

北京化工大学本科毕业设计（论文）

# 微生物土壤运移模型的求解及仿真软件编制

## 文献综述

陆秋文

北京化工大学生命科学与技术学院

指导教师

周 延

二〇一三年六月三日

## 1 研究背景与意义

在环境工程学科中有这样一个问题,即需要对受污染的土壤进行修复,其中较好的方式是原位修复.所谓原位修复,指的是在土壤中注入分离得到的或基因工程合成的细菌或营养物质,令其运动到受污染的区域,分解有毒物质,达到治理目的.这样,我们需要对土壤中微生物的运动规律进行研究.

早在 20 世纪 40 年代初,国内外学者已开始对土壤中微生物吸附和运移进行研究.早期的研究工作主要集中在垃圾处理、传染病、污水灌溉、纯化受微生物和有机物污染的水等方面.现已扩展到原位生物修复、提高石油开采量、放射性物质和有机污染物的携带运移、提高冶炼率、核废料处理和根系层内病害的生物防治及养分的转化等方面.

微生物在土壤中的运移过程看似简单,实际很复杂.其运移机理包括生长、吸附、解吸、沉积(过滤、布朗扩散、截流、沉降)、腐解、钝化、滞留等过程,为了确定微生物的运移速率、时间、分布范围,最大限度提高细菌降解的作用,减少再污染,对微生物在土壤中运移及其影响因素建立数学模型,进行定量研究,是很有必要的.

有两个方面对微生物在土壤中的运动产生影响,第一是土壤地下水环境的影响,第二是是微生物自身的因素<sup>[7]</sup>.

## 2 影响微生物运动的因素

微生物在土壤多孔介质中的迁移受到各种非生物和生物因素的影响.这些因素可概括为两个主要方面——水文地质因素和微生物因素.其中水文地质因素包括多孔介质结构、有机质的含量、氧化膜以及地下水水流的速度、溶液化学成分、溶液 pH、离子强度等;细胞的生理状态、细胞的生长与衰亡、细胞的趋磁性、细胞的吸附过程、过滤效应,属于微生物因素<sup>[7]</sup>.

### 2.1 水文地质因素

微生物生活的环境总是存在稳定或不稳定的水相.田间观察表明,水的流量和流速会影响微生物的迁移范围,在土柱实验中也观察到随着土柱水流速度的增加,细菌流出量也增加.总的来说,水的高流速能减少作用时间,使微生物吸附的可能性降低.

另外,土壤中颗粒的大小和孔径的大小也是影响微生物迁移的重要因子,影响微生物细胞在液相中的过滤作用.

## 2.2 微生物因素

### 2.2.1 吸附作用

吸附发生在微生物细胞和颗粒表面接触时. 要对微生物细胞表面和多孔介质表面之间的各种相互化学作用加以考虑. 吸附是物理化学过程, 可以分为可逆与不可逆过程. 主要有静电、疏水作用和范德华力. 一般来说, 细胞和颗粒表面的静电作用是排斥的, 因为细胞和颗粒表面都是带负电荷的. 而土壤矿物质表面的负电荷是与一些有机质中含有的大量羟基官能团等有关的. 疏水作用和范德华力趋于相互吸引. 疏水作用指水环境中非极性基团的聚合趋势. 疏水作用对微生物迁移的影响作用依赖于细胞表面的疏水性以及基质表面特性和孔隙溶液特性.

范德华力通常导致中性分子间的相互作用. 这些中性分子有着动态的电荷分布. 当两个分子相互靠近时, 这种电荷分布对两个分子之间的作用起到促进作用, 范德华力增加到最大值, 随后减小直至成为斥力. 当引力超过斥力时, 就发生初始的吸附. 初始吸附后, 细胞能够不可逆地吸附到颗粒表面. 不可逆吸附过程的发生是细胞表面结构和固体表面相互作用的结果, 或者说是产生的胞外多糖把微生物细胞黏附到表面造成的.

### 2.2.2 生理状态

微生物的大小和形状会影响其在土壤中的吸附效果, 影响它们的迁移潜力. 细菌所处的生理状态会对它们的形状大小产生影响, 从而对迁移也产生影响. 当微生物所需的营养物不受限制时, 即微生物细胞处于正常的生理状态时, 会产生包裹在细胞外表面的多聚物, 多聚物会增加细胞的有效直径并促进吸附在固体表面. 当微生物处于饥饿条件下, 细胞会缩小并脱去荚膜层, 增加了它们的迁移潜力.

### 2.2.3 运动性和趋化性

鞭毛是微生物的运动器官, 鞭毛的运动能够引起菌体的运动. 关于鞭毛的有无对于微生物迁移的影响结果并不一致. 生物趋化性是细菌对不同化学物浓度所产生的吸引聚集或避离排斥的反应运动. 有研究报道认为趋化性对细菌在土层中迁移起很重要的作用.

### 3 基本数学理论

在这一部分,我们简要地介绍在本研究中所用到的数学工具与方法,其中重要的定理和结论,我们还会对它们的推导过程进行简要的介绍.

#### 3.1 线性二阶偏微分方程理论

偏微分方程指的是含有未知函数  $u(x_1, x_2, \dots, x_n, t)$  的偏导数的方程<sup>[7]</sup>. 通常采用  $t$  表示时间变量,  $u(x_1, x_2, \dots, x_n, t)$  表示空间变量. 考虑  $\vec{x} = (x_1, x_2, \dots, x_n, t) \in \mathbb{R}$ , 当  $n = 2, 3$  时, 可以记为  $\vec{x} = (x, y)$  或  $\vec{x} = (x, y, z)$ . 我们记 Laplace 算子为

$$\Delta = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \dots + \frac{\partial^2}{\partial x_n^2}$$

记

$$\nabla = \vec{e}_1 \frac{\partial}{\partial x_1} + \vec{e}_2 \frac{\partial}{\partial x_2} + \dots + \vec{e}_n \frac{\partial}{\partial x_n}$$

其中,  $\vec{e}_i (i = 1, 2, \dots, n)$  是坐标轴上的单位向量,  $\Delta = \nabla \cdot \nabla$ .

下面, 我们来介绍二阶方程, 设  $u = u(x_1, x_2, \dots, x_n)$ , 有

$$\sum_{i,j=1}^n a_{ij} \frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_{i=1}^n b_i \frac{\partial u}{\partial x_i} + cu = f \quad (1)$$

称为二阶拟线性方程.

在方程 (1) 中, 我们假设  $a_{ij} = a_{ji}$ . 这样矩阵  $A = [a_{ij}]$  是一个  $n \times n$  的矩阵. 若在点  $(x_1, x_2, \dots, x_n)$  上,  $A$  是正定的或负定的, 那么方程 (1) 为椭圆形方程. 如果  $A$  的特征值至少有一个为零, 则称为抛物型方程. 如果  $A$  的特征值都不是零, 且有  $n - 1$  个是同号, 则称为双曲型方程.

下面我们重点讨论两个自变量的二阶方程. 设  $u = u(x, y)$ , 记  $p = \left(u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}\right)$ , 二阶拟线性方程写为

$$a(x, y, p) \frac{\partial^2 u}{\partial x^2} + 2b(x, y, p) \frac{\partial^2 u}{\partial x \partial y} + c(x, y, p) \frac{\partial^2 u}{\partial y^2} + f(x, y, p) = 0 \quad (2)$$

其中,  $a, b, c, f$  均为自变量的连续函数.

如果方程 (2) 中,  $a, b, c$  与  $p$  无关, 且

$$f(x, y, p) = d(x, y) \frac{\partial u}{\partial x} + e(x, y) \frac{\partial u}{\partial y} + r(x, y)u + s(x, y) \quad (3)$$

则方程 (2) 为二阶线性方程.

对于方程 (2) 中的系数  $a, b, c$ , 若对于固定的  $(x, y, u)$  有  $ac - b^2 > 0$ , 则方程为椭圆型的. 若  $ac - b^2 = 0$ , 则方程为抛物型的. 若  $ac - b^2 < 0$ , 则方程为双曲型的.

$x-y$  平面上的曲线  $x = \phi_1(s), y = \phi_2(s)$  ( $s$  是参数). 若

$$a[\phi_2'(s)]^2 - 2b\phi_2'(s)\phi_1'(s) + c[\phi_1'(s)]^2 = 0 \quad (4)$$

则称此曲线为方程 (2) 的特征曲线, 而方程 (2) 称为特征方程,  $x-y$  平面上的向量  $(\beta_1, \beta_2)$ , 若满足

$$\alpha\beta_2^2 - 2b\beta_1\beta_2 + c\beta_1^2 = 0 \quad (5)$$

则称为方程 (2) 在点  $(x, y)$  处的特征方向.

显然, 特征方向是特征曲线在点  $(x, y)$  点处的切线方向. 若特征曲线的方程可以写成  $y = y(x)$  的形式, 则  $y(x)$  满足

$$\frac{dy}{dx} = \frac{b \pm \sqrt{b^2 - ac}}{a} \quad (6)$$

可以看到, 对于双曲型方程, 在  $x-y$  平面上有两簇特征曲线, 而抛物型方程在  $x-y$  平面上只有一簇曲线. 椭圆形方程则没有实的特征曲线.

接下来介绍一下定解问题, 我们考虑一个典型的椭圆型方程的边值问题, 设  $u = u(x, y)$ , 有

$$\begin{cases} -\left[\frac{\partial}{\partial x}\left(k\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(k\frac{\partial u}{\partial y}\right)\right] = F(x, y), & x, y \in \Omega \\ u = g(x, y) & x, y \in \partial\Omega \end{cases} \quad (7)$$

其中,  $\Omega \subseteq \mathbb{R}^2$ ,  $k > 0, k, F, g$  均为  $x, y$  的函数, 我们看到, 这类边界条件称为第一类边界条件.

相应地我们看到

$$k\frac{\partial u}{\partial \vec{n}} = g(x, y), \quad x, y \in \partial\Omega \quad (8)$$

称为第二类边界条件, 以及

$$k\frac{\partial u}{\partial \vec{n}} + \partial u = g(x, y), \quad x, y \in \partial\Omega \quad (9)$$

为第三类边界条件.

### 3.2 Fourier 变换

我们将 Fourier 变换用于偏微分方程的分析. 有 Fourier 积分公式<sup>[7]</sup>, 设  $\int_{-\infty}^{+\infty} |v(x)|^2 dx < \infty$ , 有

$$v(x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} v(\xi) \exp(-i\lambda(\xi - x)) d\xi d\lambda \quad (10)$$

其中,  $i = \sqrt{-1}$  为虚数单位. 下面我们来推导这个公式.

设  $v(x)$  在  $[-L, L]$  可以展开成 Fourier 级数,

$$v(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left( a_n \cos \frac{n\pi x}{L} + b_n \sin \frac{n\pi x}{L} + b_n \sin \frac{n\pi x}{L} \right) \quad (11)$$

其中, Fourier 系数为

$$a_n = \frac{1}{L} \int_{-L}^L v(\xi) \cos \frac{n\pi \xi}{L} d\xi, \quad n = 0, 1, \dots \quad (12)$$

$$b_n = \frac{1}{L} \int_{-L}^L v(\xi) \sin \frac{n\pi \xi}{L} d\xi, \quad n = 1, 2, \dots \quad (13)$$

将式 (12) 和式 (13) 代入 (11) 中, 得

$$v(x) = \frac{1}{2L} \int_{-L}^L v(\xi) d\xi + \frac{1}{L} \sum_{n=1}^{\infty} \int_{-L}^L v(\xi) \cos \frac{n\pi}{L} (\xi - x) d\xi \quad (14)$$

设  $v(x)$  在  $(-\infty, +\infty)$  绝对可积, 令  $L \rightarrow \infty$ , 则式 (14) 第一项趋于零, 记  $\mu_n = n\pi/L$ , 有

$$\Delta\mu_n = \mu_{n+1} - \mu_n = \frac{\pi}{L}, \quad \mu_n = n\Delta\mu_n \quad (15)$$

则当  $L \rightarrow \infty$  时, 式 (14) 为

$$v(x) = \lim_{L \rightarrow \infty} \sum_{n=1}^{\infty} \left[ \frac{1}{\pi} \int_{-L}^L v(\xi) \cos \mu_n (\xi - x) d\xi \right] \Delta\mu_n$$

和式成为区间  $[0, +\infty)$  的积分式, 即

$$v(x) = \int_0^{\infty} \left[ \frac{1}{\pi} \int_{-\infty}^{+\infty} v(\xi) \cos \mu_n (\xi - x) d\xi \right] d\mu_n \quad (16)$$

将三角函数改写为指数函数 (欧拉定理), 有

$$\cos \mu (\xi - x) = \frac{1}{2} [\exp(i\mu(\xi - x)) + \exp(-i\mu(\xi - x))]$$

代入式 (16), 在第一项中令  $\lambda = -\mu$ , 即

$$\int_0^{\infty} \left[ \frac{1}{2\pi} \int_{-\infty}^{+\infty} v(\xi) \exp(-i\lambda\xi) d\xi \right] d\lambda$$

再与第二项相加, 得式 (10), 即 *Fourier* 积分公式.

将式 (10) 写为

$$v(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \left[ \frac{1}{\sqrt{2\pi}} v(\xi) e^{-i\lambda\xi} d\xi \right] e^{i\lambda x} d\lambda$$

将积分变量  $\xi$  换为  $x$ , 令

$$\hat{v}(\lambda) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} v(x) e^{-i\lambda x} dx \quad (17)$$

有

$$v(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \hat{v}(\lambda) e^{-i\lambda x} d\lambda \quad (18)$$

则式 (17) 的  $\hat{v}(\lambda)$  称为  $v(x)$  的 *Fourier* 变换, 而式 (18) 的  $v(x)$  称为  $\hat{v}(\lambda)$  的 *Fourier* 逆变换.

类似三角级数的性质, 可以证明

$$\int_{-\infty}^{+\infty} |v(x)|^2 dx = \int_{-\infty}^{+\infty} |\hat{v}(\lambda)|^2 d\lambda \quad (19)$$

称为 Parseval 关系式, 同时还有

$$v = 0 \quad \Leftrightarrow \quad \hat{v} = 0 \quad (20)$$

### 3.3 Laplace 变换

Laplace 变换<sup>[7]</sup> 用于解初值问题. 即已知某物理量在  $t = 0$  时的值  $f(0)$ , 求解在初始时刻之后的变化  $f(t)$ . 我们考虑

$$f(t) = 0, \quad (t < 0) \quad (21)$$

为了获得较为宽泛的变换条件, 构造  $g(t)$ ,

$$g(t) = e^{-\sigma t} f(t) \quad (22)$$

这里的  $e^{-\sigma t}$  为收敛因子, 保证  $g(t)$  在  $(-\infty, +\infty)$  上绝对可积, 于是可以对  $g(t)$  进行 *Fourier* 变换

$$G(\omega) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} g(t) e^{-i\omega t} dt = \frac{1}{2\pi} \int_0^{\infty} f(t) e^{-(\sigma + i\omega)t} dt$$

将  $\sigma + i\omega$  记为  $s$ , 并将  $G(\omega)$  改为  $\bar{f}(s)/2\pi$ , 则

$$\hat{f}(s) = \int_0^\infty f(t)e^{-st}dt \quad (23)$$

其中,  $\bar{f}(s)$  称为  $f(t)$  的 Laplace 变换函数, 式 (23) 称为 Laplace 变换.

$G(\omega)$  的 Fourier 逆变换<sup>[7]</sup> 为

$$g(t) = \int_{-\infty}^{+\infty} G(\omega)e^{i\omega t}d\omega = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \bar{f}(\sigma + i\omega)e^{i\omega t}d\omega$$

即

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \bar{f}(\sigma + i\omega)e^{\sigma + i\omega t}d\omega \quad (24)$$

由于  $\sigma + i\omega = s$ , 有  $d\omega = \frac{1}{i}ds$ , 即

$$f(t) = \frac{1}{2\pi i} \int_{\sigma-i\infty}^{\sigma+i\infty} \bar{f}(s)e^{is}ds \quad (25)$$

$\bar{f}(s)$  称为像函数, 而  $f(t)$  称为原函数, 它们之间的关系可以写为

$$\bar{f}(p) = L[f(t)] \quad (26)$$

$$f(t) = L^{-1}[\bar{f}(s)] \quad (27)$$

下面来介绍 Laplace 变换的反演, 我们推广约当定理<sup>[7]</sup>,

**定理 3.1 (推广的约当定理)** 设  $c_r$  是以  $p = 0$  为圆心, 以  $r$  为半径的圆周在直线  $\operatorname{Re} p = a (a > 0)$  左侧的圆弧. 若当  $|p| \rightarrow \infty$  时,  $\bar{f}(p)$  在  $\frac{\pi}{2} - \delta \leq \operatorname{Arg} p \leq \frac{3}{2}\pi + \delta$  中一致趋向于零, 则

$$\lim_{r \rightarrow \infty} \int_{C_R} \bar{f}(s)e^{st}ds = 0, \quad (t > 0). \quad (28)$$

我们来证明上面的定理, 如图 1, 有

$$\int_{C_R} \bar{f}(p)e^{st}dp = \int_{\widehat{AB}} \bar{f}(s)e^{st}dt + \int_{\widehat{BCD}} \bar{f}(s)e^{st}dp + \int_{\widehat{DE}} \bar{f}(s)e^{st}dp.$$

对于右端第二个积分, 做代数变换令  $s = iz$ , 由约当定理得

$$\lim_{R \rightarrow \infty} \int_{\widehat{BCD}} \bar{f}(s)e^{st}dp = \lim_{R \rightarrow \infty} i \int_{C'_R} \bar{f}(iz)e^{itz}dz = 0$$

现在来估计  $\widehat{AB}$  上的积分值, 任意给出  $\varepsilon > 0$ , 取  $R$  足够大, 使得  $|\bar{f}(s)| < \varepsilon$ , 则

$$\begin{aligned} \left| \int_{\widehat{AB}} \bar{f}(s)e^{st}dp \right| &\leq \int_{\widehat{AB}} |\exp(tR(\cos \theta + i \sin \theta))| |\operatorname{Re} i d\theta| \\ &= \int_{\widehat{AB}} |\bar{f}(s)| e^{tR \cos \theta} R d\theta < \varepsilon e^{at} R \alpha \end{aligned}$$



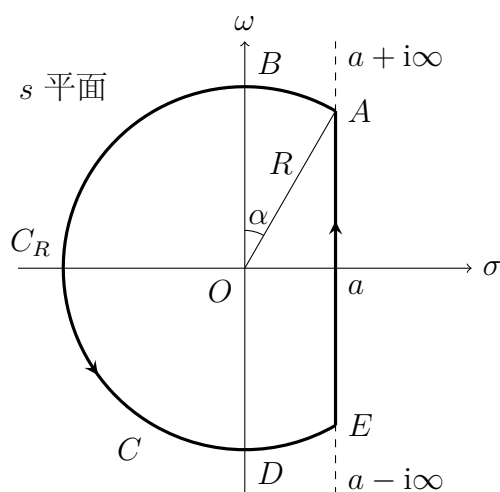


图 1 推广约当定理证明

其中,  $\alpha$  为常数. 在  $\widehat{AB}$  上,  $R \cos \theta \leq 0$ . 而当  $R \rightarrow \infty$  时,  $\alpha \rightarrow 0$ , 但由于  $R\alpha \sim R \sin \alpha = a$ , 有

$$\lim_{R \rightarrow \infty} \int_{\widehat{AB}} \bar{f}(s) e^{st} ds = 0$$

同理

$$\lim_{R \rightarrow \infty} \int_{\widehat{DE}} \bar{f}(s) e^{st} ds = 0$$

证明了 (28).

现在考虑回路积分

$$\oint_l \bar{f}(s) e^{st} ds = \int_E^A \bar{f}(s) e^{st} dp + \int_{C_R} \bar{f}(s) e^{st} dp$$

于  $R \rightarrow \infty$  时, 从而

$$\frac{1}{2\pi i} \int_{a-i\infty}^{a+i\infty} \bar{f}(s) e^{st} ds = \sum \text{Res}[\bar{f}(s) e^{st}]$$

即

$$f(t) = \sum \text{Res}[\bar{f}(s) e^{st}] \quad (29)$$

## 4 数值计算方法

在这一节中我们介绍在研究中用到的数值计算方法. 许多情况下问题是很难得到解析解的, 得到数值解是解出这一问题的唯一途径. 所以, 快速地、高精度地求得数值解是非常重要而且必要的. 下面, 我们给出两个重要的求解常微分方程的数值计算方法.

#### 4.1 欧拉折线法

欧拉折线法<sup>[7]</sup>是求解常微分方程的初值问题的一种比较简单有效的方法, 考虑一个一阶常微分方程的初值问题, 可以写为

$$\begin{cases} y' = f(x, y) & x \in [a, b] \\ y(a) = y_0 \end{cases} \quad (30)$$

将方程 (30) 中的导数项  $y'$  用差商逼近, 在节点  $x_n$  处, 可以写为

$$y'(x_n) = f(x_n, y(x_n)) \quad (31)$$

由于

$$y'(x_n) \approx \frac{y(x_{n+1}) - y(x_n)}{h}$$

故

$$y(x_{n+1}) = y_n + hf(x_n, y(x_n)). \quad (32)$$

因为真解  $y(x)$  是未知的, 所以将  $y(x_i)$  的近似值  $y_i (i = n, n+1)$  代入上式, 得到欧拉公式

$$y_{n+1} = y_n + hf(x_n, y_n). \quad (33)$$

#### 4.2 龙格库塔公式

一般的显式龙格库塔公式<sup>[7]</sup>的形式为

$$\begin{cases} y_{n+1} = y_n + \sum_{i=1}^r \omega_i k_i \\ k_1 = hf(x_n, y_n) \\ k_i = hf(x_n + \alpha_i h, y_n + \sum_{j=1}^{i-1} \beta_{ij} k_j), \quad i = 2, 3, \dots, r \end{cases} \quad (34)$$

下面, 我们来介绍二段龙格库塔公式的推导过程. 已知龙格库塔公式的一般形式为方程 (34), 得到二段龙格库塔公式的格式为

$$\begin{cases} y_{n+1} = y_n + \omega_1 k_1 + \omega_2 k_2 \\ k_1 = hf(x_n, y_n) \\ k_2 = hf(x_n + \alpha_2 h, y_n + \beta_{21} k_1) \end{cases} \quad (35)$$

由式 (35) 得

$$y_{n+1} = y_n + \omega_1 hf(x_n, y_n) + \omega_2 hf(x_n + \alpha_2 h, y_n + \beta_{21} hf(x_n, y_n)) \quad (36)$$

记  $f(x_n, y_n) = f_n$ , 利用 Taylor 展开得

$$f(x_n + \alpha_2 h + \beta_{21} h f_n) = f_n + \alpha_2 h f_x + \beta_{21} h f_y f_n + O(h^2) \quad (37)$$

其中,  $f_x$  和  $f_y$  分别表示  $\partial f / \partial x$  和  $\partial f / \partial y$  在  $(x_n, y_n)$  处的值.

假设  $y_n = y(x_n)$ , 并将上式代入式 (36), 化简得

$$\begin{aligned} y_{n+1} &= y(x_n) + \omega_1 h f_n + \omega_2 (h f_n + h^2 (\alpha_2 f_x + \beta_{21} f_x f_n)) + O(h^3) \\ &= y(x_n) + (\omega_1 f_n + \omega_2 f_n) h + \omega_2 (\alpha_2 f_x + \beta_{21} f_y f_n) h^2 + O(h^3) \end{aligned} \quad (38)$$

将  $y(x_{n+1})$  在点  $x = x_n$  处进行 Taylor 展开

$$\begin{aligned} y(x_{n+1}) &= y(x_n + h) = y(x_n) + y'(x_n) h + \frac{1}{2!} y''(x_n) h^2 + O(h^3) \\ &= y(x_n) + f_n h + \frac{1}{2} (f_x + f_x f_n) h^2 + O(h^3) \end{aligned} \quad (39)$$

逐项比较式 (38) 和式 (39). 为了使  $y(x_{n+1}) - y_{n+1} = O(h^3)$ , 则有

$$\begin{cases} \omega_1 + \omega_2 = 1 \\ \omega_2 \alpha_2 = \frac{1}{2} \\ \omega_2 \beta_{21} = \frac{1}{2} \end{cases} \quad (40)$$

若令  $\omega_1 = 0$ , 从 (40) 解得  $\omega_2 = 1, \alpha_2 = 1/2, \beta_{21} = 1/2$ , 则得到一个二阶二段的龙格库塔公式

$$\begin{cases} y_{n+1} = y_n + k_2 \\ k_1 = h f(x_n, y_n) \\ k_2 = h f\left(x_n + \frac{1}{2} h, y_n + \frac{1}{2} k_1\right). \end{cases} \quad (41)$$

该格式称为中点公式.

### 4.3 矩阵的三角分解、追赶法

将一个  $n$  阶矩阵分解成结构简单的三角形矩阵的过程称为矩阵的三角分解<sup>[7]</sup>.

考虑矩阵  $A$ , 其中的主元素  $a_{ii}^{(i)} \neq 0 (i = 1, 2, \dots, n)$ , 则令

$$L_i = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & m_{i+1,i} & 1 & \\ & & \vdots & & \ddots \\ & & m_{n,i} & & & 1 \end{bmatrix}, \quad i = 1, 2, \dots, n-1. \quad (42)$$

使得

$$L_{n-1} \cdots L_2 L_1 A = \begin{bmatrix} a_1 1^{(1)} & a_1 2^{(1)} & \cdots & a_{1,n-1}^{(1)} & a_{1n}^{(1)} \\ 0 & a_2 2^{(2)} & \cdots & a_{a,n-1}^{(1)} & a_{2n}^{(2)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & a_{n-1,n}^{(n-1)} & a_{n-1,n}^{(n-1)} \\ 0 & 0 & \cdots & 0 & a_{mn}^{(n)} \end{bmatrix} \quad (43)$$

计算得

$$L_i^{-1} = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & -m_{i+1,i} & 1 & \\ & & \vdots & & \ddots \\ & & -m_{n,i} & & 1 \end{bmatrix}, \quad i = 1, 2, \dots, n-1. \quad (44)$$

若记式 (43) 上三角形阵为  $U$ , 则有

$$A = L_1^{-1} L_2^{-1} \cdots L_{n-1}^{-1} U.$$

通过矩阵的乘法运算, 有

$$L_1^{-1} L_2^{-1} \cdots L_{n-1}^{-1} = \begin{bmatrix} 1 & & & & \\ -m_{21} & 1 & & & \\ \vdots & \ddots & & \ddots & \\ -m_{n1} & \cdots & -m_{n,n-1} & 1 \end{bmatrix}. \quad (45)$$

记  $L = L_1^{-1} L_2^{-1} \cdots L_{n-1}^{-1}$ , 则有  $A = LU$ . 其中,  $L$  为单位下三角矩阵,  $U$  是上三角矩阵. 这样的分解称为 Doolittle 分解.

我们考虑一种特殊的情况, 令  $A = LU$ , 其中  $L$  为下三角矩阵,  $U$  是单位上三角矩阵, 称为 Crout 分解.

下面我们来介绍追赶法<sup>[7]</sup>. 设有  $n$  阶方程组  $Ax = d$ , 其中  $A$  为三对角矩阵, 即

$$A = \begin{bmatrix} b_1 & c_1 & & & \\ a_2 & b_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & a_n & b_n \end{bmatrix}, \quad d = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_{n-1} \\ d_n \end{bmatrix} \quad (46)$$

对  $A$  作 Court 分解, 有

$$L = \begin{bmatrix} l_1 & & & & \\ v_2 & l_2 & & & \\ & \ddots & \ddots & & \\ & & v_{n-1} & l_{n-1} & \\ & & & v_n & l_n \end{bmatrix}, \quad U = \begin{bmatrix} 1 & u_1 & & & \\ & 1 & u_2 & & \\ & & \dots & \dots & \\ & & & 1 & u_{n-1} \\ & & & & 1 \end{bmatrix} \quad (47)$$

设  $y = (y_1, y_2, \dots, y_n)^T$ , 得到追赶法的计算方法:

1. 对  $i = 1, 2, \dots, n-1$  做

$$\begin{cases} l_i = b_i - a_i u_{i-1}, \\ y_i = (d_i - y_{i-1} a_i) / l_i, \\ u_i = c_i / l_i \end{cases}$$

在此处令  $u_0 = y_0 = a_1 = 0$ ;

2.  $l_n = b_n - a_n u_{n-1}, y_n = (d_n - y_{n-1} a_n) / l_n$ ;
3.  $x_n = y_n$ ;
4. 对  $i = 1, 2, \dots, n-1$  做

$$x_i = y_i - u_i x_{i+1};$$

这一解法称为追赶法.

## 5 研究问题的方法

研究科学问题的方法不外乎有两种, 一种为理论分析研究, 另一种为实验研究. 在本研究中, 我们采用理论研究的方法. 通过对微生物运动过程的分析, 建立起数学模型, 然后测定数学模型的各项参数, 得到微生物运动在数学上的表达. 对数学模型进行求解和仿真, 得到结果后再验证.

在本研究中, 我们对数学模型的求解和仿真是其重点, 采用的方法主要是计算机仿真. 下面, 我们介绍一些在计算机仿真中所用到的工具<sup>[7]</sup>.

### 5.1 MATLAB 矩阵实验室系统

MATLAB(矩阵实验室) 是 Matrix Laboratory 的缩写, 是一款由美国 The Math-Works 公司出品的商业数学软件. MATLAB 是一种用于算法开发、数据可视化、数

据分析以及数值计算的高级技术计算语言和交互式环境. 除了矩阵运算、绘制函数/数据图像等常用功能外, MATLAB 还可以用来创建用户界面及与调用其它语言(包括 C/C++ 和 FORTRAN) 编写的程序.

尽管 MATLAB 主要用于数值运算, 但利用为数众多的附加工具箱 (Toolbox) 它也适合不同领域的应用, 例如控制系统设计与分析、图像处理、信号处理与通讯、金融建模和分析等.

另外还有一个配套软件包 Simulink, 提供了一个可视化开发环境, 常用于系统模拟、动态/嵌入式系统开发等方面.

## 5.2 Python

Python 是一种面向对象、直译式计算机程序设计语言, 由 Guido van Rossum 于 1989 年底发明, 第一个公开发行人版发行于 1991 年. Python 语法简捷而清晰, 具有丰富和强大的类库. 它常被称为胶水语言, 它能够很轻松的把用其他语言制作的各种模块 (尤其是 C/C++) 轻松地联结在一起. 常见的一种应用情形是, 使用 Python 快速生成程序的原型 (有时甚至是程序的最终界面), 然后对其中有特别要求的部分, 用更合适的语言改写. 比如 3D 游戏中的图形渲染模块, 速度要求非常高, 就可以用 C++ 重写.

利用它的快速开发的特定, 可以使用这种语言来进行程序的编制.

## 5.3 OpenGL

OpenGL<sup>[7]</sup> 是个定义了一个跨编程语言、跨平台的编程接口的规格, 它用于三维图象 (二维的亦可). OpenGL 是个专业的图形程序接口, 是一个功能强大, 调用方便的底层图形库.

OpenGL(Open Graphics Library) 是个定义了一个跨编程语言、跨平台的程序接口 (Application Programming Interface) 的规格, 它用于生成二维、三维图像. 这个接口由近三百五十个不同的函数调用组成, 用来从简单的图形比特绘制复杂的三维景象. OpenGL 常用于 CAD、虚拟实境、科学可视化程序和电子游戏开发.

PyOpenGL 是 OpenGL 在 Python 语言上的一个绑定, 提供了 Python 访问 OpenGL 库的接口. 我们可以利用这个绑定访问 OpenGL, 从而实现 PDE 的图像绘制.

## 5.4 SciPy

SciPy<sup>[7]</sup> 是一个开源的 Python 算法库和数学工具包.

SciPy 包含的模块有最优化、线性代数、积分、插值、特殊函数、快速傅里叶变换、信号处理和图像处理、常微分方程求解和其他科学与工程中常用的计算, 与 MATLAB 功能类似. 由于它是基于 BSD 协议发布的, 可以使用在我们的仿真程序中.

## 6 总结

在本综述中我们介绍了本课题研究的背景与意义, 说明了对微生物运动规律研究对环境修复的重要作用. 同时介绍了研究中所涉及的数学理论与方法, 介绍了相应的数值计算方法, 尤其重点地介绍了微分方程的数值求解方法. 最后, 介绍了本研究中经常利用到的计算机程序设计语言和软件包.