

基于回归分析的扩散系数识别模型

摘要

本文主要研究了物质扩散模型参数识别问题，根据 Fick 第一、第二定律，利用多项式回归拟合、偏微分方程、目标优化等方法，对物质扩散组元建立函数关系，计算求解得到扩散系数，最后分析误差水平、实际测得数据量，得到了影响扩散系数识别精度的主要因素，并推广到三维建立可视化模型。

针对问题一：在一维情况下，根据 Fick 第一、第二定律，建立多项式回归拟合模型，利用已知不同时刻扩散通量 J 的测量值（附件一）与不同时刻组元浓度 C 的测量值（附件二），可识别不同位置的扩散系数 D ，得到 D 随 C 的变化曲线（图 2-6）。扩散较稳定后， D 在 5.3536×10^{-2} 附近波动。

针对问题二：

对于第一小问：在一维情况下，当组元的扩散通量 J 未知，浓度 C 的空间分布已知时，根据 Fick 第一、第二定律，建立关于浓度 C 的偏微分方程。考虑扩散系数趋于稳定，扩散特征趋于稳定，为简化计算，等间隔抽取二十一组已知数据（附件二），进行多项式回归拟合，得到 D 随 C 的变化曲线（图 8），扩散较稳定后， D 在 5.3481×10^{-2} 附近波动。

对于第二小问：当组元的扩散通量 J 已知，浓度 C 的空间分布未知时，根据 Fick 第一定律对已知扩散通量数据（附件一）逆推浓度梯度，利用最小二乘法拟合与目标优化算法得到 D 随 C 的变化曲线。

对比问题一与问题二两种情况的 $D-C$ 曲线， J 未知， C 的空间分布已知时与问题一的结果更接近。

针对问题三：

针对问题三，观察前两题得到的 $D-C$ 散点图，根据上网查到的资料，分析异常的扩散系数数据点是由实验测量误差、实验测量位置、实验测量时刻、实验测量数据量等因素导致的。通过预处理不合理的数据点、选择相对稳定的实验测量位置和实验测量时间、尽可能多地收集数据能提高扩散系数的估计精度。

针对问题四：对于实际问题，根据二维 Fick 定律，建立二维扩散模型，对扩散项采用中心差分格式处理并利用 python 编程实现。在扩散模型中进行迭代方法更新浓度值。在每次迭代通过离散化的扩散方程计算新的 C ，边界条件被设定为固定的浓度值。对于离散方程与微分方程之间的误差，利用耗散效应把尖锐的解抹平，使结果光滑，在不可避免的误差情况下提高解的光滑性，避免奇点。结果以三维可视化展示了浓度随时间变化的动态过程。此模型可根据需要预设扩散系数及其他参数，利用 python 实现二维扩散过程的可视化预测或拟合（图 14）。

关键词：菲克定律 多项式回归拟合 偏微分方程 迭代法 三维可视化

一、问题重述

1.1 问题背景

Fick 定律对物质扩散现象的宏观规律进行了描述，Fick 第一定律建立了扩散通量 J 与扩散系数 D 和浓度梯度 C 之间的关系：

$$J = -D\nabla C,$$

Fick 第二定律指出了非稳态扩散过程中浓度随时间的变化率与扩散通量随距离变化率之间的关系：

$$\frac{\partial C}{\partial t} = -\nabla \cdot J$$

Fick 定律同样适用于高维与多组元体系。在给定温度下，扩散系数通常是关于组元的函数，研究热点是根据可靠组元获取扩散系数。实验测量与计算过程属于反问题中的参数识别问题，过程不适定、非线性、计算量大。

1.2 问题提出

问题一：已知初始条件，不同测量时刻组元的扩散通量 J 和浓度 C 的空间分布时（见附件），建立扩散系数 D 识别问题的数学模型，求解并绘制随浓度 C 变化的扩散系数 D 的曲线

问题二：结合实际，只测量组元 J 或 C 之一，利用尽可能少的实验数据而得到尽可能好的测量结果。已知初始条件，不同时刻

- (1) 组元 J 未知， C 空间分布已知；
- (2) 组元 J 已知， C 空间分布未知。

基于问题 1，分别建立 D 识别问题的数学模型，结合模型的特征设计稳定可靠的求解算法。利用附件提供数据求解并绘制随浓度 C 变化的扩散系数 D 的曲线，比较分析结果与问题 1 的异同。

问题三：结合问题 1、2，具体分析实验测量数据的误差水平、数据量和迭代初值的选取方式等因素，总结影响 D 识别精度的主要因素。

问题四：实际问题中大多为高维和多组元等复杂情形，对上述模型和算法如何进行应用和推广并择一具体情形加以说明。

二、问题分析

2.1 问题一分析

问题一要求我们建立识别参数 D 的数学模型、求解并绘制 D - C 曲线。根据问题描述，我们已知不同测量时刻组元的 J 和 C 的空间分布，首先，考虑一维情形下的扩散问题，直接利用 Fick 第一定律建立函数关系 $J = -D\nabla C$ ，我们用 excel 作出 C - x 曲线进行六次多项式拟合，并用 python 求导计算得到不同时刻各位置的浓度梯度，由 Fick 第一定律得到不同时刻各位置的 D ，作出 D - C 曲线。因附件中数据较多，间隔较小，保证拟合可信度。

2.2 问题二分析

问题二结合实际情况考虑，限制了已知实验数据量，只测量组元 J 或 C 两

者之一，即只利用附件一或二进行求解。

(1) J 未知，C 空间分布已知时，结合 Fick 第一定律与非稳态扩散 Fick 第二定律建立函数关系 $\frac{\partial C}{\partial t} = D \left(\frac{\partial^2 C}{\partial x^2} \right)$ 。基于问题一不同时刻的浓度梯度多项式，用 python 求导计算得到不同时刻各位置浓度对距离的二次偏导。基于问题一，扩散系数 D 随 C 变化始终趋于常数，为保证得到不同浓度下 D 的取值，同时简化计算，我们以 $\Delta x = 0.05$ 等间距取数据点作出不同位置 C-t 曲线进行多项式拟合，并用 python 求导计算得到不同位置的浓度随时间变化率，由函数关系得到所取数据点对应 D 并作出 D-C 曲线。

(2) 在问题 2 中，我们已知组元的扩散通量 J，但浓度 C 的空间分布未知。因此，我们需要重新建立识别问题的数学模型。根据 Fick 第一定律，我们可以重新整理得到 $\frac{\partial C}{\partial x} = -\frac{J}{D}$ ，根据已知数据点构成 $(\frac{\delta C}{\delta x_i}, C_i)$ 对，利用最小二乘法拟合扩散系数 D 的数学模型。建立目标优化函数并最小化该函数，得到最佳的 D，绘制 D-C 曲线。

将问题 2 与问题 1 中绘制的结果进行比较分析异同。

2.3 问题三分析

根据题目要求，分析出影响扩散系数估计精确性的主要影响因素，应当从问题一问题二得到的扩散系数 D 随浓度 C 的变化入手，观察散点图的分布，找出偏离整体较远的数据点，主要是分布在空间上的两端的数据点和在扩散刚开始的时候收集的数据点，通过查找资料分析这些偏差造成的原因，是由溶液浓度边界、浓度梯度过大导致的，因此实验数据测量位置和实验测量时间以及实验测量本身存在的误差都是影响扩散系数估计精度的重要因素。通过观察问题二中的 D-C 散点图，以及根据科学实验方法论，收集的稳定数据量越大，得到的数据点越多，扩散系数 D 随浓度 C 变化趋势越准确，扩散系数估计精度越高，所以得出结论数据量也是影响扩散系数估计精度的重要因素。

2.4 问题四分析

问题四要求我们考虑实际问题中高维与多组元更复杂的情况进行应用推广，我们根据实际中的二维扩散中的 Fick 定律，利用 python 建立二维扩散模型。我们采取循 $nt+1$ 次迭代方法更新浓度值。在每次迭代时通过离散化的扩散方程计算新的浓度分布 C。对于离散方程与微分方程之间的误差，利用耗散效应把尖锐的解抹平，使结果光滑。最终利用编程实现对二维扩散随时间的动态模拟。

三、基本假设

- 1、假设所有涉及的扩散模型中都满足 Fick 定律适用的宏观条件。
- 2、假设仅考虑给定温度下扩散对物质运输的影响。
- 3、假设仅考虑给定温度下扩散对物质运输的影响假设所有涉及的扩散模型中对扩散系数 D 的参数识别自变量仅为组元扩散通量 J 与扩散浓度 C。

四、符号说明

符号	定义	单位
J	扩散通量	原子数/(m ² ·s)
C	扩散物质的体积浓度	原子数/m ³ 或 kg/m ³
D	扩散系数	m ² /s
∇	梯度算子	/
X	位置参数	m
dt	时间步长	s
nt	时间步的数量	/
Δ	前差分算子	/

五、扩散模型建立与扩散系数参数识别

5.1 问题一：多项式回归拟合模型

首先，考虑一维情形下的扩散问题，直接由题干中的 Fick 第一定律已知，扩散通量 J 与该截面处的浓度梯度 C 成正比，函数关系为：

$$J = -D\nabla C \quad (1)$$

对于等式右边，为得到不同时刻各位置处对应的浓度梯度，我们根据附件二中的数据，利用 excel 作出 C-x 散点图，同时采用多项式回归拟合得到不同时刻下 C 与 x 的函数关系，如图 1，图中的 R 平方值是趋势线拟合程度的指标，它的数值大小可以反映趋势线的估计值与对应的实际数据之间的拟合程度，R 的平方越接近 1，表明拟合程度越高，趋势线的可靠性就越高，在 t=0.2、t=0.4、t=0.6、t=0.8 和 t=1 时刻的 C-x 曲线 R 的平方都等于 1 或非常逼近 1，故拟合得到多项式具有可靠性。

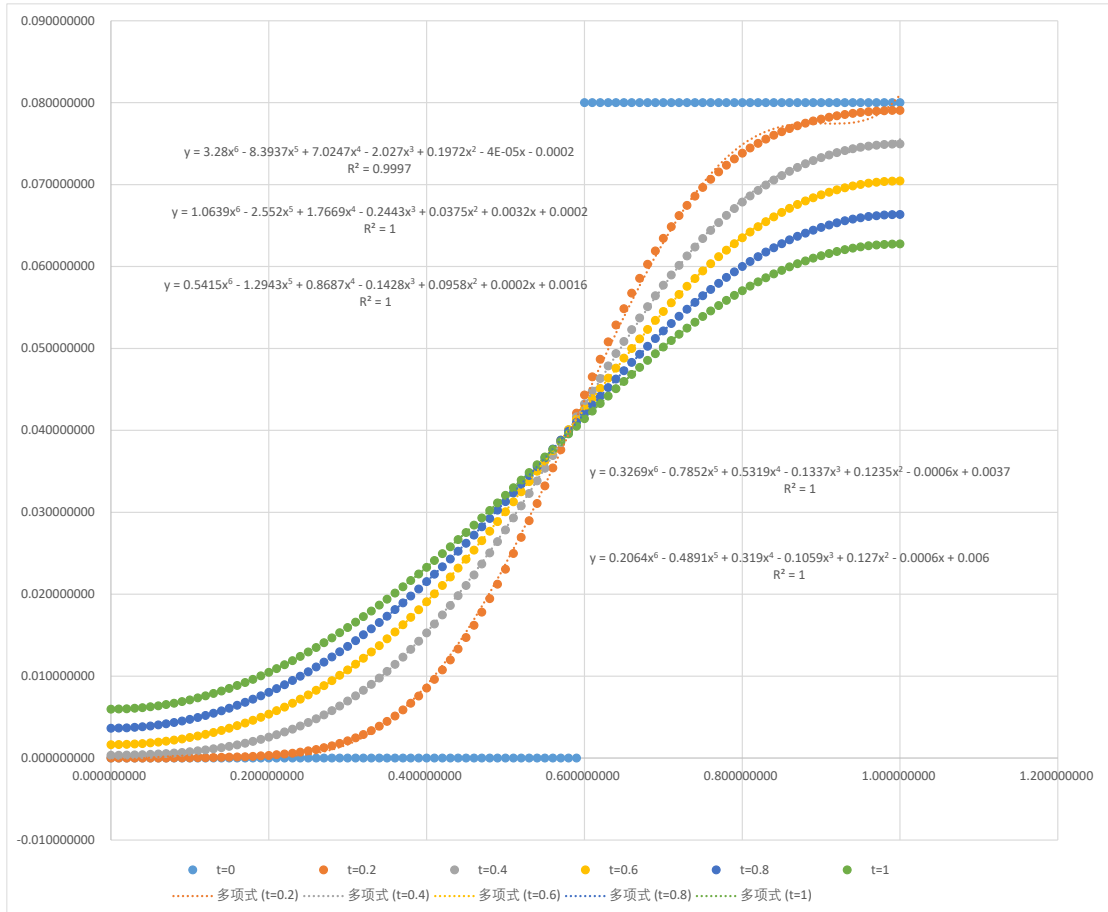


图 1 不同时刻下 C 与 x 的多项式拟合

对所拟合多项式求导得到不同时刻各位置的浓度梯度的多项式拟合：

表 1 不同时刻下浓度梯度的多项式拟合

t	$\frac{\partial C}{\partial x}$
0.2s	$3.28 \times 6x^5 - 8.39 \times 5x^4 + 7.02 \times 4x^3 - 2.03 \times 3x^2 + 0.20 \times 2x - 0.00004$
0.4s	$1.06 \times 6x^5 - 2.55 \times 5x^4 + 1.77 \times 4x^3 - 0.24 \times 3x^2 + 0.04 \times 2x + 0.0032$
0.6s	$0.54 \times 6x^5 - 1.29 \times 5x^4 + 0.87 \times 4x^3 - 0.14 \times 3x^2 + 0.10 \times 2x + 0.0002$
0.8s	$0.33 \times 6x^5 - 0.79 \times 5x^4 + 0.53 \times 4x^3 - 0.13 \times 3x^2 + 0.12 \times 2x - 0.0006$
1.0s	$0.21 \times 6x^5 - 0.49 \times 5x^4 + 0.32 \times 4x^3 - 0.11 \times 3x^2 + 0.13 \times 2x - 0.0006$

使用 python 计算得到不同时刻各位置的浓度梯度具体数值，同时利用附件一中的不同时刻各位置的扩散通量 J，由 Fick 第一定律可求得不同时刻各位置的 D，在 excel 可用此数据作出 D-C 曲线，部分数据与曲线图：

表 2 部分扩散系数 D 与对应的浓度 C 值

t=0.2	C	t=0.4	C	t=0.6	C	t=0.8	C	t=1	C
5.2627E-02	1.4720E-02	5.3590E-02	2.1077E-02	5.3590E-02	2.4273E-02	5.3613E-02	2.6229E-02	5.3585E-02	2.7539E-02
5.3329E-02	1.6218E-02	5.3707E-02	2.2360E-02	5.3603E-02	2.5388E-02	5.3598E-02	2.7218E-02	5.3562E-02	2.8425E-02
5.3993E-02	1.7802E-02	5.3819E-02	2.3681E-02	5.3617E-02	2.6526E-02	5.3585E-02	2.8223E-02	5.3539E-02	2.9322E-02
5.4615E-02	1.9472E-02	5.3920E-02	2.5038E-02	5.3631E-02	2.7685E-02	5.3574E-02	2.9240E-02	5.3518E-02	3.0227E-02

5.5186E-02	2.1226E-02	5.4018E-02	2.6430E-02	5.3642E-02	2.8864E-02	5.3564E-02	3.0271E-02	5.3498E-02	3.1141E-02
5.5697E-02	2.3058E-02	5.4101E-02	2.7855E-02	5.3653E-02	3.0061E-02	5.3552E-02	3.1312E-02	5.3478E-02	3.2063E-02
5.6152E-02	2.4967E-02	5.4178E-02	2.9311E-02	5.3663E-02	3.1275E-02	5.3542E-02	3.2364E-02	5.3460E-02	3.2990E-02
5.6536E-02	2.6945E-02	5.4243E-02	3.0793E-02	5.3671E-02	3.2504E-02	5.3530E-02	3.3424E-02	5.3444E-02	3.3923E-02
5.6851E-02	2.8987E-02	5.4294E-02	3.2301E-02	5.3675E-02	3.3746E-02	5.3521E-02	3.4492E-02	5.3429E-02	3.4860E-02
5.7090E-02	3.1086E-02	5.4337E-02	3.3829E-02	5.3681E-02	3.4999E-02	5.3516E-02	3.5566E-02	5.3416E-02	3.5800E-02

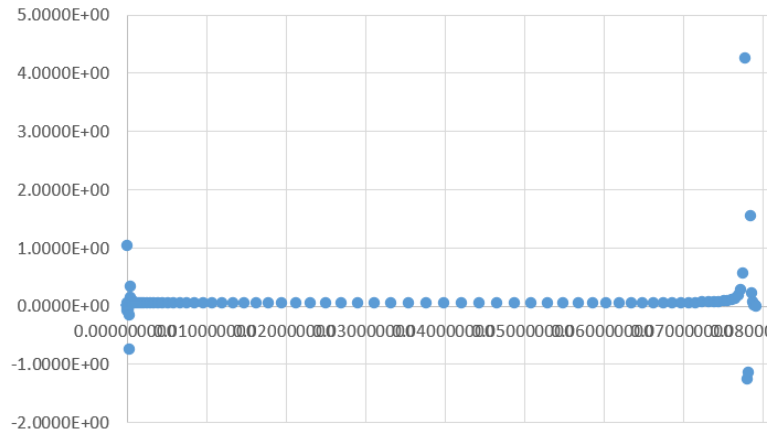


图 2 $t=0.2s$ 时 D-C 曲线

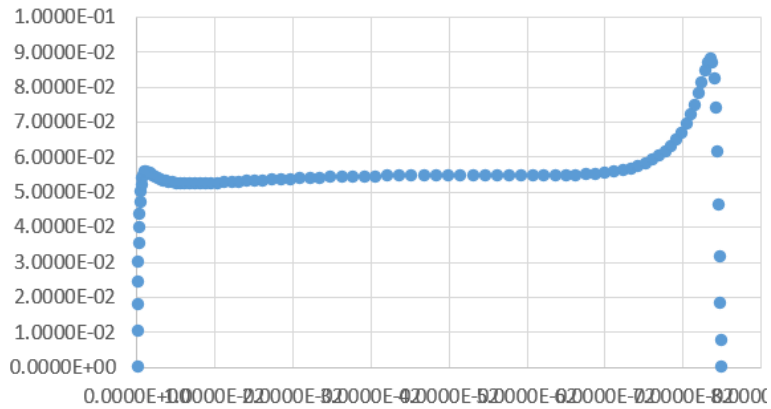


图 3 $t=0.4s$ 时 D-C 曲线

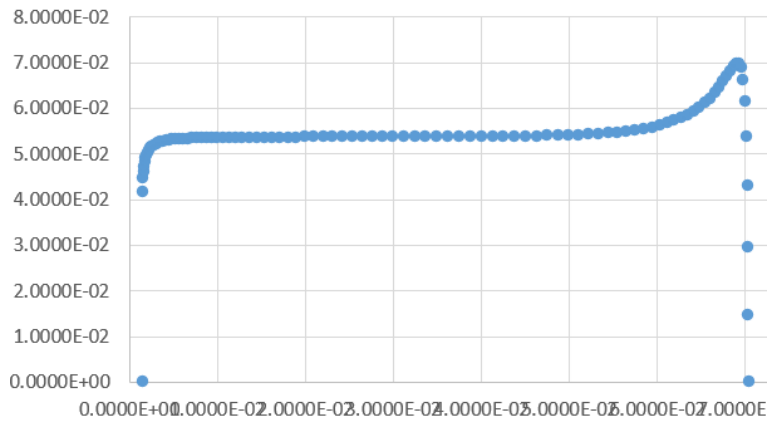


图 4 $t=0.6s$ 时 D-C 曲线

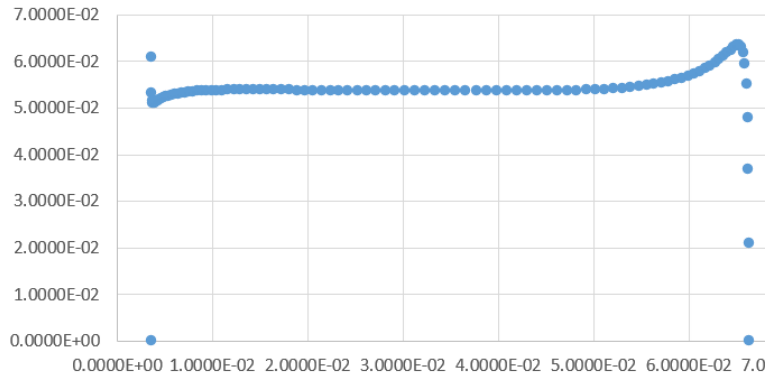


图 5 t=0.8s 时 D-C 曲线

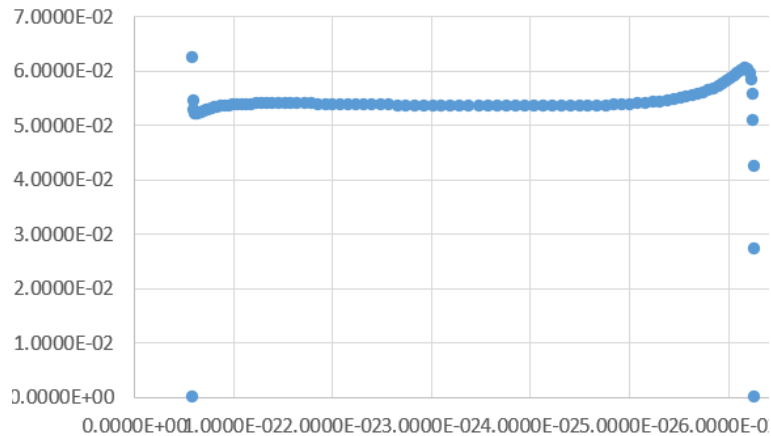


图 6 t=1.0s 时 D-C 曲线

最终得到 D-C 曲线除两端个别点偏离，整体上看扩散系数 D 基本不随浓度 C 变化而变化，稳定在某个常数附近，符合实际情况下扩散系数 D 只受温度和压强影响，贴合模型，对于扩散系数 D 的识别精度较高。

5.2 问题二：有限条件下的多项式拟合与目标优化函数

5.2.1 第一小问：

在仅已知浓度 C 的空间分布的情况下，由 Fick 第二定律，在指定位置处，浓度随时间 t 的变化率等于该处的扩散通量随距离变化率的负值，即

$$\frac{\partial C}{\partial t} = -\nabla \cdot J. \quad (2)$$

与 Fick 第一定律联立，在扩散系数 D 为常数时，得

$$\frac{\partial C}{\partial t} = D \left(\frac{\partial^2 C}{\partial x^2} \right) \quad (3)$$

故要得到某一浓度 C 条件下扩散系数 D 的值，要先得到浓度 C 关于时间的倒数和浓度 C 对空间位置的二阶偏导。考虑到在不同位置浓度 C 关于时间的曲线数量太多，计算时间导数计算量太大，我们计划减少位置上的采样。通过观察问题一中得到的 D-C 散点分布，我们发现扩散系数 D 的值相对稳定，基本不随浓度 C 的变化而变化，所以对数据进行抽样是可行的。又因为题目需要得到有限条件下的 D-C 关系，则需要 C 的抽样尽可能均匀，在浓度扩散过程中，通过观察

附件二中的数据和生活常识可知，浓度随 x 的增大而增大，故距离上每间隔 0.05m 抽样 21 组数据，能够使 D-C 散点分布尽可能均匀。

抽取的数据和拟合的 C-t 多项式如下：

表 3 从附件二中抽取的浓度数据

X	T=0	t=0.2	t=0.4	t=0.6	t=0.8	t=1
0.0000E+00	0.0000E+00	4.3597E-06	3.4771E-04	1.6392E-03	3.6533E-03	5.9782E-03
5.0000E-02	0.0000E+00	9.4420E-06	4.4781E-04	1.8514E-03	3.9215E-03	6.2610E-03
1.0000E-01	0.0000E+00	3.3259E-05	7.7944E-04	2.5063E-03	4.7310E-03	7.1073E-03
1.5000E-01	0.0000E+00	1.1026E-04	1.4351E-03	3.6544E-03	6.0944E-03	8.5115E-03
2.0000E-01	0.0000E+00	3.2884E-04	2.5614E-03	5.3663E-03	8.0275E-03	1.0463E-02
2.5000E-01	0.0000E+00	8.7876E-04	4.3411E-03	7.7165E-03	1.0543E-02	1.2945E-02
3.0000E-01	0.0000E+00	2.0999E-03	6.9622E-03	1.0766E-02	1.3643E-02	1.5935E-02
3.5000E-01	0.0000E+00	4.4817E-03	1.0580E-02	1.4551E-02	1.7317E-02	1.9398E-02
4.0000E-01	0.0000E+00	8.5576E-03	1.5284E-02	1.9068E-02	2.1532E-02	2.3288E-02
4.5000E-01	0.0000E+00	1.4720E-02	2.1077E-02	2.4273E-02	2.6229E-02	2.7539E-02
5.0000E-01	0.0000E+00	2.3058E-02	2.7855E-02	3.0061E-02	3.1312E-02	3.2063E-02
5.5000E-01	0.0000E+00	3.3234E-02	3.5375E-02	3.6261E-02	3.6644E-02	3.6741E-02
6.0000E-01	8.0000E-02	4.4323E-02	4.3227E-02	4.2617E-02	4.2041E-02	4.1430E-02
6.5000E-01	8.0000E-02	5.4845E-02	5.0860E-02	4.8811E-02	4.7283E-02	4.5964E-02
7.0000E-01	8.0000E-02	6.3442E-02	5.7716E-02	5.4516E-02	5.2144E-02	5.0173E-02
7.5000E-01	8.0000E-02	6.9664E-02	6.3422E-02	5.9465E-02	5.6428E-02	5.3908E-02
8.0000E-01	8.0000E-02	7.3824E-02	6.7863E-02	6.3506E-02	5.9999E-02	5.7053E-02
8.5000E-01	8.0000E-02	7.6451E-02	7.1109E-02	6.6594E-02	6.2787E-02	5.9537E-02
9.0000E-01	8.0000E-02	7.8003E-02	7.3296E-02	6.8753E-02	6.4770E-02	6.1322E-02
9.5000E-01	8.0000E-02	7.8807E-02	7.4553E-02	7.0023E-02	6.5953E-02	6.2395E-02
1.0000E+00	8.0000E-02	7.9054E-02	7.4962E-02	7.0442E-02	6.6345E-02	6.2752E-02

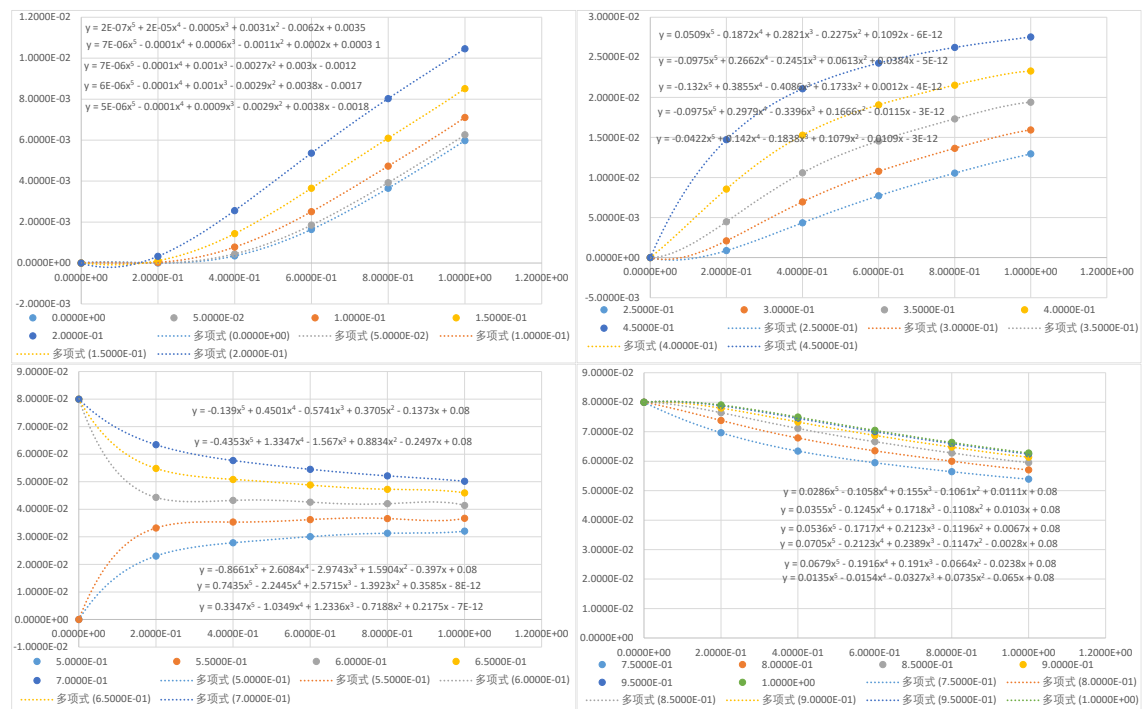


图 7 抽样数据拟合的 C-t 曲线

用 Python 对 21 组数据拟合的曲线在 t=0.2、t=0.4、t=0.6、t=0.8 和 t=1 处求导，得到浓度对时间的导数。

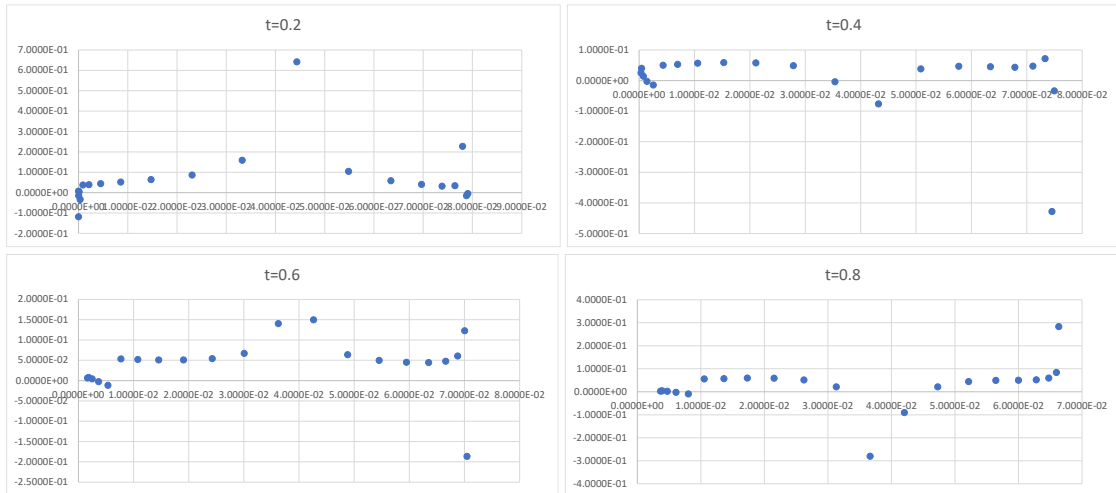
表 4 抽样数据浓度 C 对时间的导数

x	t=0.2	t=0.4	t=0.6	t=0.8	t=1
0.0000E+00	2.7448E-03	1.8870E-03	1.2088E-03	6.9344E-04	3.2500E-04
5.0000E-02	2.7568E-03	1.9352E-03	1.3175E-03	8.8749E-04	6.3000E-04
1.0000E-01	2.0369E-03	1.2953E-03	7.5814E-04	4.0954E-04	2.3500E-04
1.5000E-01	-1.7114E-04	-4.1670E-04	-5.5386E-04	-5.9846E-04	-5.6500E-04
2.0000E-01	-5.0194E-03	-3.9549E-03	-3.0026E-03	-2.1586E-03	-1.4190E-03
2.5000E-01	1.4410E-02	1.8146E-02	1.5418E-02	1.3234E-02	1.0500E-02
3.0000E-01	2.3141E-02	2.2554E-02	1.5858E-02	1.3447E-02	7.0000E-03
3.5000E-01	3.2768E-02	2.5504E-02	1.5408E-02	1.3136E-02	4.0000E-03
4.0000E-01	4.1246E-02	2.5459E-02	1.4069E-02	1.1386E-02	3.0000E-03
4.5000E-01	4.6469E-02	2.1200E-02	1.2110E-02	7.6896E-03	6.2000E-03
5.0000E-01	4.7573E-02	1.2495E-02	9.9600E-03	1.9224E-03	1.4600E-02
5.5000E-01	4.4284E-02	-4.4400E-04	7.5000E-03	-5.9480E-03	2.7900E-02
6.0000E-01	-4.1216E-02	4.5456E-03	-8.3392E-03	5.2144E-03	-3.6000E-02
6.5000E-01	-4.5152E-02	-9.1752E-03	-1.0874E-02	-2.9288E-03	-2.1600E-02
7.0000E-01	-4.4701E-02	-1.9034E-02	-1.3914E-02	-9.6392E-03	-1.3200E-02
7.5000E-01	-3.9909E-02	-2.4110E-02	-1.6674E-02	-1.4075E-02	-1.0200E-02
8.0000E-01	-3.3028E-02	-2.5598E-02	-1.8743E-02	-1.6658E-02	-1.0500E-02
8.5000E-01	-2.6242E-02	-2.5213E-02	-2.0171E-02	-1.8038E-02	-1.2200E-02
9.0000E-01	-2.0730E-02	-2.4170E-02	-2.1152E-02	-1.8913E-02	-1.4400E-02
9.5000E-01	-1.7104E-02	-2.3204E-02	-2.1680E-02	-1.9396E-02	-1.6400E-02
1.0000E+00	-1.6017E-02	-2.3044E-02	-2.2058E-02	-1.9646E-02	-1.6300E-02

再将问题一中多项式拟合的 C-X 曲线对 x 求两阶偏导，利用

$$D = \frac{\frac{\partial C}{\partial t}}{\frac{\partial^2 C}{\partial x^2}} \quad (4)$$

计算出不同位置不同时刻扩散系数的值，绘制 D-C 散点图



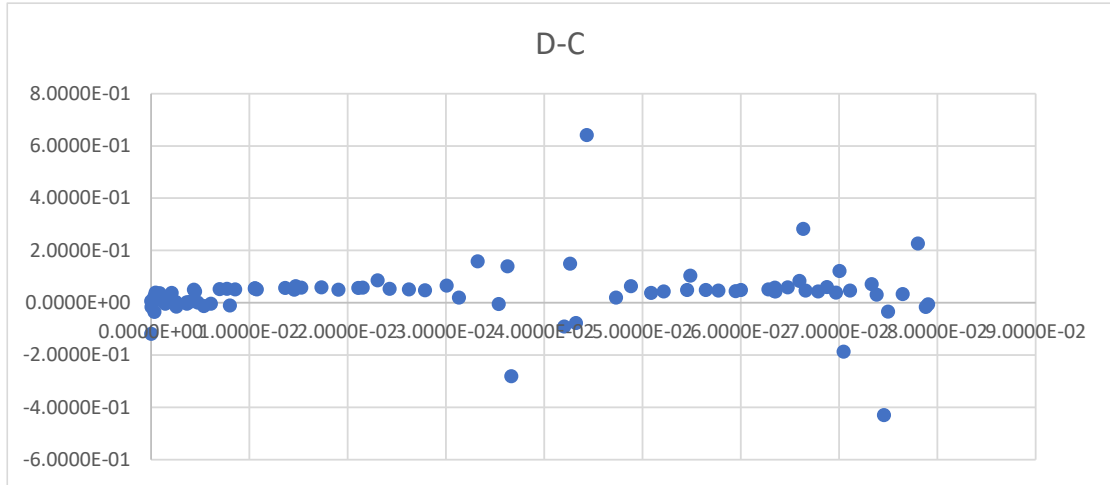


图 8 仅由浓度 C 空间分布得到的 D-C 分布

5.2.2 第二小问：

在问题 2 中，我们已知组元的扩散通量 J ，但浓度 C 的空间分布未知。因此，我们需要重新建立识别问题的数学模型。根据 Fick 第一定律，我们重新整理方程(1)得到：

$$\frac{\partial C}{\partial x} = -\frac{J}{D} \quad (5)$$

在这种情况下，我们只通过实验测量得到扩散通量 J 的数据，而无法直接测量浓度梯度，因此我们需要使用扩散通量 J 来逆推浓度梯度，并进一步解决扩散系数识别问题。

对于一组已知的，扩散通量位置数据点 (J_i, x_i) ，我们可以利用公式(5)来估计浓度梯度，进而得到浓度 C 的变化情况。然后我们可按照问题一中的方法，利用拟合方法基于已估计的浓度梯度与扩散通量求解扩散系数 D 。

根据已知数据点构成 $(\frac{\partial C}{\partial x_i}, C_i)$ 对，利用最小二乘法，将 $(\frac{\partial C}{\partial x_i}, C_i)$ 视为已知数据点以拟合扩散系数 D 的数学模型，建立优化目标函数并最小化该函数，得到最佳的扩散系数 D 。

表 5 不同位置处 C 对应 D 值部分数据

C	D(0.55)	D(0.56)	D(0.57)	D(0.58)	D(0.59)	D(0.60)
0.01	-1.1588e-03	-1.1746e-03	-1.1850e-03	-1.1898e-03	-1.1889e-03	-1.1824e-03
0.02	-5.7940e-04	-5.8730e-04	-5.9250e-04	-5.9569e-04	-5.9727e-04	-5.9762e-04
0.03	-3.8633e-04	-3.9094e-04	-3.9333e-04	-3.9468e-04	-3.9537e-04	-3.9550e-04
0.04	-2.8974e-04	-2.9263e-04	-2.9419e-04	-2.9510e-04	-2.9550e-04	-2.9555e-04
0.05	-2.3189e-04	-2.3370e-04	-2.3474e-04	-2.3546e-04	-2.3590e-04	-2.3611e-04

0.06	-1.9323e-04	-1.9436e-04	-1.9500e-04	-1.9541e-04	-1.9572e-04	-1.9594e-04
0.07	-1.6560e-04	-1.6637e-04	-1.6684e-04	-1.6718e-04	-1.6745e-04	-1.6769e-04
0.08	-1.4462e-04	-1.4514e-04	-1.4552e-04	-1.4580e-04	-1.4602e-04	-1.4619e-04

将问题二的两种情形计算结果与问题一对比，我们得到问题二的第一种情形，即当组元的扩散通量 J 未知，浓度 C 的空间分布已知时， $D-C$ 曲线整体呈与 x 轴平行的直线，扩散随时间趋于稳定，扩散系数 D 随 C 的分布趋于常值，与问题一中趋势相符。除去两端边界上的特殊偏离点后，对数据求平均可得 D 在 5.3481×10^{-2} 附近波动，与问题一中 D 的估计值 5.3536×10^{-2} 接近。而对于问题二的第二种情形，即当组元的扩散通量 J 已知，浓度 C 的空间分布未知时，其趋势与数值都与问题一中相差较大，相似度更低。

5.3 问题三：影响扩散系数识别精度的因素

从图 2 至图 6 发现，大部分点都波动在某一常数附近，但存在少数点在偏离该常数较远的位置，如图所示：

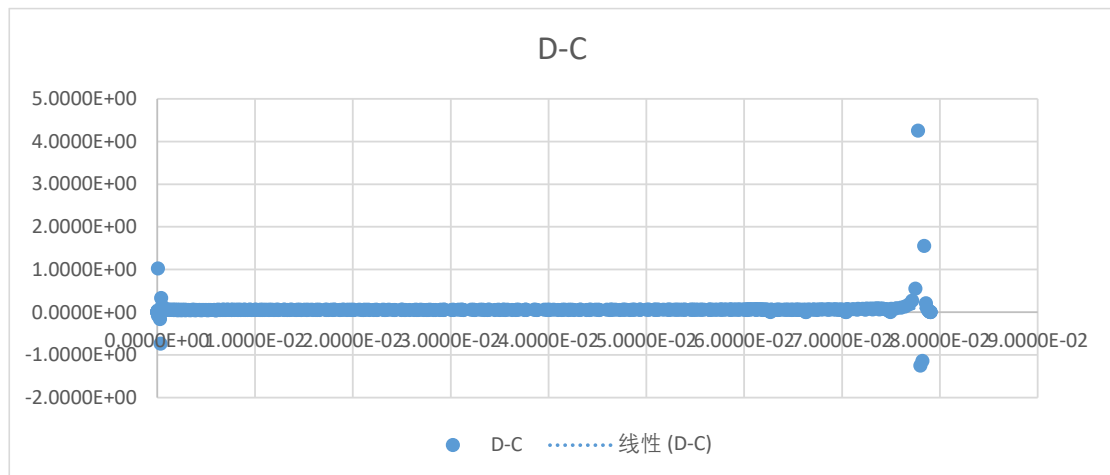


图 9 题目一中未预处理的数据绘制的 $D-C$ 散点图

判断这些少部分的点偏离大部分点是因为实验原始测量数据存在误差。去掉大于 0.1 的数据和小于 0 的数据，得到测量误差更小的 $D-C$ 分布，如下图所示。

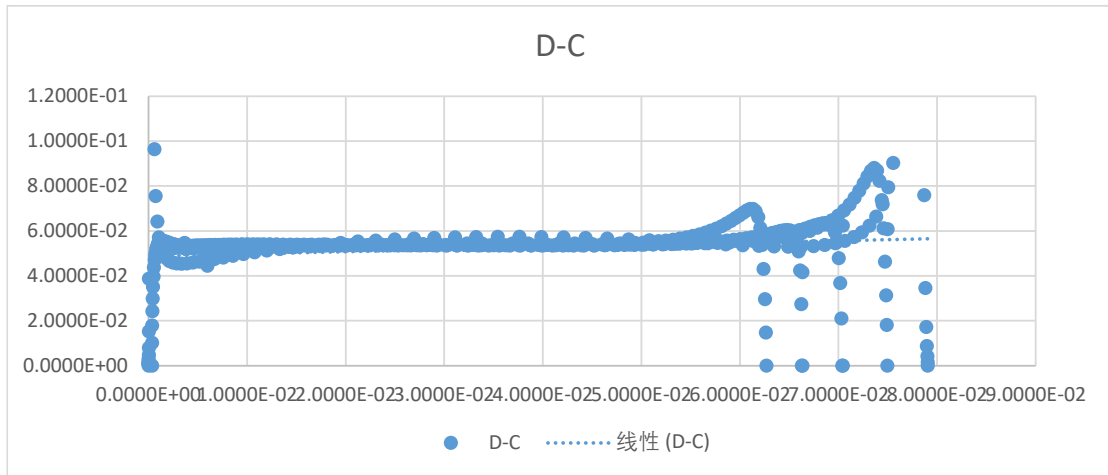


图 10 题目一中预处理的数据绘制的 D-C 散点图

通过观察经过预处理数据得到的 D-C 散点图，浓度在 $[0.02, 0.05]$ 的范围内，数据点基本稳定在一条直线附近，但是在浓度 C 最小与最大的位置上，数据点偏离直线较大，搜集资料得知，气体或液体由于分子运动，其流体内部的浓度是相对均匀的。但当流体与固体接触时，根据边界层理论，它的分子向界面扩散受到阻力，导致界面上物质的浓度和流体内部的浓度有较大差异。因此可以判断，附件一、附件二中的实验数据是从长度仅为 0.1m 左右的盛有溶液的容器中采集得到的，才会导致 D-C 分布两端数据点偏离直线较远。所以，若要得到 D 在该特定环境中的值，应当在浓度为 $[0.02, 0.05]$ 中取值。

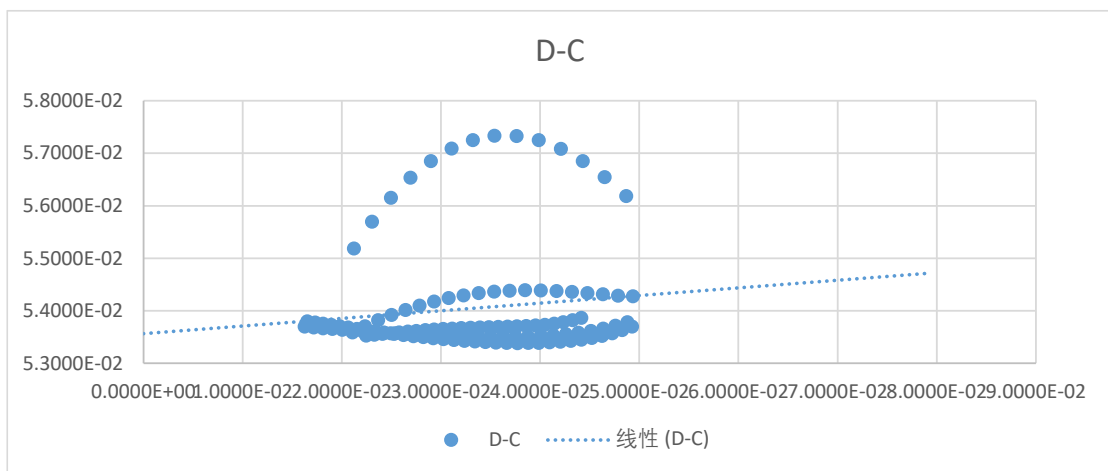


图 11 浓度为 $[0.02, 0.05]$ 范围内的 D-C 散点图

发现 $t=0.2$ 时刻测量的数据得到的扩散系数值 D 偏离其他时刻扩散系数值 D 较远，分析是刚开始扩散时，浓度梯度大，扩散速度很快，测量时刻溶液浓度也在变化，导致了测量的溶液浓度有误差。去掉 $t=0.2$ 时刻和 $t=0.4$ 时刻的数据，D-C 散点分布更加准确，如下图所示：

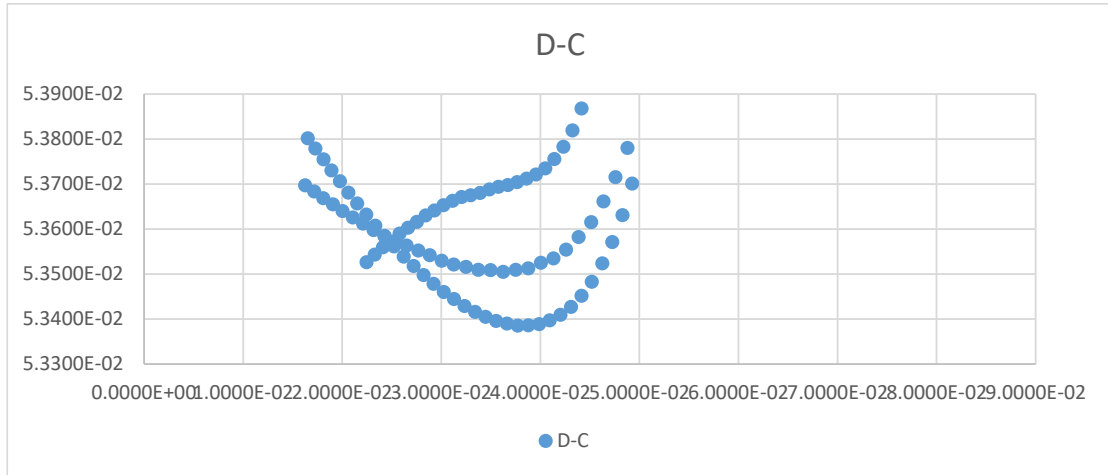


图 12 $t=0.6$ 、 $t=0.8$ 、 $t=1$ 时刻绘制的 D-C 散点图

从上图纵坐标看出，扩散系数基本在 $[0.0534, 0.0538]$ 小范围内变化，可以取其平均值为该环境条件下的扩散系数 D 值。

观察问题二第一小问中图 8，除了存在测量误差的点以外，还发现数据点集中分布在两侧，中间的数据点少且分散，D-C 分布不确定性很大，这是由数据采集数据量少导致的。若处理更多的数据，D-C 散点分布将更加准确。

综上所述，实验测量数据的准确性、实验测量数据的位置、实验测量数据的时间、测量数据量都对扩散系数识别精度产生影响。

5.4 问题四：考虑空气阻力的斜抛模型的建立与求解

由 Fick 一、二定律，在二维情形下扩散方程为

$$\frac{\partial C}{\partial t} = D \left(\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2} \right) \quad (6)$$

对其中的扩散项采用中心差分：

$$\frac{C_{i,j}^{n+1} - C_{i,j}^n}{\Delta t} = D \left(\frac{C_{i+1,j}^n - 2C_{i,j}^n + C_{i-1,j}^n}{\Delta x^2} + \frac{C_{i,j+1}^n - 2C_{i,j}^n + C_{i,j-1}^n}{\Delta y^2} \right) \quad (7)$$

整理得到：

$$C_{i,j}^{n+1} = C_{i,j}^n + \frac{v\Delta t}{\Delta x^2} (C_{i+1,j}^n - 2C_{i,j}^n + C_{i-1,j}^n) + \frac{v\Delta t}{\Delta y^2} (C_{i,j+1}^n - 2C_{i,j}^n + C_{i,j-1}^n) \quad (8)$$

这是离散化的二维扩散方程，描述二维空间中某点浓度时间变化情况，随其中 $C_{i,j}^n$ 表示 (i, j) 处第 n 步时间的浓度， Δt 表示时间步长， $C_{i+1,j}^n$ 表示在相邻位置处第 n 步时间的浓度， Δx 、 Δy 表示空间网格的大小。

用 python 实现二维扩散模型的建立，分别定义各参数，创建坐标系。为贴合实际，我们根据附件中的数据，在中间一个矩形区域设置初始浓度为 0.08。

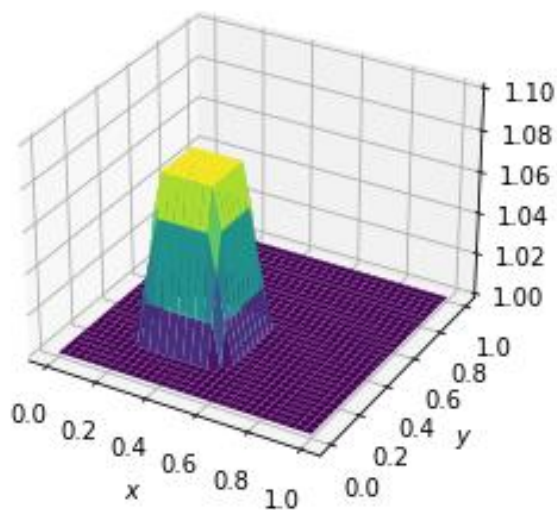


图 13 二维扩散初始状态

我们定义了一个名为 `diffuse` 的函数，其输入参数为 `nt` 表示需要求解的时间步数。在函数内部，重新设置浓度 `C` 的值，并进行循环 `nt+1` 次迭代来更新浓度值。在每次迭代时，使用当前时间步 `n` 上的浓度分布 `Cn`，通过离散化的扩散方程计算新的浓度分布 `C`。边界条件被设定为固定的浓度值（全 1）。

每次迭代后，都会创建一个新的 3D 图形并绘制更新后的浓度分布 `C`。最后，调用 `diffuse` 函数并传入任意时间参数，如 50，以执行 50 个时间步的求解和可视化过程，其模拟二维扩散动态变化过程如下（图示间隔 4 个时间步）：

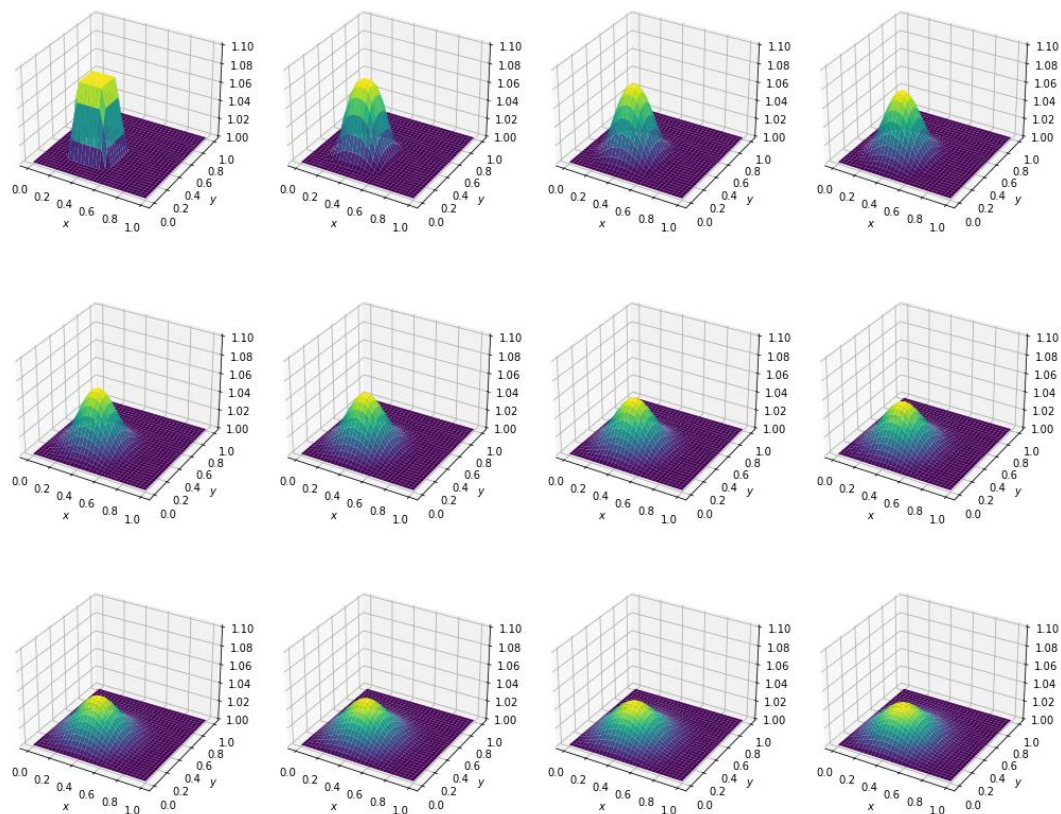


图 14 二维扩散模型动态模拟

对于离散方程与微分方程之间的误差，利用耗散效应把尖锐的解抹平，使结果光滑，在不可避免存在误差的情况下提高解的光滑性，避免奇点。结果以三维可视化展示了浓度随时间变化的动态过程。

为应用于实际问题，推广至高维与多组元的情况，可继续再此模型基础上加入更高维参数，或是引入新的组元与浓度的函数关系，使模型适用于不同情况下的实际问题。此模型优势在于可根据需要预设扩散系数及其他参数，实现二维扩散过程的可视化预测或与实验测量所得数据对比拟合，在高维与多变量情况下使结果更加直观。

六、参考文献

- [1]童强. 基于多项式回归模型的物质浓度测定方法[J]. 中国建材科技, 2019, 第28卷(5): 29, 38
- [2]丁锐, 桂泰江, 蒋建明, 余海斌. 应用拉普拉斯变换和留数法求解常见非稳态扩散情况下的菲克定律[J]. 数学的实践与认识, 2017, 第47卷(1): 271-279
- [3]史明英. 非线性对流扩散问题的全隐迎风有限差分格式与迭代方法[D]. 东北师范大学, 2021

附录

求浓度梯度、二次导函数

```
def calculate_y(x):  
    # 在这里定义函数表达式，对于不同的 x 返回不同的 y 值  
    y = 1.0639*6*x**5-2.552*5*x**4+1.7669*4*x**3-  
    0.2443*3*x**2+0.0375*2*x+0.0032 # 这是一个二次函数的示例  
  
    return y
```

定义起始点、结束点和步长

```
start_x = 0.00  
end_x = 1.01  
step = 0.01
```

使用循环遍历等间隔的不同 x 值，并计算相应的 y 值

```
current_x = start_x  
while current_x <= end_x:  
    current_y = calculate_y(current_x)  
    print(f"当 x = {current_x:.2e}, y = {current_y:.4e}")  
    current_x += 0.01
```

二维可视化扩散模型函数

```
import numpy as np  
from matplotlib import cm  
import matplotlib.pyplot as plt  
from mpl_toolkits.mplot3d import Axes3D  
nx = 31
```

```

ny = 31
nt = 17
nD = 0.05
dx = 2 / (nx - 1)
dy = 2 / (ny - 1)
sigma = 0.25
dt = sigma * dx * dy / nD

x = np.linspace(0, 1, nx)
y = np.linspace(0, 1, ny)

C = np.ones((ny, nx)) # create a 1xn vector of 1's
Cn = np.ones((ny, nx)) #非常常见的手法了
C[int(.5 / dy):int(1 / dy + 1), int(.5 / dx):int(1 / dx + 1)] = 1.08
fig = plt.figure()
ax = fig.gca(projection='3d')
X, Y = np.meshgrid(x, y)
surf = ax.plot_surface(X, Y, C, rstride=1, cstride=1, cmap=cm.viridis,
                      linewidth=0, antialiased=False)

ax.set_xlim(0, 2)
ax.set_ylim(0, 2)
ax.set_zlim(1, 1.1)

ax.set_xlabel('$x$')
ax.set_ylabel('$y$');
def diffuse(nt):
    C[int(.5 / dy):int(1 / dy + 1), int(.5 / dx):int(1 / dx + 1)] = 1.08
    #初场定义

    for n in range(nt + 1):
        Cn = C.copy()
        C[1:-1, 1:-1] = (Cn[1:-1, 1:-1] +
                       nD * dt / dx**2 *
                       (Cn[1:-1, 2:] - 2 * Cn[1:-1, 1:-1] + Cn[1:-1, 0:-2])) +
                       nD * dt / dy**2 *
                       (Cn[2:, 1:-1] - 2 * Cn[1:-1, 1:-1] + Cn[0:-2, 1:-1]))
        #更新场
        C[0, :] = 1
        C[-1, :] = 1
        C[:, 0] = 1
        C[:, -1] = 1
        #边界条件
        fig = plt.figure()

```

```
ax = fig.gca(projection='3d')
surf = ax.plot_surface(X, Y, C[:, :], rstride=1, cstride=1, cmap=cm.viridis,
linewidth=0, antialiased=True)
ax.set_zlim(1, 1.1)
ax.set_xlabel(' $x$ ')
ax.set_ylabel(' $y$ ');#计算完成后绘制 u 的分布
```

```
diffuse(50)
```