基于最小二乘法的污染源测定模型

摘　要

本文通过最小二乘法、循环迭代法，利用指数型函数建立了污染源扩散模型，使用 MATLAB 的 solve 函数建立了单污染源测定模型，通过运用 MATLAB 的 fsolve 函数建立了双污染源测定模型。

针对问题一：首先根据题意建立浓度与扩散距离的指数函数关系，得知初始浓度为 100.0，将指数函数化简为线性关系，利用最小二乘法，求解函数系数为-0.02。

针对问题二：基于污染源扩散模型，建立多个非线性方程，利用 MATLAB 的 solve 函数进行求解，建立了单污染源测定模型，得出单污染源位置坐标为 (-1,2)，初始浓度为 189.0。

针对问题三：基于单污染源测定模型，建立多个线性方程，运用 MATLAB 的 fsolve 函数[1]，建立双污染源测定模型，循环迭代求解，得出双污染源位置坐标分别为 (1,2)、(-1,2)，初始浓度分别为 144.8，125.1。

本模型建立中运用 MATLAB r2021b(图像绘制，方程求解)，从而使建模过程顺利进行，使所建模型更加精简。

关键词：最小二乘法，solve，fsolve，循环迭代法

一　问题重述

本题考察的是根据所给数据，利用数学手段合理设计点污染源的测定模型，并基于模型解决以下问题：

1. 已知污染物的浓度衰减函数关系，根据所给数据，构建污染源的浓度扩散模型。

2. 基于污染源的浓度扩散模型，根据一个污染源不同点位上浓度值，构建单个污染源位置和初始浓度的确定模型。

3. 基于单个污染源位置和初始浓度的确定模型，根据两个污染源不同点位上浓度值，构建双污染源位置和初始浓度的确定模型。

二　问题分析

问题一：

由题可知，污染源浓度和扩散距离呈指数函数关系，因此建立污染源扩散模型，又从数据中得知初始浓度为 100.0，先将指数函数化简为线性函数，利用最小二乘法，拟合求解函数系数。

问题二：

基于污染源扩散模型，建立单污染源测定模型，需要求解多个非线性方程，利用 MATLAB 的 solve 函数进行求解污染源位置坐标和初始浓度，并代入模型与原数据进行比对和误差分析。

问题三：

基于单污染源扩散模型，建立双污染源测定模型，由于求解方程未知量太多，solve 函数无法求解，运用 fsolve 函数，循环迭代改变初值求解，得出结果后代入模型进行误差分析和模型评估。

三　模型假设

1. 该污染物是点污染源，中心向外扩散，每个方向的扩散没有差异性；

2. 该污染物在平面扩散的过程中，浓度随着扩散距离进行衰减的函数为指数函数关系；

3. 该污染物的扩散环境和测量精度为理想状态；

4. 有多个污染源时，每个点位上的污染浓度为这些污染源的叠加结果。

四　符号说明

符号表示	文字说明
x	横坐标
y	纵坐标
d	直线距离
c	浓度
cc	计算浓度
c_0	初始浓度
m	函数系数
(x_0, y_0)	单个污染源初始坐标
(x_1, y_1)、(x_2, y_2)	双污染源各自的坐标
c_1、c_2	双污染源各自的初始浓度
n	迭代次数

五　模型建立

5.1　模型一　污染物扩散模型

由题意可知，该污染物在平面扩散的过程中，浓度会随着扩散距离 d 进行衰减，其衰减函数大致是指数函数关系，不妨将函数设为：

$$c = c_0 e^{md} \tag{1}$$

由图中数据得知，$d = 0$ 时，初始浓度 $c_0 = 100.0$，等式两边同取自然对数：

$$\ln c = \ln 100 + md \tag{2}$$

利用实验数据，使用最小二乘法，即可拟合出函数系数 m 的值:

最小二乘法能通过最小化误差的平方和寻找数据的最佳函数匹配，应用差分法，当实测值 $\ln c$ 与计算值 $\ln c$ 的差值平方和最小时为最佳的 m 估计值，令

$$\phi = \sum_{i=0}^{11} (\ln cc - \ln c)^2 \tag{3}$$

当 ϕ 最小时，可用函数 ϕ 对 m 求导数，令这个导数等于零:

$$\frac{\mathrm{d}\phi}{\mathrm{d}m} = -2\sum_{i=0}^{11} (d_i - (\ln 100 + md_i)) = 0 \tag{4}$$

得到 m 的求解形式:

$$m = \frac{\sum_{i=1}^{11}(d_i - \bar{d})(\ln c_i - \bar{\ln c})}{\sum_{i=1}^{11}(d_i - \bar{d})^2} \tag{5}$$

决定系数：度量拟合优度 (回归曲线对观测值的拟合程度) 的统计量是决定系数 R^2,R^2 的值越接近 1，说明回归曲线对观测值的拟合程度越好；反之，值越小，说明回归曲线对观测值的拟合程度越差。

$$R^2 = 1 - \frac{\sum_{i=1}^{11}(\ln c_i - \ln cc_i)^2}{\sum_{i=1}^{11}(\ln c_i - \bar{\ln c})^2} \tag{6}$$

求解结果为 $m = -0.02$,$R^2 = 1$, 拟合效果非常好，如图所示：

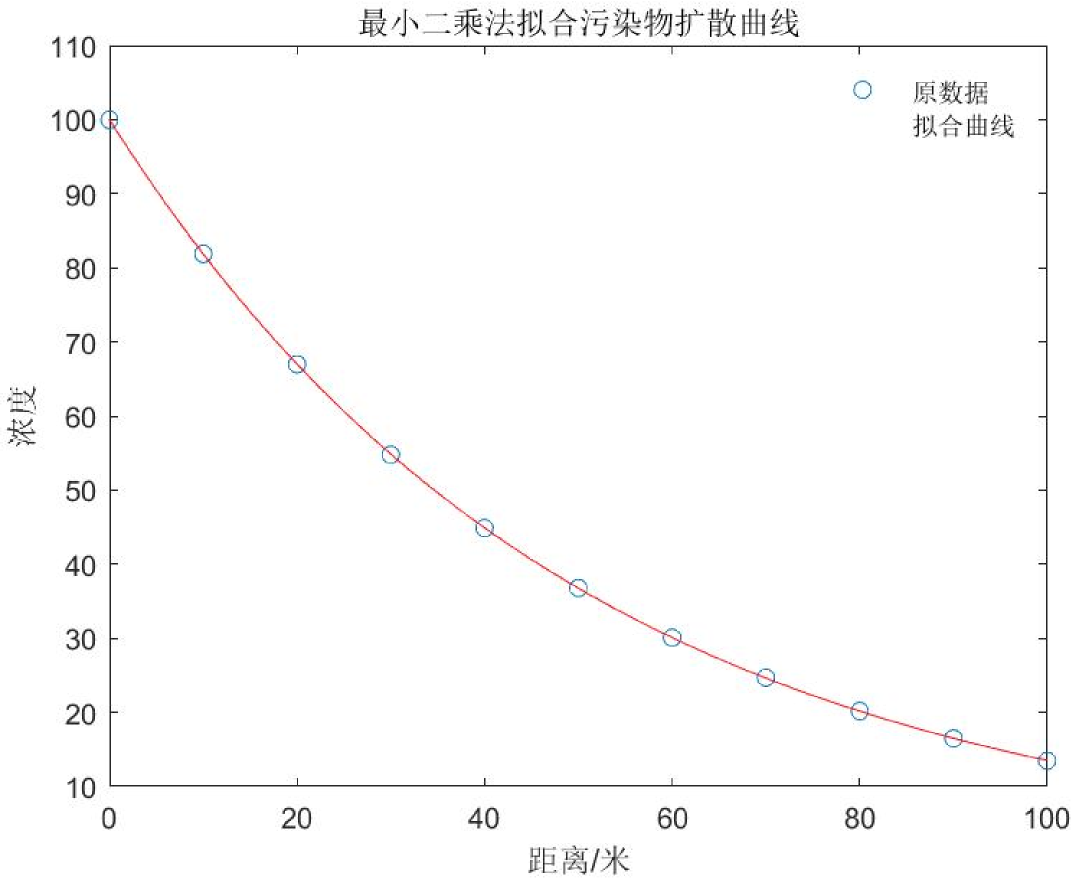

图 1: 最小二乘法拟合污染物扩散曲线

整理方程可得污染物浓度随扩散距离变化的函数：

$$c(d) = c_0 e^{-0.02d} \tag{7}$$

5.2 模型二 单污染源测定模型

根据模型一建立的污染物扩散模型，在平面上，利用每个点位与污染源位置的距离:

$$d=\sqrt{(x-x_0)^2+(y-y_0)^2} \tag{8}$$

代入模型函数求解:

$$c=c_0e^{-0.02\sqrt{(x-x_0)^2+(y-y_0)^2}} \tag{9}$$

题中给出九组数据，可以利用 MATLAB 的 solve 函数将数据分为三组进行求解，并计算均值和方差，最终得出污染源位置为 (-1.0,2.0), 初始浓度为 189.0。

横坐标方差为 0.0007, 纵坐标方差为 0.0006，初始浓度方差为 0.0031, 结果很稳定, 模型方程为:

$$c=189e^{-0.02\sqrt{(x+1)^2+(y-2)^2}} \tag{10}$$

5.3 模型三 双污染源测定模型

根据单个污染源确定模型的思想可知，两个污染源分别扩散，在每个点位上分别叠加，形成的浓度:

$$c=c_1e^{-0.02\sqrt{(x-x_1)^2+(y-y_1)^2}}+c_2e^{-0.02\sqrt{(x-x_2)^2+(y-y_2)^2}} \tag{11}$$

由于方程过于复杂，用 MATLAB 的 solve 函数无法解出结果，而 MATLAB 的 fsolve 函数[1] 需要赋以初值，这里利用循环迭代的方法：首先假定初值，将所给的十八组数据分为三组，一组求解后将解值作为下一组的初值，然后经过三次求解，将得到的值作为下一次迭代的初值，设定迭代次数为 n，进行循环迭代求解。

求解过程发现，当 $n=1000$ 或更大时，只要保证初始浓度 c_1、c_2 不为零，初始坐标都处于正常范围内，最终的求解结果均为:

$c_1=144.8$、$c_2=125.1$，$(x_1,y_1)=(1.0,2.0)$、$(x_2,y_2)=(-1.0,2.0)$

模型方程为:

$$c = 144.8e^{-0.02\sqrt{(x-1)^2+(y-2)^2}} + 125.1e^{-0.02\sqrt{(x+1)^2+(y-2)^2}} \tag{12}$$

六 误差分析

将数据的坐标输入模型中，与原数据浓度进行比对，分析其误差和相对误差。

表 1: 模型一误差和相对误差

序号	1	2	3	4	5	6
误差	0.0000	0.0356	0.0178	0.0637	0.0138	0.0316
相对误差	0.00%	0.04%	0.03%	0.12%	0.03%	0.09%
序号	7	8	9	10	11	
误差	0.0002	0.0586	0.0275	0.0141	0.0192	
相对误差	0.00%	0.24%	0.14%	0.09%	0.14%	

表 2: 模型二误差和相对误差

序号	1	2	3	4	5	6	7	8	9
误差	0.0084	0.0310	0.0397	0.2194	0.0637	0.0449	0.1569	0.1183	0.0952
相对误差	0.01%	0.02%	0.03%	0.14%	0.04%	0.03%	0.09%	0.07%	0.06%

表 3: 模型三误差和相对误差

序号	1	2	3	4	5	6	7	8	9
误差	0.1208	0.1491	0.0657	0.0074	0.0044	0.1150	0.1863	0.0878	0.1286
相对误差	0.06%	0.07%	0.03%	0.00%	0.00%	0.05%	0.08%	0.04%	0.06%
序号	10	11	12	13	14	15	16	17	18
误差	0.1917	0.2531	0.0065	0.0414	0.1386	0.1858	0.1380	0.1934	0.0948
相对误差	0.08%	0.10%	0.00%	0.02%	0.07%	0.09%	0.07%	0.10%	0.04%

可以明显看出，每个模型的误差都很小，三个模型都构建成功。

七 模型评价

7.1 模型一

模型一使用了最小二乘法进行求解，决定系数 $R^2 = 1$，回归曲线对观测值的拟合程度非常好，如果可以有更多数据，拟合效果会更佳。

7.2 模型二

模型二主要使用了 MATLAB 的 solve 函数，但由于所解方程只有三个，可能具有偶然性，解值的误差均很小，说明三个方程的解结果接近，不过 solve 函数所解值是近似解，并没有得到问题的精确解。模型二也可以使用模型三的循环迭代法求解，不过仍无法得到精确解，具有一定的局限性。

7.3 模型三

模型三主要使用了 MATLAB 的 fsolve 函数，又使用了循环迭代法来解决 fsolve 的初值问题，当迭代次数足够大时，可以求出一个精确度很高的近似解，可以大致确定污染源的初始浓度和位置。

八 模型推广

在双污染源测定模型的基础上，如果有更多的数据支撑，也可以建立多污染源测定模型。在实际生活中，污染源的位置确定和初始浓度测定是一个重要的问题，本文把所解决的问题归结为解方程问题，建立的数学模型清晰合理。运用 MATLAB 软件处理数据和进行运算，降低运算量，简单易行，有很大的可操作性，且所得数据较为合理可靠。但在实际运用本方案中还应考虑后来污染因素对模型的影响，在应用的过程中根据实际情况进行灵活改变。

参考文献

[1] GentleMin，非线性方程 (组)：MATLAB 内置函数 solve, vpasolve, fsolve, fzero, roots [MATLAB], https://www.cnblogs.com/gentle-min-601/p/9672221.html

附录

以下代码均用 MATLAB 编写。

8.1 模型一 pollution1.m

```
clear
clc
%d为距离，c为浓度，dp为距离平均值，lncp为浓度自然对数平均值
d=[0:10:100];
c=[100.0 81.9 67.0 54.8 44.9 36.8 30.1 24.7 20.2 16.5 13.5];
lnc=log(c);
dp=sum(d)/11;
lncp=sum(lnc)/11;
for i=1:11
    A(i)=(d(i)-dp)*(lnc(i)-lncp);
    B(i)=(d(i)-dp)^2;
    C(i)=(lnc(i)-lncp)^2;
end
%m均为lnc=ln100+m*d的系数
m=sum(A)/sum(B);
%lncc为拟合浓度的自然对数值
for i=1:11
    lncc(i)=log(100)+m*d(i);
    D=(lnc(i)-lncc(i))^2;
end
%R为决定系数
R=1-sum(D)/sum(C);
%cc为拟合浓度
dd=0:100;
cc=100*exp(m*dd);
%RR为决定系数，越接近1拟合度越好
plot(d,c,'o');
hold on;
plot(dd,cc,'r');
 title ('最小二乘法拟合污染物扩散曲线');
xlabel('距离/米');
ylabel('浓度');
legend('原数据','拟合曲线');
%计算误差
for i=1:11
    cc(i)=100*exp(m*d(i));
    wc(i)=abs(cc(i)-c(i));
    wwc(i)=wc(i)/c(i);
end
 xlswrite ('wc1.xlsx',wc);
 xlswrite ('wwc1.xlsx',wwc);
```

8.2 模型二 pollution2.m

```
clear
clc
x=[-10 10 0 0 0 -5 5 -5 5];
y=[0 0 -10 10 0 5 5 -5 -5];
c=[157.2 151.1 148.5 160.8 180.7 171.1 165.2 160.9 157.2];
%求解方程
syms c0 x0 y0
for i=1:3
    r=c0;
    p=x0;
    q=y0;
    eq1=c(i)-r*exp(-0.02*((x(i)-p)^2+(y(i)-q)^2)^0.5);
    eq2=c(i+3)-r*exp(-0.02*((x(i+3)-p)^2+(y(i+3)-q)^2)^0.5);
    eq3=c(i+6)-r*exp(-0.02*((x(i+6)-p)^2+(y(i+6)-q)^2)^0.5);
    [r,p,q]=solve([eq1,eq2,eq3]);
    e(i)=r;
    f(i)=p;
    g(i)=q;
end
%求初始浓度和坐标的平均值
c0=mean(e);
x0=mean(f);
y0=mean(q);
%计算方差
sc=var(e);
sx=var(f);
sy=var(g);
%计算误差
for i=1:9
    cc(i)=c0*exp(-0.02*((x(i)-x0)^2+(y(i)-y0)^2)^0.5);
    wc(i)=abs(cc(i)-c(i));
    wwc(i)=wc(i)/c(i);
end
 xlswrite ('wc2.xlsx',double(wc));
 xlswrite ('wwc2.xlsx',double(wwc));
```

8.3 模型三 pollution3.m

```
clc
i=1;
global x y c
t =[230,3,2,40,0,2];
%进行1000次循环迭代
for n=1:1000
    while i<19
        t0=t;
        t=fsolve(@fun,t0);
        i=i+6;
    end
    i=1;
end
```

```
t
function equation=fun(t)
    global i
    x=[-10 10 0 0 0 -5 5 -5 5 -5 5 0 0 -10 10 -10 10 5];
    y=[0 0 -10 10 0 5 5 -5 -5 0 0 -5 5 10 10 -10 -10 10];
    c=[219.7 220.6 212.2 229.7 258.1 239.7 240.6 226.8 227.4 241.8
        242.9 234.3 253.4 208.5 209.3 197.1 197.8 223.6];
    %运行失败时先加这一行代码，然后删除即可
    %i=1;
    equation(1)=c(i)-t(1)*exp(-0.02*((x(i)-t(2))^2+(y(i)-t(3))^2)
        ^0.5)-t(4)*exp(-0.02*((x(i)-t(5))^2+(y(i)-t(6))^2)^0.5);
    equation(2)=c(i+1)-t(1)*exp(-0.02*((x(i+1)-t(2))^2+(y(i+1)-t
        (3))^2)^0.5)-t(4)*exp(-0.02*((x(i+1)-t(5))^2+(y(i+1)-t(6))
        ^2)^0.5);
    equation(3)=c(i+2)-t(1)*exp(-0.02*((x(i+2)-t(2))^2+(y(i+2)-t
        (3))^2)^0.5)-t(4)*exp(-0.02*((x(i+2)-t(5))^2+(y(i+2)-t(6))
        ^2)^0.5);
    equation(4)=c(i+3)-t(1)*exp(-0.02*((x(i+3)-t(2))^2+(y(i+3)-t
        (3))^2)^0.5)-t(4)*exp(-0.02*((x(i+3)-t(5))^2+(y(i+3)-t(6))
        ^2)^0.5);
    equation(5)=c(i+4)-t(1)*exp(-0.02*((x(i+4)-t(2))^2+(y(i+4)-t
        (3))^2)^0.5)-t(4)*exp(-0.02*((x(i+4)-t(5))^2+(y(i+4)-t(6))
        ^2)^0.5);
    equation(6)=c(i+5)-t(1)*exp(-0.02*((x(i+5)-t(2))^2+(y(i+5)-t
        (3))^2)^0.5)-t(4)*exp(-0.02*((x(i+5)-t(5))^2+(y(i+5)-t(6))
        ^2)^0.5);
end
```

8.4 **模型三误差分析** error3.m

```
%计算pollution3.m误差
clear
clc
x=[-10 10 0 0 0 -5 5 -5 5 -5 5 0 0 -10 10 -10 10 5];
y=[0 0 -10 10 0 5 5 -5 -5 0 0 -5 5 10 10 -10 -10 10];
c=[219.7 220.6 212.2 229.7 258.1 239.7 240.6 226.8 227.4 241.8
   242.9 234.3 253.4 208.5 209.3 197.1 197.8 223.6];
for i=1:18
    cc(i)=144.8*exp(-0.02*((x(i)-1)^2+(y(i)-2)^2)^0.5)+125.1*exp
        (-0.02*((x(i)+1)^2+(y(i)-2)^2)^0.5);
    wc(i)=abs((cc(i)-c(i)));
    wwc(i)=wc(i)/c(i);
end
xlswrite('wc3.xlsx',double(wc));
xlswrite('wwc3.xlsx',double(wwc));
```