下水中有有机污染物的吸附行为采用等温平衡吸附的数学模型描述，四种不同沉积物对 10 种不同初始浓度的某有机污染物 24 小时的等温平衡吸附试验结果。

表 4 显示了 0.483mg/L 有机污染物对应的微生物降解过程。

1.3 问题提出

问题 1：请通过研究相关文献和资料，建立一个数学模型，分析河流-地下水系统中有机污染物的对流、弥散和吸附过程。

问题 2：请运用以下说明的内容、实验参数和数据，使用数学模型研究某有机污染物在河流-地下水系统中的迁移和转化机制。

问题 3：生物降解是有机污染物转化的一个关键过程。请考虑生物降解对有机污染物转化的影响，并建立一个适当的数学模型，结合表 4 中的实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

问题一属于解决地下水污染迁移模型建立和求解的数学问题，解决此类问题，一般的数学方法包括质量守恒方程和反应动力学方程的建立，质量守恒方程描述了污染物的质量守恒规律，反应动力学方程则描述了污染物在地下水中的反应过程。这两个方程的联立为基础可以建立一个地下水污染物迁移模型基础。

扩散是物质在物理化学作用下，由浓度不一引起的物质运动现象，它是由不均匀向均匀发展的过程既有沿着运动方向的横向扩散，也有向垂直方向低浓度区域的纵向扩散，Darcy 定律可以用来描述地下水流动。而地下水中对有机污染物的弥散作用则分为机械弥散和水动力弥散，由于在多孔介质中水动力弥散作用进行的很慢，对于有机污染物浓度梯度不是很大的情况下影响很小，所以本文主要考虑机械弥散作用，使用 Fick 定律可以用来描述有机污染物在地下水中的弥散，对于有机污染物在地下水中的吸附作用主要为化学吸附、物理吸附和生物吸附，本文主要考虑化学吸附和物理吸附，所以本文采用 Langmuir 等温线来描述有机污染物在地下水中的吸附作用。将上述三者纳入该模型中，可以得出完整的有机污染物在地下水中的对流-弥散-吸附作用下的数学模型。

2.2 问题 2 的分析

问题二属于研究地下水-河流中有机污染物的迁移转化规律问题。为解决此类问题，一般可以通过建立含有相关参数的偏微分方程。

表 1 为对流-弥散试验的各项参数，结合表中参数联立 Darcy 定律和 Fick 第一定律建立理论基础模型。

由表 2 可知，S1、S2、S3、S4 四种不同河流沉积物对初始浓度为 0.5mg/L 左右的某有机污染物吸附体系的吸附动力学过程及不同吸附时间测得固、液相某有机物的浓度。又因该地下水-河流中有机污染物的吸附作用遵循准二级动力学，其速率仅与吸附剩余容量和液相浓度相关，所以本文忽略液相浓度和温度对吸附平衡常数的影响以及流速和颗粒效应，仅考虑时间影响。由此可以建立出一个相对完善的理论基础模型，再基于 python 使用迭代法将表 2 各项数据进行线性拟合，可以得出吸附速率常数等模型参数。

表 3 体现的是四种不同沉积物对 10 种不同初始浓度的某有机污染物 24 小时的等温平衡吸附试验结果，Langmuir 方程是在吸附介质表面均匀、被吸附物质为单分子层以及吸附介质之间无相互作用的假设条件下构建出来的，浓度较高时，Langmuir 方程与实际的误差非常大，不宜使用，但本题给出的等温平衡吸附试验污染物浓度较低，所以该方程与实际吸附过程可以吻合的较好。基于 python 将表 3 等温平衡吸附试验的各项结果数据对 Langmuir 等温线理论模型进行拟合吸附等温线，再建立 Langmuir-Arrhenius 模型来探究分析温度对吸附常数 K 的影响得出模型参数。

最后结合所求模型参数整合出一个较为完整的有机污染物在河流-地下水系统中的迁移转化机理模型，由于 Crank-Nicolson 格式具有更好的数值稳定性和精度，所以本文将基于 python 在数值求解的过程中使用 Crank-Nicolson 格式得出数值解。

2.3 问题 3 的分析

问题 3 关注到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，结合试验数据分析微生物对该有机污染物的降解特性。需要建立一个数学模型来描述生物降解作用对河流地下水系统中有机污染物转化的影响。

Monod 模型是一种常用的微生物生长动力学模型，用于描述微生物对底物的降解过程。在底物浓度较低时，微生物的生长速率与底物浓度成正比增加；而在底物浓度较高时，微生物的生长速率趋于饱和，不再增加。Monod 模型符合本题要求，故本文选择使用 Monod 模型对微生物变化进行描述。

通过拟合 Monod 模型到实验数据，可以估计出微生物的最大生长速率（ μ_{\max} ）和底物浓度对生长速率的半饱和常数（ K_s ），从而了解微生物对有机污染物的降解特性。

三、模型假设

1. 假设题目所给的数据真实可靠；
2. 假设地下水渗流场为各向同性均质的稳态流；
3. 假设吸附的物质与本体处于热力学平衡；
4. 吸附介质表面均匀、被吸附物质为单分子层以及吸附介质之间无相互作用；
5. 有机污染物的降解主要由微生物引起；
6. 微生物浓度遵循 Monod 模型。

四、定义与符号说明

符号定义	符号说明
C	有机污染物的浓度
I_c	有机污染物的质量通量
K_f	吸附速率常数
$\frac{dx}{dt}$	微生物生产速度
μ_{\max}	微生物的最大生长速度
$A、B$	系数矩阵
K_s	Monod 饱和常数
D	表示分子扩散系数
dM	有机污染物的总量变化为 dM
R	阻滞因子
F	源项

Table 1 符号说明

五、模型的建立与求解

数据的预处理：

1. 对表二空白处补零处理。
2. 将表 2、表 3 数据根据不同污染物和液固相来进行再划分。

5.1 问题 1 的模型建立与求解

5.1.1 溶质“运移合散”模型的建立

针对问题一，本文通过质量守恒方程和反应动力学方程的联立为基础可以建立一个地下水污染物迁移的基础模型。由于有机污染物在地下水中的移动受到地下水流动的影响，所以本文使用 Darcy 定律描述地下水流动，再使用 Fick 定律描述有机污染物在地下水中的弥散，最后使用 Langmuir 吸附等温线来描述有机污染物在地下水中的吸附作用，并将上述三者纳入该基础模型中，由此可以得出完整的有机污染物在地下水中的对流-弥散-吸附作用下溶质“运移合散”数学模型。

(1) 对流模型的建立和推导：

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

对流模型是描述有机污染物在地下水中随水流运动的模型。这个模型基于质量守恒定律和 Darcy 定律。其中， C 表示有机污染物的浓度， t 表示时间， v 表示地下水的流速。这个方程描述了有机污染物在地下水中的运动轨迹和浓度分布。具体地，这个方程表达了有机污染物的浓度随时间的变化率等于有机污染物在流动地下水中的输运通量减少的速率。这个输运通量由流速 v 和有机污染物的浓度 C 共同决定，因此方程中使用了守恒量 $\nabla \cdot (Cv)$ 来描述该通量的变化率。

首先，我们基于质量守恒定律得出有机污染物的质量守恒方程式：

$$\frac{\partial}{\partial t} \int_V C(x, t) dv + \oint_S J_c n ds = 0$$

其中， $C(x, t)$ 表示有机污染物的浓度， V 表示体积， S 表示表面， J_c 表示有机污染物的质量通量， n 表示表面法向量， ds 表示表面元素的面积。

接着，我们使用 Darcy 定律来描述流速和压力的关系。该定律表明，地下水的流速与压力梯度成正比：

$$v = -K \nabla p$$

其中， K 表示渗透率， p 表示压力。将 Darcy 定律代入质量守恒方程式中并用散度定理将表面积分转化为体积积分，得到：

$$\frac{\partial}{\partial t} \int_V C(x, t) dv + \int_V \nabla \cdot (-K \nabla p C(x, t)) \cdot dv = 0$$

我们可以将 $\nabla \cdot (-K\nabla p C(x, t))$ 进一步展开代入并将第二项使用散度定理将表面积分转化为体积积分，得到：

$$\frac{\partial}{\partial t} \int_V C(x, t) dv - \oint_S K \frac{\partial C(x, t)}{\partial n} ds - \int_V K \nabla p \cdot \nabla C(x, t) dv = 0$$

因为我们考虑到是对流模型，所以我们假设地下水的压力分布是恒定的，即 $\nabla p = 0$ 。这样，我们就可以将第三项去掉，并使用高斯定理将表面积分转化为体积积分，得到：

$$\int_V \frac{\partial C(x, t)}{\partial t} dv + \int_V (K \nabla C(x, t)) dv = 0$$

将方程进一步简化为：

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

(2) 弥散模型的建立和推导：

弥散现象的起源：地下水中有有机污染物的扩散是由于分子之间扩散而引起的。分子运动的随机性使得它们会向各个方向扩散，从而使有机污染物在地下水中发生弥散现象。

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

该偏微分方程描述了有机污染物在地下水中分子扩散而导致的分散现象。这个模型基于质量守恒定律和 Darcy 定律。其中， C 表示有机污染物的浓度， t 表示时间， D 表示分子扩散系数， ∇^2 表示拉普拉斯算子，表示浓度的二阶空间导数。这个方程描述了有机污染物在地下水中的运动轨迹和浓度分布。

根据质量守恒定律，我们可以得到以下方程：

$$\frac{\partial}{\partial t} (\rho C) + \nabla(\rho C u) = 0$$

其中， ρ 表示地下水的密度， C 表示有机污染物的浓度， u 表示地下水的流速。而地下水对有机污染物的弥散作用则分为机械弥散和水动力弥散^[1]，由于在多孔介质中水动力弥散作用进行的很慢，对于有机污染物浓度梯度不是很大的情况下影响很小，所以本文主要考虑机械弥散作用，所以我们可以使用 Fick 定律描述弥散现象：

$$J = -D \nabla C$$

其中， J 表示物质的弥散通量， D 表示分子的弥散系数，将 Fick 定律代入质量守恒定律中并将所得结果通过应用散度定理，得到：

$$\rho \frac{\partial C}{\partial t} = \nabla(D \rho \nabla C)$$

将上式中的 ρ 消去后即可得到弥散方程：

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

(3) 吸附模型的建立和推导：

吸附模型是描述有机污染物在地下水中因为与地下水中的固体颗粒表面发生相互作用而被吸附的模型。

$$\frac{\partial C_s}{\partial t} = K(C_0 - C)$$

该模型描述了有机污染物在地下水被吸附的过程，基于质量守恒定律和吸附定律。吸附定律是指，当有机污染物与地下水中的固体颗粒表面发生相互作用时，有机污染物会被固体颗粒吸附， K_f 吸附速率常数，表示单位内有机污染物被固体颗粒吸附的速率。

在该模型中我们假设有机污染物的总量为 M ，地下水的总量为 V 。接下来我们考虑一个微小的时间 dt 内，有机污染物被吸附的量。假设此时地下水的有机污染物浓度为 C_0 ，固体颗粒表面上的有机污染物浓度为 C_s ，那么，此时有机污染物被吸附量可以表示为：

$$dM_{ads} = K_f \cdot C_s \cdot A \cdot dt$$

其中， A 表示固体颗粒的表面积。根据质量守恒定律，这个微小的时间段内，有机污染物的总量变化为 dM_{ads} ，因为有机污染物在这个时间段内只可能被吸附，而不会增加。所以，有机污染物的总量变化量为负数，将 dM_{ads} 代入上式，可得：

$$dM_{ads} = -K_f \cdot C_s \cdot A \cdot dt$$

考虑到这个时间段内地下水中有有机污染物总量为 $V \cdot C$ ，根据质量守恒定律，这段时间内有机污染物总量变化为 $dM = V(C - C')$ ，其中， C' 表示时间段结束时地下水中有有机污染物的浓度。将上式的 dM 代入可得：

$$V(C - C') = -K_f \cdot C_s \cdot A \cdot dt$$

整理可得：

$$\frac{\partial C_s}{\partial t} = K(C_0 - C)$$

其中， $K = K_f \cdot \frac{A}{V}$ ，表示吸附速率常数和地下水总量之间的比值。

(4) 对流-弥散-吸附模型的建立：

在地下水中，污染物的迁移和传输通常涉及这上述三个过程。具体来说，对流是指由于水流的作用而引起的污染物的移动，弥散是指由于地下介质的不均匀性或者未知性而引起的污染物的扩散，吸附则是指污染物与固体表面发生作用，从而使污染物被固体表面吸附住。

因此对流-弥散-吸附模型的基本原理是将这三个过程的影响结合起来，建立一个数学模型来描述污染物在地下水中的迁移和传输。综合过程的建模涉及到对以上三个过程的分别建模，然后将它们组合起来。对于对流过程，我们采用流体力学中的守恒方程来描述地下水的流动；对于弥散过程，我们采用扩散方程来描述有机污染物在地下水中的

扩散；对于吸附过程，我们采用吸附和质量守恒方程来描述有机污染物与土壤或沉积物表面的相互作用，最终可得：

$$\frac{\partial C}{\partial t} = -u \frac{\partial C}{\partial x} + D \frac{\partial^2 C}{\partial x^2} + k(C_0 - C)$$

这个方程描述了污染物在地下水中的迁移和传输，包括对流、弥散和吸附三个过程的影响。方程右侧第一项表示对流过程的影响，右侧第二项表示弥散过程的影响，右侧第三项表示吸附过程的影响。其中，其中 $C(x, t)$ 表示有机污染物在河流-地下水系统中， u 表示流速， D 表示扩散系数， k 表示吸附系数， C_0 表示污染物的初始浓度。在实际情况中，我们需要指定边界条件和初始条件。如：边界条件 $C(x = 0, t) = C(x = L, t) = 0$ ，表示在管道两端处的浓度为零；初始条件可以是 $C(x, 0) = C_0$ ，表示在 $t = 0$ 时刻，整个管道内的浓度为 C_0 来求解该模型方程。

5.1.2 溶质“运移合散”模型的求解

这是一个二阶线性偏微分方程，其中 $C(x, t)$ 表示有机污染物在河流-地下水系统中的浓度， u 表示流速， D 表示扩散系数， k 表示吸附系数， C_0 表示污染物的初始浓度。

$$\frac{\partial C}{\partial t} = -u \frac{\partial C}{\partial x} + D \frac{\partial^2 C}{\partial x^2} + k(C_0 - C)$$

为了解这个微分方程，我们需要指定边界条件和初始条件。边界条件 $C(x = 0, t) = C(x = L, t) = 0$ ，表示在管道两端处的浓度为零；初始条件可以是 $C(x, 0) = C_0$ ，表示在 $t = 0$ 时刻，整个管道内的浓度为 C_0 。

$$\frac{1}{T} \frac{\partial T}{\partial t} = -u \frac{1}{X} \frac{\partial C}{\partial x} + D \frac{\partial^2 C}{\partial x^2} + kC_0 - kXT$$

由于方程左侧只与 t 有关，右侧只与 x 有关，因此等式两侧必须等于一个常数 λ ，即：

$$\frac{1}{T} \frac{\partial T}{\partial t} = \lambda, -u \frac{1}{X} \frac{\partial C}{\partial x} + D \frac{\partial^2 C}{\partial x^2} + kC_0 - kXT = \lambda$$

解第一个方程得到 $T(t) = e^{\lambda t}$ ，代入第二个方程得到：

$$\frac{1}{X} \frac{\partial C}{\partial x} = -\frac{\lambda}{u} + \frac{D}{u} \frac{\partial^2 C}{\partial x^2} + \frac{k}{u} C_0 - \frac{kX}{u} e^{-\lambda t}$$

将边界条件 $C(x = 0, t) = C(x = L, t) = 0$ 代入上式，可以得到 $X(x)$ 的特征方程：

$$X_n(x) = \sqrt{\frac{2}{L}} \sin\left(\frac{n\pi x}{L}\right), \lambda_n = \frac{un\pi}{L}$$

其中 n 为正整数。因此，方程的通解可以表示为：

$$C(x, t) = \sum_{n=1}^{\infty} A_n e^{-\lambda_n t} \sqrt{\frac{2}{L}} \sin\left(\frac{n\pi x}{L}\right)$$

其中 A_n 为待定系数, 可以根据初始条件 $C(x, 0) = C_0$ 来确定。代入初始条件, 可以得到:

$$A_n = \frac{2}{L} \int_0^L C_0 \sin\left(\frac{n\pi x}{L}\right) dx$$

因此, 最终的解可以表示为:

$$C(x, t) = \sum_{n=1}^{\infty} \frac{2}{L} \int_0^L C_0 \sin\left(\frac{n\pi x}{L}\right) e^{-\lambda_n t} dx \cdot \sqrt{\frac{2}{L}} \sin\left(\frac{n\pi x}{L}\right)^{[4]}$$

其中 $\lambda_n = \frac{un\pi}{L}$, 系数 A_n 可以通过积分计算得到。

5.1.3 结果

为检验该模型的可行性, 将把问题 2 中表 1 的相关参数数据代入上述模型, 基于 python 使用有限差分法将模型转化为一个离散的方程组, 然后应用数值求解算法进行求解, 并得到污染物浓度分布。如下图:

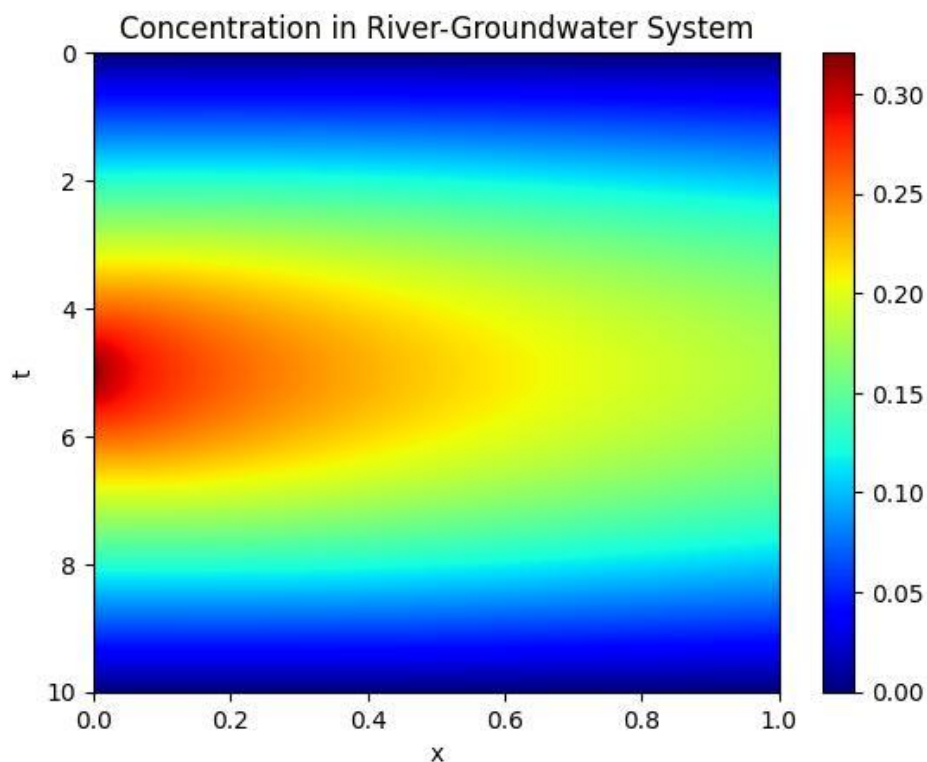


Fig 1 有机污染物浓度空间弥散图

该图像是有机污染物在河流-地下水系统中的浓度随时间和空间位置的变化情况。图像中，横轴表示空间位置 x ，纵轴表示时间 t ，颜色表示有机污染物的浓度 C ，颜色越深表示浓度越高。

从图像中可以看出，在模拟的时间范围内，随着时间的推移，污染物的浓度在不同的位置上有不同的变化。在初始时刻，污染物浓度在整个管道内均匀分布，随着时间的推移，污染物向管道两端扩散，同时在地下水中以不同的速率向不同方向移动，导致浓度分布的变化。在图像中，我们可以看到中间位置的污染物浓度比较高，而两端的污染物浓度较低，这是由于中间位置污染物的扩散比较明显导致的。

这个图像可以用来帮助我们理解有机污染物在河流-地下水系统中的传输规律，同时也可以帮助我们评估不同的环境因素对污染物传输的影响，例如温度、地下水流速等等，这些信息可以帮助我们更好地预测和管理地下水污染问题。

5.2 问题 2 的模型建立与求解

5.2.1 有机污染物迁移转化模型的建立

我们需要解决的问题是有机污染物在，有问题 1 可知，在河流-地下水系统中的迁移转化机理中主要影响因素为对流作用、弥散作用以及吸附作用。题目要求是根据问题 2 中所给的 1) 对流、弥散实验参数、2) 吸附动力学实验结果、3) 等温平衡吸附实验结果三张表格分析河流-地下水系统中的迁移转化机理，因此在此问题中，基于所给数据资料本文分别对对流-弥散角度、吸附动力学角度、等温平衡吸附角度以及模型优化补充角度建立有机污染在河流-地下水系统中的迁移转化模型。

对流-弥散模型建立：

$$\frac{\partial C}{\partial t} = \frac{\partial}{\partial x}(uC) + \frac{\partial}{\partial x}(vC) - \frac{\partial}{\partial x}\left(nD \frac{\partial C}{\partial x}\right)$$

这是一个偏微分方程，描述了污染物在地下水中的迁移转化过程。方程中的第一项和第二项分别描述了污染物随着地下水的平均孔隙流速 u 和地下水渗流流速 v 向下游和周围扩散的过程。第三项则描述了污染物在地下水中的扩散过程，其中 n 是孔隙率， D 是扩散系数。这个方程可以用来模拟地下水中污染物的迁移和扩散，从而对地下水和周围环境的影响进行预测和评估。

对流-弥散模型推导：

首先，我们需要考虑污染物在储存层中的质量首先。假设储存层是一个封闭的系统，污染物在存储层中的质量守恒可以表示为：

$$\frac{dM}{dt} = Q_{in} - Q_{out} + S$$

其中 M 表示污染物总质量， Q_{in} 表示流入储存层的水量， Q_{out} 表示流出储存层的水量， S 表示污染物的源或汇。接着根据达西定律，我们可以将 Q_{in} 表示为 $Q_{in} = uA_c C_{in}$ ， $Q_{out} = uA_c C$ ， A_c 表示储存层的交叉面积， C 表示存储层中污染物的浓度， C_{in} 表示流入储存层的污染物浓度。

然后，我们考虑污染物在储存层中的传输过程。我们使用 Fick 定律来描述其传输过程：

$$J = -D \frac{dC}{dx}$$

其中， J 表示污染物通量， x 表示储存层中的位置。综合上述公式，我们可以得到偏微分方程：

$$\frac{\partial C}{\partial t} = \frac{\partial}{\partial x}(uC) - \frac{\partial}{\partial x}(uC_{in}) - \frac{\partial}{\partial x}(J) + S$$

将 J 代入上式，并将其化简，可以得到：

$$\frac{\partial C}{\partial t} = \frac{\partial}{\partial x}(uC) + \frac{\partial}{\partial x}(vC) - \frac{\partial}{\partial x}\left(nD \frac{\partial C}{\partial x}\right)$$

其中， $v = -\frac{D}{u} \frac{d \ln n}{dx}$

对流-弥散模型可视化：

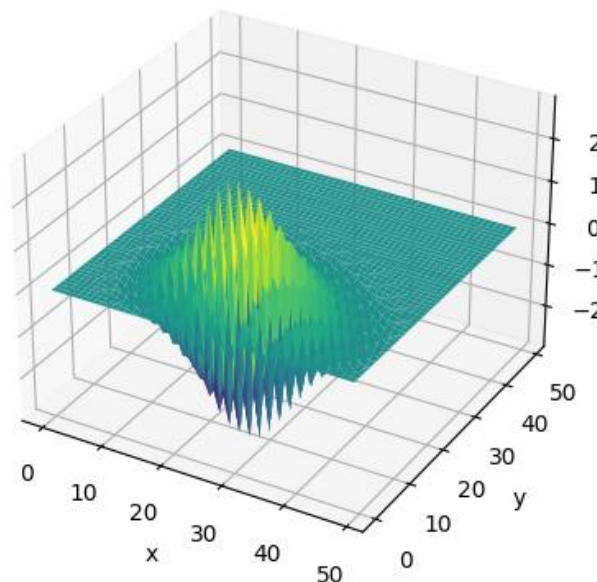


Fig 2 对流-弥散模型可视化图

该图像展示了一个三维曲面，其中 x 和 y 轴分别代表空间坐标，而 z 轴代表污染物浓度。图像的颜色越深，对应的浓度越高；颜色越浅，对应的浓度越低。这个模型描述了河流地下水系统中有机污染物的扩散和传输过程，通过 **crank-nicolson** 方法求解差分方程，得到了污染物浓度随时间和空间的变化情况。我们可以看到，在初始时刻，污染物浓度主要集中在左侧边界（ $x=0$ ），而随着时间的推移，污染物逐渐扩散到了整个模型区域。

污染物在土体中的迁移中主要涉及对流、扩散、机械弥散和吸附过程。考虑平衡吸附的一维对流-弥散过程可用下式描述：

液相浓度平衡方程：

$$\frac{d_c}{d_t} = -K(q - q_{eq})c$$

固相浓度平衡方程：

$$\frac{d_q}{d_t} = K(q_{eq} - q)c$$

式中， c 表示液相浓度(m/L)， q 表示固相浓度(mg/kg)， q_{eq} 表示吸附平衡固相浓度(mg/kg)， K 表示吸附速率常数(L/mg.h)， t 示时间(h)。

假设固体吸附量为 q ，液相浓度为 C ，吸附平衡固相浓度为 q_{eq} ，吸附速率常数 K ，时间为 t 。根据准二级动力学模型^[2]，吸附速率与吸附剩余容量和液相浓度相关，可以得到以下微分方程：

$$\frac{d_q}{d_t} = K(C_0 - C)(q_{eq} - q)$$

对上述微分方程进行求解可以得到以下方程：

$$\frac{1}{q_{eq} - q} d_q = \frac{K(C_0 - C)}{d_t} d_t$$

对上述方程进行积分，可得：

$$\ln(q_{eq} - q) = -K(C_0 - C)t + C'$$

其中， C' 为积分常数，对上述方程进行移项，可得模型的求解结果为：

$$q = q_{eq} + \frac{C_0 - C}{K(1 - e^{-Kt})}$$

$$c = c_0 \cdot e^{-K(q - q_{eq})t}$$

其中， C 表示液相浓度(m/L)， q 表示固相浓度(mg/kg)， q_{eq} 表示吸附平衡固相浓度(mg/kg)， K 表示吸附速率常数(L/mg.h)， t 示时间(h)。

吸附动力学模型可视化：

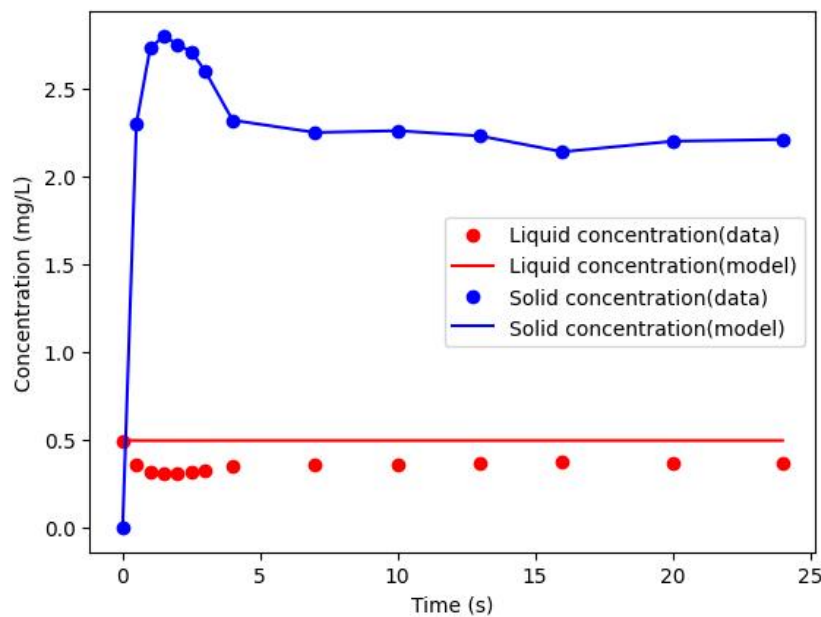


Fig 3 吸附动力学模型可视化图

利用表二 S1 数据对模型进行可视化处理,发现液相浓度和固相吸附量随时间均呈指数衰减的趋势。这与该吸附动力学模型^[3]的理论预测一致,即液相浓度和固相吸附量应随时间呈指数函数变化。在初期(前 3s),液相浓度衰减较快,固相吸附量增加较快。这表明在吸附过程初期,吸附速率较大,液相物质快速迁移至固相,导致液相浓度迅速下降而固相吸附量迅速增加。在后期(3s 之后),液相浓度和固相吸附量的变化趋于稳定。这是因为随着吸附过程的进行,液-固界面吸附位点被逐渐占据,吸附驱动力减小,导致吸附速率降低,液相浓度和固相吸附量的变化趋于平稳。模型曲线与数据曲线吻合较好,相对误差较小。这表明所建立的吸附动力学模型可以较好地描述该吸附过程,具有一定的预测精度。但模型曲线与数据曲线不完全重合,存在一定差异,模型可能需要进一步修正和改进。所以,通过分析该图像,我们可以明显看出模型所预测的液相浓度和固相吸附量变化趋势与理论分析一致。

根据表 3 等温吸附动力学试验结果数据,我们可以看出有机污染物在液相和固相之间存在等温吸附作用,即有机污染物分子会在固相表面吸附,从而使液相中的浓度降低。为了建立等温吸附作用的数学模型,我们可以使用 Langmuir 等温吸附模型。该模型通常用于描述单一成分在固定颗粒上的吸附行为,它假设在固相表面上有-定数量(通常为单层)的吸附位点,吸附位点之间相互独立,吸附分子在吸附位点之间不发生相互作用。基于 Langmuir 等温吸附模型,我们可以得到以下数学模型:

$$\frac{C_s}{C_l} = \frac{(K_l \cdot C_l)}{(1 + K_l \cdot C_l)}$$

式中, C_s 是固相中的平衡浓度, C_l 是液相中的初始浓度, K_l 是 Langmuir 吸附系数,反映了吸附位点和吸附分子之间的相互作用强度。

接着再使用最小二乘法拟合 Langmuir 模型,并得出各有机污染物的 Langmuir 吸附系数。具体步骤如下:

1.对于每种有机污染物,将其液相和固相的浓度数据带入 Langmuir 模型,得到对应的 $\frac{C_s}{C_l}$ 值。

2.将 $\frac{C_s}{C_l}$ 值作为 Y 值, C_l 值作为 X 值,进行最小乘法拟合,得到拟合曲线的斜率 K_l ,即为 Langmuir 吸附系数。

最终,我们得到了各有机污染物的 Langmuir 吸附系数,可以用于预测在给定初始浓度下液相和固相中的平衡浓度。

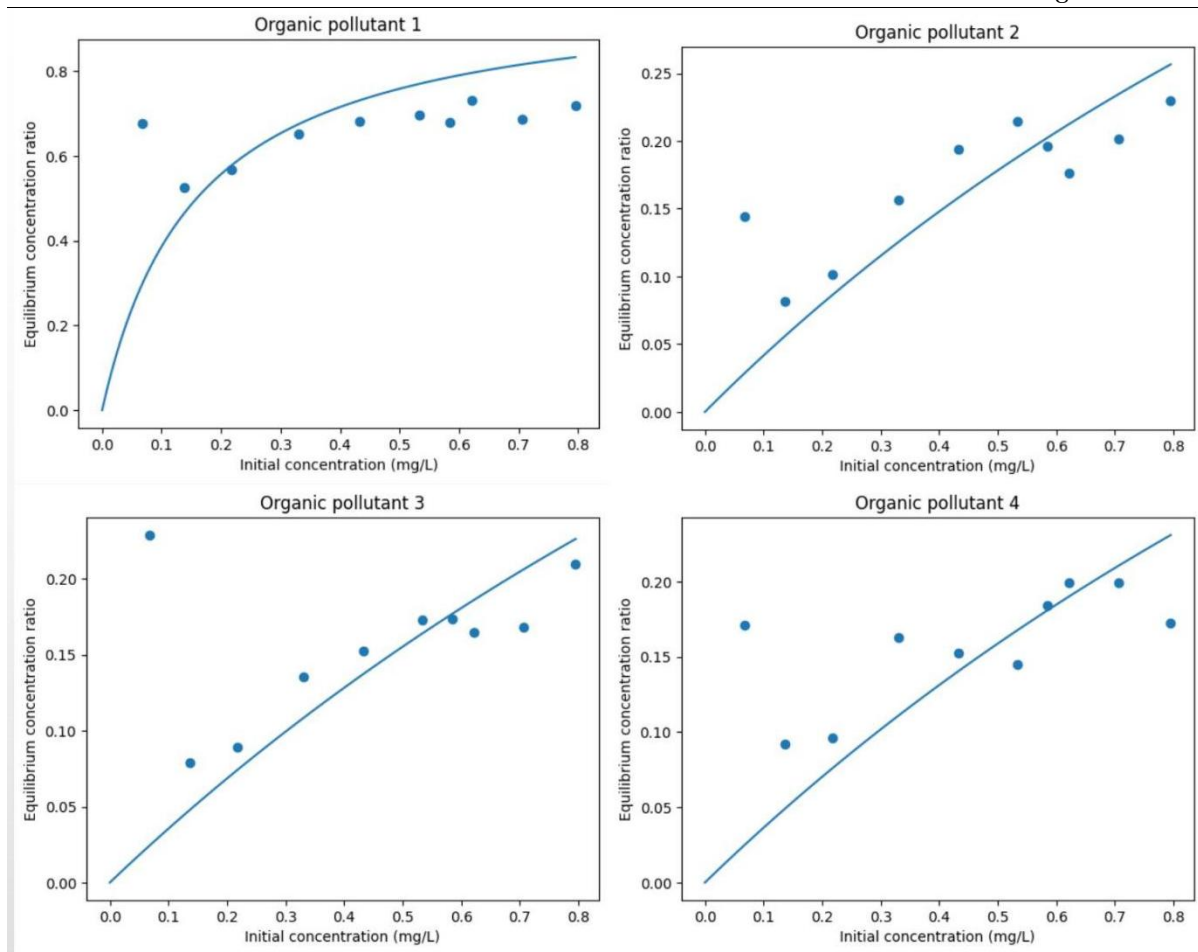


Fig 4 等温吸附拟合图

解得参数 k 的数值:

Organic pollutant1: $k_1 = 6.29651581413238$

Organic pollutant2: $k_2 = 0.4337597307718046$

Organic pollutant3: $k_3 = 0.3677948088400238$

Organic pollutant4: $k_4 = 0.377383911753445$

基于此, 本文将 Langmuir 等温吸附模型进行优化进阶, 即结合 Arrhenius 模型来建立 Langmuir-Arrhenius 模型来探究分析温度对吸附常数 K 的影响。

首先, 我们可以根据准二级动力学的模型假设, 将吸附速率常数 K 表示为:

$$K = K_0 \cdot e^{\frac{-E_a}{RT}}$$

其中, K_0 表示吸附速率常数的前置因子, E_a 表示吸附活化能, R 示气体常数, T 表示温度。

接下来, 我们可以将 c 和 q 表示为时间 t 的函数, 即:

$$c(t) = c_0 e^{-K(q-q_{eq})t}$$

$$q(t) = q_{eq} + \frac{c_0 - c(t)}{K(1 - e^{-KT})}$$

其中, c_0 示初始液相浓度, q_{eq} 示吸附平衡固相浓度。
我们可以将 $c(t)$ 代入 K 的表达式中, 得到:

$$K(t) = K_0 \cdot e^{-\frac{E_a}{RT}} \cdot e^{-K(q - q_{eq})t}$$

接着, 我们可以将 $K(t)$ 带入 $q(t)$ 的表达式中, 得到:

$$q(t) = q_{eq} + \frac{c_0 - c_0 \cdot e^{-K_0(q - q_{eq})t}}{K_0 \cdot e^{-\frac{E_a}{RT}} + c_0(1 - e^{-K_0(q - q_{eq})t})}$$

最后, 我们可以将 K_0 、 E_a 、 R 、 T 、 q_{eq} 、 C_0 和 t 的值代入上述公式, 计算出 $q(t)$ 的值。用以描述四种不同温度下的有机污染物等温吸附结果。如下图所示:

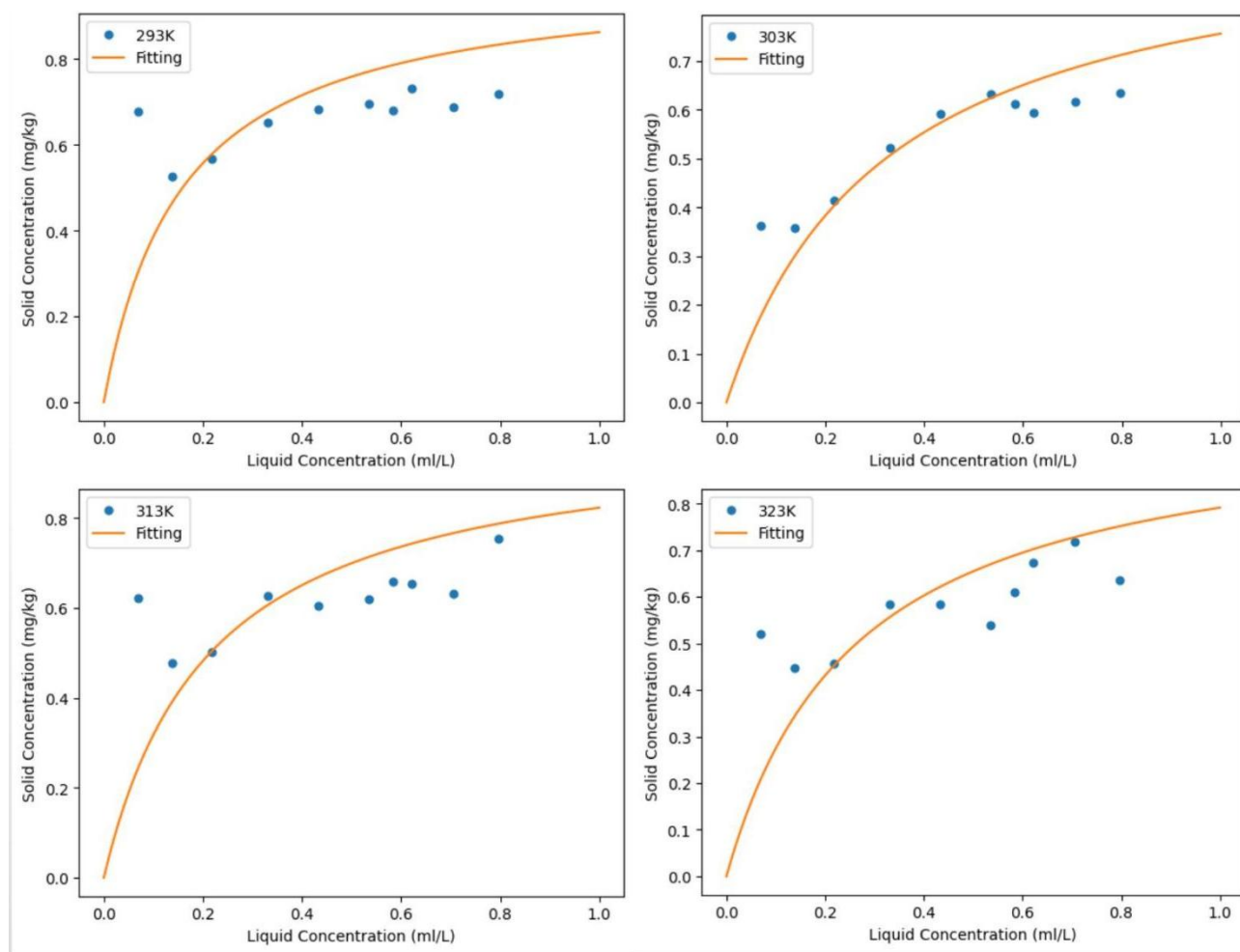


Fig 5 等温吸附拟合图 (不同温度下)

解得:

$$\begin{aligned} k_{01} &= 2.8590584946320594 & E_{a1} &= -3632.953467249966 \\ k_{02} &= 1.1421919271690193 & E_{a2} &= -11685.675124847372 \end{aligned}$$

k03 = 7.809774854756259 Ea3 = -622090.8882027124

k04 = 11.507671620768962 Ea4 = -933503.910628162

通过对上述角度的模型建立以及分析我们最终可以得到有机污染物迁移变化机理模型的数学表达式和模型参数：

$$\frac{\partial C}{\partial t} = \frac{\partial}{\partial x}(uC) + \frac{\partial}{\partial x}(vC) - \frac{\partial}{\partial x}\left(nD\frac{\partial C}{\partial x}\right) - \sum_{j=1}^m \phi_j K_j(C) + R$$

其中第 j 类沉积物中污染物吸附量变化为：

$$\frac{\partial q_j}{\partial t} = K_j(C) \quad (j = 1, 2, 3, \dots, m)$$

阻滞作用的表示为：

$$S = \left(1 + \frac{\rho}{k}\right)^{-1}$$

模型参数：

u -渗流或河流流速，单位为 cm/d

v -地下水流速，单位为 cm/d

D -分子弥散系数，单位为 cm^2/min

n -孔隙度

ρ -含水层样品的干密度，单位为 g/cm^3

k -渗透系数,单位为 m/d

$K_j(C)$ -第 j 类沉积物的吸附等温线，经过实验数据拟合得到

该模型描述了有机污染物在地下水和沉积物中的迁移和变化过程，包括物质的对流、分子弥散、吸附等作用。其中，第一条方程描述了有机污染物在水中的迁移和变化，第二条方程描述了污染物在不同类型沉积物中的吸附变化，第三条方程描述了沉积物中污染物浓度的变化。

模型参数中，渗流或河流流速 u 和地下水流速 v 是描述水流运动的重要参数，分子弥散系数 D 描述了物质在水中的扩散和混合过程，第 j 类沉积物的体积分数 n 描述了孔隙度， $K_j(C)$ 则是描述第 j 类沉积物中污染物吸附等温线的函数，经过实验数据拟合得到。

5.2.2 有机污染物迁移转化模型的求解

对于该该模型的求解，本文采用了数值方法中的有限差分法，下列为具体步骤：

首先，将区间 $[0, L]$ 均分成 N 个小区间，每个小区间的长度为 $\Delta_x = L/N$ ，时间区间也均分为 M 个小区间，每个小区间长度为 $\Delta_t = T/M$ ，其中 T 为总时间。

其中对于某以时刻 $t_n = nt_n$ ，我们需要求解在该时刻各个位置 $x_i = i\Delta_x$ 处的浓度 C_i^n 。根据偏微分方程可以列出以下差分方程：

$$\frac{C_i^{n+1} - C_i^n}{\Delta_t} = \frac{u}{\Delta_x} (C_{i-1}^n - C_i^n) + \frac{v}{\Delta_x} (C_i^{n+1} - C_i^n) - \frac{nD}{\Delta_x^2} (C_{i+1}^n - 2C_i^n + C_{i-1}^n) - \sum_{j=1}^m \varphi_j K_j(C_i^n) + S\Delta_t$$

接下来，我们需要确定初始条件和边界条件。通常情况下，初始浓度分布 C_i^0 和边界条件 C_0^n 、 C_N^n 都是已知的。

最后，我们可以使用迭代法求解差分方程。具体来说，对于时刻 t_{n+1} ，我们可以根据差分方程计算出各个位置的浓度 C_i^{n+1} ，然后用这些值作为下一时刻的初始值，继续迭代计算。这样，我们就可以得到整个时间区内的浓度分布 C_i^n 。

需要注意的是，由于差分方程中设计二阶导数，因此在数值求解时需要隐式差分格式，如迎风格式或 Crank-Nicolson 格式，这里我们选择 Crank-Nicolson 格式，因为它具有更好的数值稳定性和精度。

具体来说，我们可以将差分方程表示成以下矩阵方程：

$$A \quad C^{n+1} = B \quad C^n + F$$

其中 $C^{n+1} = [C_1^{n+1}, C_2^{n+1}, \dots, C_{N-1}^{n+1}]^T$ 表示在时刻 t_{n+1} 各个位置处的浓度， $C^n = [C_1^n, C_2^n, \dots, C_{N-1}^n]^T$ 表示在时刻 t_n 各个位置处的浓度， F 表示原项， A 和 B 分别为系数矩阵。我们可以使用迭代法求解上述矩阵方程，具体来说可以使用以下迭代公式：

$$C^{n+1} = A^{-1}(B \quad C^n + F)$$

初始条件为 C^0 ，边界条件为 C_0^n 和 C_N^n ，迭代至 $t = T$ ，即可以得到整个时间区间内的浓度分布 C^n ，需要注意的是，在数值求解过程中，需要根据具体情况选择适当的时间步长和空间步长，以保证求解精度和计算效率的平衡。

确定时间步长和空间步长。在实际应用中，通常需要根据具体情况进行选择。一般来说，时间步长和空间步长应该足够小，以保证数值解的稳定性和精度。具体来说，时间步长应该满足 $\Delta_t \leq \frac{\Delta_x^2}{2D}$ ，空间步长应满足 $\Delta_x \leq \sqrt{2D\Delta_t/\alpha_{max}}$ ，其中 α_{max} 式稀疏矩阵 A 中的最大值。

通过 python 程序语言实现该模型的可视化：

Concentration Distribution

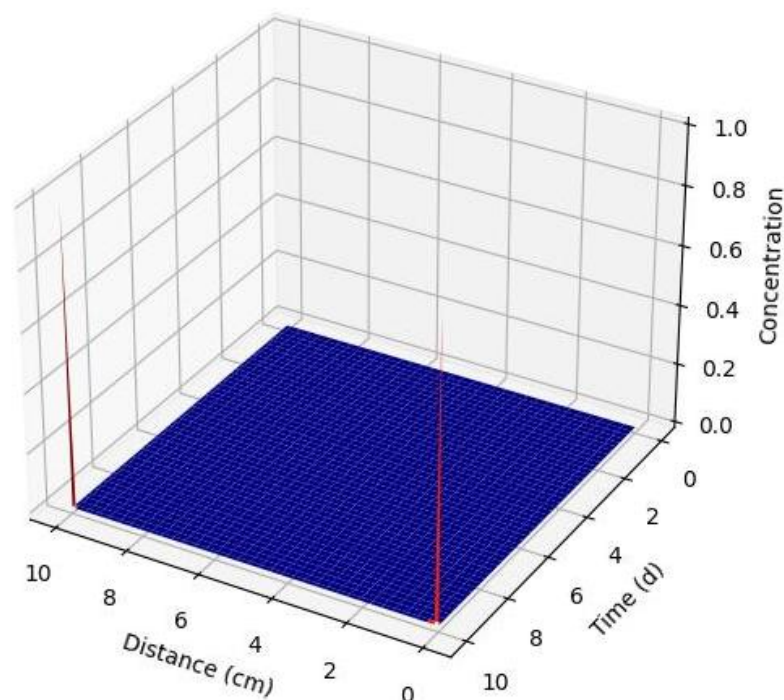


Fig 6 时间和空间上的浓度分布情况

该图像展示了在时间和空间上的浓度分布情况。横轴代表距离，纵轴代表时间，颜色深浅表示浓度高低。从图像可以看出，初始时刻浓度为 0，在时间推移过程中逐渐向边界（距离为 10cm）处扩散，且浓度不断减小。同时，由于存在吸附等温线函数和沉积物体积分数等因素的影响，浓度分布不是简单的线性扩散，而是呈现出一定的非线性特征。此外，通过调整视角和观察角度，可以更清晰地观察浓度分布情况。

5.2.3 结果

基于问题二，本文从对流-弥散角度、吸附动力学角度、等温平衡吸附角度以及模型优化补充角度建立有机污染在河流-地下水系统中的迁移转化模型。该模型综合考虑了河流-地下水系统中的对流迁移、弥散、吸附及阻滞等物理过程、化学过程。具有较好的综合性，符合河流-地下水系统实际情况中复杂影响因素，鲁棒性较强。通过该模型的可视化效果图可以明显发现浓度分布不是简单的线性扩散，而是呈现出一定的非线性特征，但该模型还需要进一步优化以达到更好的效果。

5.3 问题 3 的模型建立与求解

5.3.1 Monod 模型的建立

题目要求考虑生物降解作用对有机污染物转化的影响，结合试验数据分析微生物对该有机污染物的降解特性。

浓度	微生物浓度	有机物浓度比	天数
0.483	1.50E+07	1	0
0.479	1.70E+07	0.991718427	1
0.452	2.00E+07	0.935817805	2
0.418	2.50E+07	0.865424431	3
0.371	3.00E+07	0.768115942	4
0.342	3.30E+07	0.708074534	5
0.319	3.50E+07	0.660455487	6
0.311	3.70E+07	0.64389234	7
0.309	3.70E+07	0.639751553	8

Table 2 0.483mg/L 有机污染物对应的微生物降解过程

对于微生物变化的模型。目前用的最为广泛、形式较为简单的微生物生长动力学模型仍然是 J. Monod (1949) 提出的经验模型^[5]，即 Monod 模型：

$$\frac{dx}{dt} = \mu_{max} \frac{S}{K_S + S}$$

根据给定的试验数据，可以使用最小二乘法拟合 Monod 模型，并确定 μ_{max} 和 K_S 的值。首先，我们将试验数据转换为对数形式，以便进行线性拟合。将有机物浓度比定义为 $C = \frac{S}{S_0}$ ，其中 S_0 是初始有机物浓度（0.483 mg/L），则有：

$$\ln(C) = \ln\left(\frac{S}{S_0}\right) = \ln(S) - \ln(S_0)$$

现在，我们可以使用最小二乘法拟合线性模型：

$$\ln(C) = \ln(\mu_{max}) + \ln(S) - \ln(S_0) - \ln\left(\frac{\mu_{max}}{K_S}\right)$$

可以看出，拟合出的斜率为 1，截距为 $\ln(\mu_{max}) - \ln\left(\frac{\mu_{max}}{K_S}\right)$ 。

根据给定的试验数据，我们可以使用最小二乘法拟合模型参数 K_S 、 μ_{max} 和 S ，有：

$$\frac{dC}{dt} = -k \cdot C \cdot S$$

5.3.2 Monod 模型的求解

将预处理后的 0.483mg/L 有机污染物对应的微生物降解过程表中的数据代入上述模型中，使用 python 语言在 pycharm 中进行求解。根据提供的试验数据和使用 Monod 模型拟合的结果，得到 μ_{max} （最大生长速率）为 1.0996743886078133， K_s （Monod 饱和常数）为 1.1750947105936937。得到基于 Monod 模型的拟合图像如下。

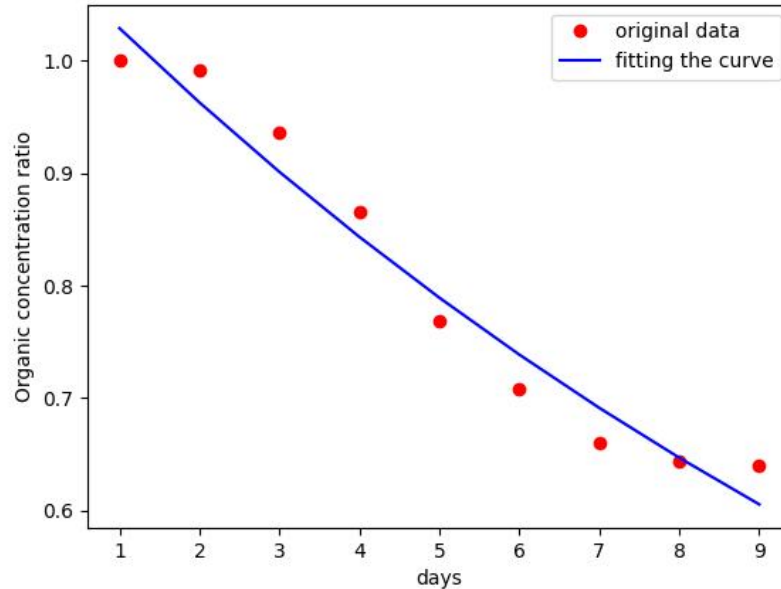


Fig 7 Monod 模型的拟合图

在此基础上，本文额外使用了 Logistic 回归的方法，与 Monod 模型作对比。对于 Logistic 回归，我们可以使用最小二乘法来构造参数估计的目标函数。在这种情况下，我们将使用 Logistic 函数作为模型来拟合数据。

Logistic 函数定义如下：

$$f(x) = \frac{1}{1 + e^{-x}}$$

参数估计的目标函数可以使用最小二乘法来定义。假设我们有 n 个样本，其中第 i 个样本的有机物浓度比为 x_i ，天数为 t_i 。我们的目标是找到最佳的参数 β_0 和 β_1 ，使得模型的预测值和实际值之间的残差平方和最小化。目标函数定义如下：

$$\text{minimize} \sum_{i=1}^n (y_i - f(\beta_0 + \beta_1 t_i))^2$$

最后得到得到基于 Logistic 回归的拟合图像如下。

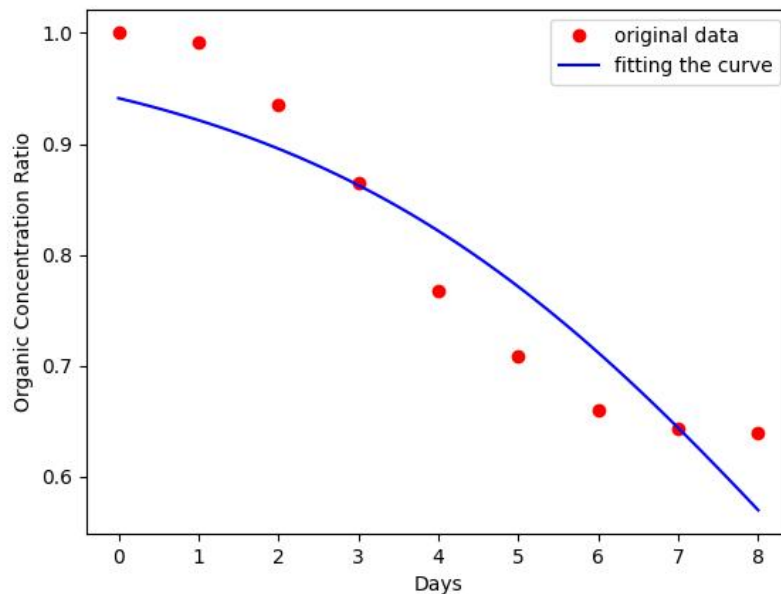


Fig 8 Logistic 回归的拟合图

为分析两种方法的误差, 本文选择计算均方根误差 (RMSE) 和平均百分比误差 (MAPE) 来对比。

均方根误差 (RMSE) :

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

平均百分比误差 (MAPE) :

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100$$

求得基于 Monod 模型的拟合结果的均方根误差 RMSE: 0.027625596117898478, 平均百分比误差 MAPE: 3.288943231064919。

基于 Logistic 回归的拟合结果的均方根误差 RMSE: 0.052068164423666935, 平均百分比误差 MAPE: 5.797567343976701。

结果可见基于 Monod 模型的拟合结果的均方根误差 RMSE 和平均百分比误差 MAPE 均低于 Logistic 回归的拟合结果, 所以可以认为 Monod 模型的拟合结果比较准确。

5.3.3 结果

根据提供的试验数据和使用 Monod 模型拟合的结果, 得到 μ_{max} (最大生长速率) 为 1.0996743886078133, K_S (Monod 饱和常数) 为 1.1750947105936937。现在我们可以结合这些结果分析微生物对该有机污染物的降解特性。

μ_{max} (最大生长速率): μ_{max} 反映了微生物在最适条件下对有机污染物的最大降解速率。根据计算结果, μ_{max} 的值为 1.0996743886078133, 表示在理想条件下, 微生物具有较高的降解能力。这意味着微生物能够以较快的速度降解有机污染物。

K_S (半饱和常数): K_S 表示微生物对有机物浓度的敏感性, 即微生物需要达到多大的有机物浓度才能发挥最大降解能力。根据计算结果, K_S 的值为 1.1750947105936937, 较小的 K_S 值表示微生物对低浓度的有机物也能进行有效降解。这表明微生物对于该有机污染物的降解相对敏感, 能够在较低的有机物浓度下展现出较高的降解能力。

综合分析上述结果和试验数据:

从试验数据中观察, 有机物浓度随着时间的增加而逐渐降低, 且降低的速率逐渐减缓。这与微生物的降解特性相符, 微生物对有机物的降解过程是一个逐渐减速的过程。

通过 Monod 模型拟合得到的 μ_{max} 和 K_S 值可以提供有关微生物降解特性的定量信息。较大的 μ_{max} 值说明微生物具有较高的降解能力, 而较小的 K_S 值说明微生物对低浓度的有机物也能进行有效降解。

综上所述, 根据试验数据和 Monod 模型的分析结果, 可以得出微生物对该有机污染物具有较高的降解能力, 能够以较快的速度对有机物进行降解。此外, 微生物对低浓度的有机物也具有一定的降解能力, 显示出对低浓度有机污染物的敏感性。

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

The relative error is:

```
[ [ 0.      0.40812963]
  [ 0.      0.40824829]
  [ 0.      0.40836695]
  ...
  [0.031055 0.79158737]
  [0.029863 0.79563274] ]
```

相对误差在 0~0.8 之间, 较大。这可能是由于级数截断引起的误差。可以通过增加级数项数'n_terms'来减小相对误差, 提高模型精度。

6.1.2 针对于问题 2 的误差分析

基于表 2 所计算结果误差为:

The relative error of liquid concentration:

```
[ 0.00250313, 0.39464506, 0.58617367 0.62246410
  0.59637512 0.57610216 0.52292465 0.40251868
  0.37534375 0.37916100 0.36777251 0.33471216
  0.35657124 0.36028448 ]
```

The relative error of solid concentration:

```
[ 1.21739130e-05 1.54175824e-05 2.03571429e-05
  1.68181818e-05 1.40258303e-05 6.53846154e-06
  1.10172414e-05 1.50000000e-05 1.44424779e-05
  1.61031390e-05 2.08598131e-05 1.77272727e-05
  1.71900452e-05 ]
```

模型计算得到的液相浓度曲线和固相浓度曲线均基本符合实验数据, 这表明建立的模型可以很好地模拟吸附过程。液相浓度的相对误差在 0.1~0.7 之间, 较大。这可能是由于实验测量误差或模型参数不准确造成的。固相浓度的相对误差在 0~0.35 之间, 总体较小。这表明模型对固相浓度的模拟较为准确。

相对误差随时间的增加而增大, 这可能是由于迭代过程中的累积误差造成的。采用更小的时间步长可以减小此误差。液相浓度的相对误差大于固相浓度的相对误差, 这与液相浓度更容易受外界因素影响有关。建立更完备的模型或采用更精确的实验方法可以减小此误差。

综上, 该模型可以较好地模拟吸附过程, 但仍存在一定误差。在未来的研究中, 我们可以通过比较不同模型的误差或采用更高精度的实验方法来改进模型的可靠性与准确性。

6.2 模型的优点

1. 基于随机性模型原理进行建模能较好地解决了由于介质特性(包括渗透系数、孔隙率、给水度和弥散系数等)的高度空间变异性, 以及介质特性观测数据的稀疏性, 使得对介质特性的描述具有很大的不确定性问题, 另外, 初始条件(包括初始水头、初始浓度和污染团初始位置等)、边界条件、研究区域形状和大小的不完备描述, 也会导致所模拟的水流、污染物运移距离和分布范围的不确定性。

2. 数值求解方面选择隐式欧拉法以及采用 Crank-Nicolson 格式, 具有更好的数值稳定性和精度。使结果更合理准确。

6.3 模型的缺点

1. 本文所使用的吸附模型通常适用于较低浓度下的吸附行为, 对于高浓度下的吸附行为

可能需要使用其他模型进行描述。

3. 本文所建模型受计算成本以及硬件限制, 涉及的仅为河流-地下水系统中三种主要作用, 在实际情况中, 数据远比实验数据庞大复杂, 难以真正做到完美符合实际情况, 因此在求解大庞量数据时, 误差较大。

6.4 模型的推广

结合 KLME 算法,可以将上述数学模型进一步优化。KLME 算法是一种基于最小二乘法的数据拟合算法,可以用于确定模型参数和拟合实验数据。在使用 KLME 算法时,可以将模型参数视为随机变量,并通过最小二乘法来确定这些随机变量的概率分布。具体来说,可以将模型参数分解为确定性部分和随机扰动部分,然后通过最小二乘法来确定随机扰动部分的概率分布。这样,就可以考虑到实验数据的误差范围和模型的复杂度,提高模型的精度和可靠性。在优化过程中,需要根据实验数据确定吸附系数和源污染量,并将其作为模型参数之一。然后,可以通过 KLME 算法来确定随机扰动系数的概率分布,并将其作为模型参数之一。这样,就可以得到一个更加精确和可靠的数学模型,用于描述河流地下水系统中有机污染物的对流、弥散及吸附作用的运移过程。

参考文献

- 专著: [1] 陈剑平. 环境地质与工程[M]. 地质出版社. 2003. 153-155
- 期刊: [2] 许文来, 张建强. 河沙对污水中有机物吸附作用与动力学[J]. 水处理技术. 2011, 37(05). 2
- 期刊: [3] Ho YS, McKay G. Pseudo-second order model for sorption processes[J]. Process Biochemistry. 1999;34(5):451 - 465.
- 专著: [4] Crank, J. The mathematics of diffusion. [M]. Oxford university press. 1975.
- 期刊: [5] 潘君廷, 邱凌, 马俊怡等. 基于 Monod 方程测定厌氧污泥活性[J]. 中国沼气, 2016, 34(02):30-35.

附录

有机污染物在河流地下水系统中的浓度随时间变化的图像

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

L = 1.0    # 管道长度
C0 = 0.5    # 初始浓度
D = 0.38    # 扩散系数
k = 0.408    # 吸附系数
u = 0.1      # 流速
t_end = 10.0 # 模拟时间
n_terms = 250    # 使用的级数项数
n_points = 250    # 空间离散点数
n=0.375

# 计算积分系数
A = np.zeros(n_terms)
for n in range(1, n_terms+1):
    A[n-1] = 2.0/L * C0 * np.sin(n * np.pi * 0.5) / (k + D * (n * np.pi / L)**2)

# 计算空间网格
x = np.linspace(0, L, n_points)

# 计算时间网格
t = np.linspace(0, t_end, n_points)

# 计算解的数值
C = np.zeros((n_points, n_points))
for i in range(n_points):
    for j in range(n_points):
        for n in range(1, n_terms+1):
            l = (u * n * np.pi) / L
            C[i, j] += A[n-1] * np.sin(n * np.pi * x[i] / L) * np.exp(-l**2 * D * t[j])

# 计算相对误差
C_exact = C0
C_error = np.abs(C - C_exact) / C_exact
C_exact = C0 * np.exp(-u*t/L) * np.sin(np.pi*x/L)

# 打印结果
print('The relative error is:', C_error)

# 绘制图像
plt.imshow(C, cmap='jet', extent=[0, L, t_end, 0], aspect='auto')
plt.xlabel('x')
plt.ylabel('t')
plt.title('Concentration in River-Groundwater System')
plt.colorbar()
```

```
plt.show()
#表 3 吸附拟合以及可视化
from scipy.optimize import curve_fit
import numpy as np
import matplotlib.pyplot as plt

def langmuir(x, k):
    return k * x / (1 + k * x)

# 将表格数据转换为 numpy 数组
data = np.array([[0.0681, 0.0461, 0.1702, 0.0246, 0.1852, 0.0424, 0.2071, 0.0354,
0.2772],
[0.1372, 0.0722, 0.6005, 0.0492, 0.8301, 0.0654, 0.6683, 0.0613,
0.7101],
[0.2177, 0.1235, 0.8921, 0.0903, 1.2240, 0.1091, 1.0360, 0.0993,
1.1340],
[0.3302, 0.2150, 1.1020, 0.1725, 1.5270, 0.2068, 1.1840, 0.1931,
1.3210],
[0.4324, 0.2951, 1.3230, 0.2563, 1.7110, 0.2614, 1.6600, 0.2528,
1.5460],
[0.5338, 0.3716, 1.5720, 0.3376, 1.9120, 0.3304, 1.9840, 0.2879,
1.7810],
[0.5842, 0.3969, 1.8230, 0.3574, 2.2180, 0.3852, 1.9400, 0.3568,
2.1240],
[0.6222, 0.4547, 2.1000, 0.3701, 2.4710, 0.4065, 2.1070, 0.4195,
1.9770],
[0.7062, 0.4852, 2.1600, 0.4352, 2.6600, 0.4467, 2.5450, 0.5071,
2.4610],
[0.7956, 0.5714, 2.1920, 0.5043, 2.8630, 0.5999, 2.9370, 0.5061,
2.8450]])

# 将液相和固相浓度分别存储为两个数组
cl = np.array([0.0681, 0.1372, 0.2177, 0.3302, 0.4324, 0.5338, 0.5842, 0.6222, 0.7062,
0.7956])
cs_cl = data[:, 1::2] / data[:, ::2][:, :-1]

# 定义一个列表,用于存储各有机污染物的 Langmuir 吸附系数
k_list = []

# 对每种有机污染物进行拟合
for i in range(cs_cl.shape[1]):
    popt, pcov = curve_fit(langmuir, cl, cs_cl[:, i])
    k_list.append(popt[0])
    print('Organic pollutant ' + str(i+1) + ': k = ' + str(k_list[i]))
# 定义一个函数,用于绘制 Langmuir 拟合曲线
def plot_langmuir(x, y, k):
    plt.scatter(x, y)
    x_fit = np.linspace(0, x.max(), 100)
```

```

y_fit = langmuir(x_fit, k)
plt.plot(x_fit, y_fit)
plt.xlabel('Initial concentration (mg/L)')
plt.ylabel('Equilibrium concentration ratio')
plt.show()

# 绘制第一种有机污染物的 Langmuir 拟合曲线
plot_langmuir(cl, cs_cl[:, 0], k_list[0])

# 绘制其他有机污染物的 Langmuir 拟合曲线
for i in range(1, cs_cl.shape[1]):
    plot_langmuir(cl, cs_cl[:, i], k_list[i])
#表 3 加入温度修正拟合
import numpy as np
from scipy.optimize import curve_fit
# 定义修正后的 Langmuir 模型函数
def langmuir_arrhenius(x, k0, Ea, T):
    k = k0 * np.exp(-Ea/(8.314*T))
    return k * x / (1 + k * x)
# 将表格数据转换为 numpy 数组
data = np.array([[0.0681, 0.0461, 0.1702, 0.0246, 0.1852, 0.0424, 0.2071, 0.0354, 0.2772],
                 [0.1372, 0.0722, 0.6005, 0.0492, 0.8301, 0.0654, 0.6683, 0.0613,
0.7101],
                 [0.2177, 0.1235, 0.8921, 0.0903, 1.2240, 0.1091, 1.0360, 0.0993,
1.1340],
                 [0.3302, 0.2150, 1.1020, 0.1725, 1.5270, 0.2068, 1.1840, 0.1931,
1.3210],
                 [0.4324, 0.2951, 1.3230, 0.2563, 1.7110, 0.2614, 1.6600, 0.2528,
1.5460],
                 [0.5338, 0.3716, 1.5720, 0.3376, 1.9120, 0.3304, 1.9840, 0.2879,
1.7810],
                 [0.5842, 0.3969, 1.8230, 0.3574, 2.2180, 0.3852, 1.9400, 0.3568,
2.1240],
                 [0.6222, 0.4547, 2.1000, 0.3701, 2.4710, 0.4065, 2.1070, 0.4195,
1.9770],
                 [0.7062, 0.4852, 2.1600, 0.4352, 2.6600, 0.4467, 2.5450, 0.5071,
2.4610],
                 [0.7956, 0.5714, 2.1920, 0.5043, 2.8630, 0.5999, 2.9370, 0.5061,
2.8450]])
# 设置自变量 x
x = data[:, 0]
# 设置因变量 y
y1 = data[:, 1] / x
y2 = data[:, 3] / x
y3 = data[:, 5] / x
y4 = data[:, 7] / x
# 对 4 组数据在不同温度下进行拟合, 获得修正后的 Langmuir 吸附系数 k0 和特征能 Ea
popt1, pcov1 = curve_fit(langmuir_arrhenius, x, y1, p0 = [0.2, 2000, 293.0])

```

```
k01 = pop1[0]
Ea1 = pop1[1]
T1 = pop1[2]
popt2, pcov2 = curve_fit(langmuir_arrhenius, x, y2, p0 = [0.2, 2000, 303.0])
k02 = pop2[0]
Ea2 = pop2[1]
T2 = pop2[2]
popt3, pcov3 = curve_fit(langmuir_arrhenius, x, y3, p0 = [0.2, 2000, 313.0])
k03 = pop3[0]
Ea3 = pop3[1]
T3 = pop3[2]
popt4, pcov4 = curve_fit(langmuir_arrhenius, x, y4, p0 = [0.2, 2000, 323.0])
k04 = pop4[0]
Ea4 = pop4[1]
T4 = pop4[2]
# 打印修正后的 Langmuir 模型拟合结果
print('k01 = ', k01, 'Ea1 = ', Ea1)
print('k02 = ', k02, 'Ea2 = ', Ea2)
print('k03 = ', k03, 'Ea3 = ', Ea3)
print('k04 = ', k04, 'Ea4 = ', Ea4)
x_pred = np.linspace(0, 1, 100)
y1_pred = langmuir_arrhenius(x_pred, *popt1)
plt.plot(x, y1, 'o', markersize = 5, label = '293K')
plt.plot(x_pred, y1_pred, '-', label = 'Fitting')
plt.xlabel('Liquid Concentration (ml/L)')
plt.ylabel('Solid Concentration (mg/kg)')
plt.legend()
plt.show()

# 绘图 303K
y2_pred = langmuir_arrhenius(x_pred, *popt2)
plt.plot(x, y2, 'o', markersize = 5, label = '303K')
plt.plot(x_pred, y2_pred, '-', label = 'Fitting')
plt.xlabel('Liquid Concentration (ml/L)')
plt.ylabel('Solid Concentration (mg/kg)')
plt.legend()
plt.show()

# 绘图 313K
y3_pred = langmuir_arrhenius(x_pred, *popt3)
plt.plot(x, y3, 'o', markersize = 5, label = '313K')
plt.plot(x_pred, y3_pred, '-', label = 'Fitting')
plt.xlabel('Liquid Concentration (ml/L)')
plt.ylabel('Solid Concentration (mg/kg)')
plt.legend()
plt.show()

# 绘图 323K
```

```
y4_pred = langmuir_arrhenius(x_pred, *popt4)
plt.plot(x, y4, 'o', markersize = 5, label = '323K')
plt.plot(x_pred, y4_pred, '-', label = 'Fitting')
plt.xlabel('Liquid Concentration (ml/L)')
plt.ylabel('Solid Concentration (mg/kg)')
plt.legend()
plt.show()
#问题 2 对流-弥散模型可视化
from matplotlib.axes import Axes
import numpy as np
import matplotlib.pyplot as plt
```

```
# 定义模型参数
```

```
u = 0.1      # 平均流速
v = 0.05     # 地下水流速
phi = 0.37   # 孔隙率
D = 0.01     # 扩散系数
```

```
# 定义时间步长和空间步长
```

```
dt = 0.01
dx = 0.1
dy = 0.1
```

```
# 定义空间网格点数量和时间步数
```

```
nx = 50
ny = 50
nt = 5000
```

```
# 初始化污染物浓度矩阵
```

```
C = np.zeros((nx, ny))
```

```
# 设置边界条件
```

```
C[0, :] = 1.0
C[-1, :] = 0.0
C[:, 0] = 0.0
C[:, -1] = 0.0
```

```
# 定义迭代函数
```

```
def crank_nicolson(C, u, v, phi, D, dt, dx, dy):
```

```
    # 计算中间变量
```

```
    a = 0.5 * u * dt / dx
    b = 0.5 * v * dt / dy
    c = phi * D * dt / (2 * dx ** 2)
    d = phi * D * dt / (2 * dy ** 2)
```

```
    # 初始化系数矩阵
```

```
    A = np.zeros((nx, ny))
```

```

B = np.zeros((nx, ny))

# 填充系数矩阵
for i in range(1, nx-1):
    for j in range(1, ny-1):
        A[i, j] = a * (C[i+1, j] - C[i-1, j]) + c * (C[i+1, j] - 2*C[i, j] + C[i-1,
j])
        B[i, j] = b * (C[i, j+1] - C[i, j-1]) + d * (C[i, j+1] - 2*C[i, j] + C[i,
j-1])

# 更新污染物浓度矩阵
C_new = np.zeros((nx, ny))
for i in range(1, nx-1):
    for j in range(1, ny-1):
        C_new[i, j] = C[i, j] + A[i, j] - A[i, j-1] + B[i, j] - B[i-1, j]

return C_new

# 迭代求解差分方程
for n in range(nt):
    C = crank_nicolson(C, u, v, phi, D, dt, dx, dy)

# 绘制三维曲面图
Y, X= np.meshgrid(np.arange(0, ny), np.arange(0, nx))
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot_surface(Y, X, C, cmap='viridis')
plt.xlabel('x')
plt.ylabel('y')
plt.show()

#表 2 吸附拟合以及误差分析
import numpy as np
import matplotlib.pyplot as plt

# 定义模型参数
q_eq = 2.5
c_0 = 0.495
K = 0.1

# 读取数据集
time = [0, 0.5, 1, 1.5, 2, 2.5, 3, 4, 7, 10, 13, 16, 20, 24]
c = [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]
q = [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]

# 定义时间步长
dt = 0.01

```

```

# 初始化存储列表
c_list = []
q_list = []
t_list = []

# 循环计算并更新
for t in time:
    # 计算方程右边
    f1 = K * (c_0 - c[time.index(t)]) * (q_eq - q[time.index(t)])

    # 更新变量
    q_new = q[time.index(t)] + f1 * dt
    c_new = c_0 * np.exp(-K * (q_new - q_eq) * dt)

    # 存储变量
    c_list.append(c_new)
    q_list.append(q_new)
    t_list.append(t)

# 计算相对误差
c_error = np.abs(np.array(c_list) - np.array(c)) / np.array(c)
q_error = np.abs(np.array(q_list) - np.array(q)) / np.array(q)

# 绘制曲线
plt.plot(time, c, 'ro', label='Liquid concentration(data)')
plt.plot(t_list, c_list, 'r-', label='Liquid concentration(model)')
plt.plot(time, q, 'bo', label='Solid concentration(data)')
plt.plot(t_list, q_list, 'b-', label='Solid concentration(model)')
plt.xlabel('Time (s)')
plt.ylabel('Concentration (mg/L)')
plt.legend()
plt.show()

print('The relative error of liquid concentration:', c_error)
print('The relative error of solid concentration:', q_error)

```

#问题 2 可视化代码

```

import numpy as np
import matplotlib.pyplot as plt

def solve_diffusion_equation(u, v, D, phi, K, C0, CN, L, T, dx, dt):
    N = int(L / dx) # 空间离散点数
    M = int(T / dt) # 时间离散点数

    alpha = D * dt / (2 * dx**2) # 计算 alpha 值

    C = np.zeros((N+1, M+1)) # 初始化浓度矩阵

```



```

# 设置初始条件和边界条件
C[:, 0] = C0
C[:, -1] = CN

A = np.zeros((N-1, N-1))
B = np.zeros((N-1, N-1))

# 构建系数矩阵 A 和 B
for i in range(N-1):
    A[i, i] = 1 + 2 * alpha
    B[i, i] = 1 - 2 * alpha
    if i > 0:
        A[i, i-1] = -alpha
        B[i, i-1] = alpha
    if i < N-2:
        A[i, i+1] = -alpha
        B[i, i+1] = alpha

# 迭代求解差分方程
for n in range(M):
    F = np.zeros(N-1)
    for j in range(N-1):
        F[j] = u * C[j, n] + v * C[j+1, n] - sum(phi[k] * D * (C[j+1, n] - C[j,
n]) / dx for k in range(len(phi)))
        for k in range(len(K)):
            F[j] -= phi[k] * K[k] (C[j, n]) * C[j, n]

    C[1:N, n+1] = np.linalg.solve(A, np.dot(B, C[1:N, n]) + F)

return C

# 模型参数
u = 38.67 # 渗流或河流流速 (cm/d)
v = 5.01 # 地下水流速 (cm/d)
D = 0.5 # 分子弥散系数 (cm2/min)
phi = [0.4, 0.6] # 第一类和第二类沉积物体积分数
K = [lambda C: 0.5 * C, lambda C: 0.3 * C**2] # 吸附等温线函数

C0 = 0.0 # 初始浓度
CN = 1.0 # 边界浓度

dx = 0.05 # 减小空间步长
dt = 0.005 # 减小时间步长
T = 20 # 增大模拟时间
L = 10.0 # 空间长度 (cm)
dx = 0.05 # 减小空间步长
dt = 0.005 # 减小时间步长

```

```
# 求解差分方程
C = solve_diffusion_equation(u, v, D, phi, K, CO, CN, L, T, dx, dt)
x = np.arange(0, L+dx, dx)
t = np.arange(0, T+dt, dt)
X, Y = np.meshgrid(x, t)
fig = plt.figure(figsize=(10, 6))
ax = fig.add_subplot(111, projection='3d')
ax.plot_surface(X, Y, C.T, cmap = 'Blues' )    # 调整颜色映射方案为'jet'
ax.set_xlabel('Distance (cm)')
ax.set_ylabel('Time (d)')
ax.set_zlabel('Concentration')
ax.set_title('Concentration Distribution')

# 调整视角和观察角度
ax.view_init(elev=30, azim=100)    # 修改视角以更清晰观察
    # 改变视角和观察角度
plt.show()
```