

摘要

基因调控网络是由一组基因、蛋白质、RNA、小分子以及它们之间的相互作用所构成的一种生化系统。对基因调控网络的算法研究在理论和实践上有着重要的意义。

本文首先给出化学主方程，并介绍几种精确随机模拟算法和快速近似算法。其次，对刚性生化系统的混合模拟算法进行了介绍，并给出多尺度随机模拟算法，然后在其基础上发展了改进多尺度随机模拟算法。数值模拟结果证实了改进多尺度随机模拟算法的优点。最后介绍时滞生化系统的精确模拟算法，并提出一种模拟时滞生物化学反应系统的自适应 τ -Leap算法(DA τ -Leap)。该算法能够较大地提高模拟速度。数值实验证实了DA τ -Leap算法的优点。

关键词：基因调控网络；多尺度算法；改进多尺度算法；时滞生化系统；自适应 τ -Leap算法

ABSTRACT

The gene regulatory network(GRN) is a biochemical system which is composed of genes, proteins, RNAs, small molecules together with their interactive effect . The research of algorithm for (GRN) is very meaningful in theory and practice.

First, this paper gives chemical master equation which depicts the stochastic system and presents some exact stochastic simulating algorithms and fast approximate algorithms. Then, we introduce the hybrid algorithm for stiffness biochemical system and develop multiscale stochastic simulation algorithm (MSSA). Improved multiscale stochastic simulation algorithm (IMSS τ) is developed. Numerical experiments demonstrate the advantage of IMSS τ . At last, the exact stochastic algorithm for simulating biochemical systems with delay is introduced and an adaptive τ - Leap algorithm for simulating chemically reacting system with delays is proposed. This algorithm largely accelerates the speed of simulating. Numerical experiments demonstrate the advantage of DA τ - Leap algorithm.

Key words: GRN; MSSA; IMSS τ ; Biochemical system with delays; DA τ - Leap

原创性声明

本人声明：所呈交的论文是本人在导师指导下进行的研究工作。除了文中特别加以标注和致谢的地方外，论文中不包含其他人已发表或撰写过的研究成果。参与同一工作的其他同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

签 名：闫正楼 日期：2008.6.5

本论文使用授权说明

本人完全了解上海大学有关保留、使用学位论文的规定，即：学校有权保留论文及送交论文复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内容。

（保密的论文在解密后应遵守此规定）

签 名：闫正楼 导师签名：程飞 日期：2008.6.5

第一章 绪论

1.1 系统生物学

系统生物学^[1, 2, 3, 4, 5]是生物学的新领域,其目的在于从系统水平上理解生物系统,力求阐述作为一个系统的生物系统。系统生物学是在细胞、组织、器官和生物体整体水平上研究结构和功能各异分子之间的相互作用,并通过计算机模拟来定量地描述和预测生物功能、表型和行为。系统生物学将在基因组序列的基础上完成由生命密码到生命过程的研究。这是一个逐步整合的过程:由对生物体内各种分子的鉴别及其相互作用的认识到对途径、网络、模块的研究并最终完成整个生命活动的路线图。系统生物学的研究大致有以下四个方面^[2, 3]:

1) 系统结构 (System Structures) 的识别和研究 这包括基因相互作用、生化代谢途径以及这些相互作用以何种机制调节生物系统的研究;

2) 系统的动态特征 (System Dynamics) 研究生物系统在不同条件下的行为,我们可以对系统进行代谢分析 (metabolic analysis)、敏感性分析 (sensitivity analysis)、动态分析 (dynamic analysis) 如相图 (phase portrait) 和分叉分析 (bifurcation analysis) 等等,从而发现生物系统的特定行为的机理,分叉分析又可以研究系统在多维空间中行为的时域变化 (time-varying changes), 其中每一维代表系统中一个生化成分的特定浓度;

3) 系统的控制方法 (The Control Method) 系统控制一个细胞的状态能够使得细胞功能受损最小,从而为疾病治疗提供潜在的药物靶标;

4) 系统的设计方法 (The Design Method) 将原有的生物系统修改和重建成具有所需功能的系统,其方法和策略可以从确定的设计原则、仿真模拟来发展,这样可以避免盲目的反复试验 (blind trial-and-error)。

系统生物学不仅是生命科学理论的重大发展,而且具有极其广阔的应用背景。系统生物学在人类疾病诊断与治疗、药物的发现与开发等方面将具有巨大应用潜力。系统生物学还可应用于传统的生物技术产业,开拓能源生物技术、

材料生物技术和环境生物技术等新领域。系统生物学的研究将不仅对生命科学和生物技术产生重大影响，而且对整个国民经济、社会和人类本身产生重大影响。

1.2 基因调控网络的动力学

系统生物学的研究内容之一就是研究基因调控网络 (gene regulatory network) 的动力学性质。基因调控网络是由一组或多组基因、蛋白质、小分子以及它们之间的相互调控作用所构成的一种复杂的生化系统。

最早的关于基因调控网络的模型可以追溯到分子生物学研究的早期。Jacob 和 Monod 于 1963 年发表的“乳糖操纵子的调控”开了基因调控网络研究的先声。由于细胞中的众多组件可以通过复杂的途径相互作用，所以大多数基因调控网络庞大而且复杂，如图 1.1 中所示的一个基因调控网络图^[9]。因此对于如此庞大并且复杂的调控过程的动力学性质，仅仅靠观察和描述的方法是很难理解的，因此需要引入数学工具通过对其建立模型来进行分析和模拟。随着计算机的普及，利用计算机进行分析和模拟也日益普遍。

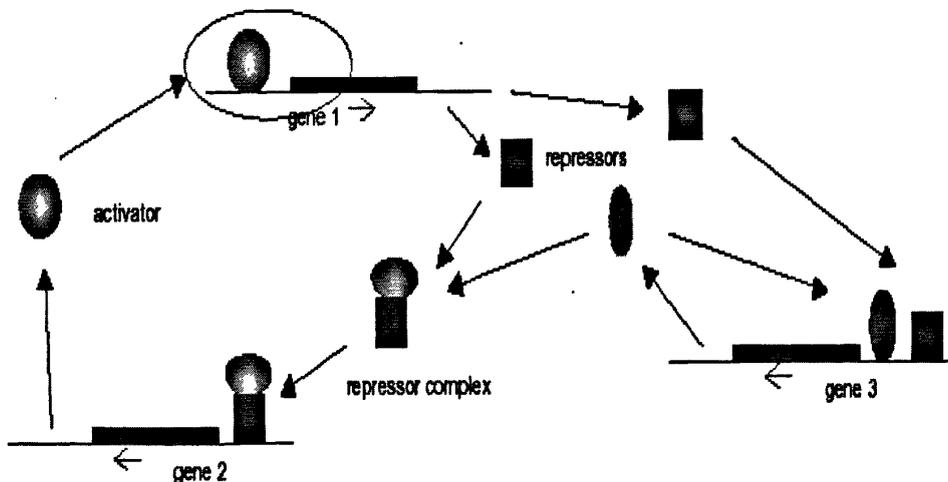


图 1.1 基因调控网络图

生命的基本单元—细胞是由许许多多不同种类的生物大分子所组成，其中

每一种生物大分子，如一种 mRNA 或一种蛋白质，在细胞内都存在着若干个相同的分子。例如，参与细胞构造的蛋白质如肌动蛋白（actins）的分子数目通常都比较大，可能达到数百万个，而负责细胞活动调控的如转录因子则分子数目一般较少，可能只有几十个。但不管是多还是少，在细胞内不存在只有单个分子的生物大分子种类。也就是说，过去人们从定性角度谈论的某某基因或某某蛋白质，实际上都是某某基因或某某蛋白质的若干个分子的集合体。

这种生物分子集合体的行为与无机世界里的分子集合体的行为有着巨大的不同，其动力学性质也有着明显的区别。例如，大气中的各种气体分子的运动，可以用玻尔兹曼方程等统计物理学的理论进行描述，而生物分子集合体则不会遵守同样的方程。我们知道，新陈代谢是生命存在的必要形式，所有的生物分子都有着一定的寿命，在生命的每时每刻，总是有一些分子在“死亡”（降解），有一些分子在“诞生”（合成）。因此，在细胞内的每一种分子集合体，都有若干个分子不停地被降解，若干个分子不停地被合成。除了分子数量的波动外，这种集合体内每个分子的活性也不会是均一的，有的高，有的低。这种在数量和活性上均具有随机波动的分子，在参与细胞的活动时会引入一种扰动，称为外在噪声（extrinsic noise）。另一方面，即使在分子的数量和活性完全一致的细胞群体中，反应的速率和进行的时间在不同的细胞中仍然存在着随机差异。这种随机性质被称为内在噪声（intrinsic noise）。现有的研究表明^[6,7]，随机噪声广泛地存在于细胞的 DNA 复制、基因转录和蛋白质合成。

噪声的存在对细胞的正常活动有重要的影响，它有不利的方面。比如会影响细胞的正常生理过程，对基因表达造成影响，产生突变，引起疾病。但它也有有利的一面。例如，生物体内许多活动都具有节律性，也就是通常所说的“生物钟”。研究人员发现，噪声在生物钟的调控方面扮演着重要的角色。此外，在真核生物中，噪声导致基因表达的不一致可以作为一种手段，用来参与细胞分化的控制。当前，对生命过程中的随机性的认识及其对生命活动影响的研究正在成为一个新的科学前沿^[7,8,9,10,11,12]。更重要的是，由于在细胞内的生命活动过程中存在着各种随机性，使得细胞的特性和行为表现出一种复杂性和多样性，这使得生命体在应付各种环境的变化上具有更强的生存机会。这一点不只是体

现在整个物种群体应对环境的变化上，对于单个细胞也是如此。

当然，在细胞内，不同的噪声源所引起的噪声对细胞的影响是不一样的。同样，细胞内也存在着不同的机制可以对噪声进行抑制、放大甚至加以利用。由于在生命过程中噪声无处不在，它不仅是生命过程的一部分，也为我们探索一些生命活动提供了一个途径。这些噪声不仅仅与参与生化反应的各种分子数目有关，也与浓度、温度等因素有关。从通讯的角度来看，这些噪声也是一种通讯手段，可以传达各种信息，不但使若干个反应之间的联系变得更为微妙，也在细胞间的通讯上起着重要的作用。因此，研究生命系统中的随机行为，对于更好地了解生命过程以及在计算机中重构细胞和生命方面有着重要的意义。

另外由于基因调控网络中生物化学反应的时间多尺度性，如 DNA 与蛋白质结合以及蛋白质聚合等快速反应，转录翻译和降解等慢速反应，以及时滞特性 (time delay) 等使得我们研究基因调控网络中的动力学性质变得尤为困难。已有一些这方面的进展^[14-19]。但这些理论都基于这样一个假设前提：所有的生物化学反应过程均为服从泊松统计的马尔科夫过程，即不考虑现实中的时滞特性。而在实际的基因调控网络中，转录和翻译等反应过程不仅相对较慢，而且常常由多个基元反应组成，这种过程服从带有时滞的高斯统计，研究这种时滞的调控网络的解析方法和数值模拟方法也就尤为重要^[21-24]。

1.3 本文的工作及安排

本文重点研究基因调控网络的随机模拟算法，给出其数学模型以及常见的数值模拟算法^[25-28]。第二章介绍连续确定性模型和离散随机性模型，给出几种精确的蒙特卡罗算法和快速近似算法。针对生化反应系统的多尺度现象，第三章介绍几种有效克服多尺度的算法，给出 Cao 等人提出的多尺度随机算法，并且在其基础上发展了改进多尺度随机算法。该算法比多尺度算法能更有效地模拟刚性系统。第四章介绍时滞基因调控网络的精确时滞随机模拟算法。针对模拟复杂的时滞系统的低效性，提出了模拟时滞生化系统的自适应 τ -Leap 算法 (DA τ -Leap)。该算法能够自动调节 τ ，使得在时间区间 $[t, t+\tau)$ 内发生多次反

应事件并且能够精确地满足 Leap 条件。数值试验证实了 $DA\tau$ -Leap 算法能够有效地模拟复杂的时滞生化网络。

论文安排如下：

本章 阐述了系统生物学的背景，简要介绍了基因调控网络的相关背景，阐述了本文的研究目的和意义，并介绍了目前国内外在这方面的研究进展。

第二章 分别介绍了基因调控网络的两种动力学模型：连续确定性模型和离散随机模型以及常用的算法。

第三章 介绍刚性生化网络的特性和一些常用克服刚性的随机算法。给出多尺度算法，并且在其基础上提出了改进多尺度算法。

第四章 介绍时滞生化反应系统的精确算法并且提出了 $DA\tau$ -Leap 算法。

第五章 对本文进行总结与展望。

第二章 基因调控网络的动力学模型及其模拟算法

从计算角度出发, 系统生物学中基因调控网络的研究大致可以分为以下两条路线: 一支是基于数据挖掘的基因调控网络的重构, 该领域主要是从大量的试验数据(如相互作用数据)中提取信息, 如从 Genbank、PDB 等数据库中采用神经网络、隐马氏模型、支持向量机等方法重构基因调控网络, 这些方法都是一种静态的研究手段, 主要研究基因调控网络的拓扑性质。而另一支研究路线则是从整体出发研究基因调控网络(如反应通道、单细胞和多细胞通信等)的动力学性质, 这是一种动态的研究方法。因此需要建立设计能够准确描述细胞内生物化学网络的动力学模型。生化网络中的化学反应呈现远离热力学平衡的现象^[25, 26], 并且分子复制数目有时候非常低, 化学反应具有明显的随机性。但是当分子数目巨大时, 随机不明显可以忽略不计, 可以用连续的确定的微分方程来描述。一般的生物化学反应系统中参加反应的分子数目通常不是很大, 而随机性则在反应中起着重要的作用, 因此可用化学主方程刻画其动力学行为。

本文主要讨论基因调控网络中的动力学性质的数值模拟。由于对基因调控网络的刻画的不同, 其数值模拟大致可以分为以下 2 种不同的方法: (1) 以连续的方式处理, 变量为反应分子的浓度, 表示为确定性方程(组), 在 2.1 小节给出; (2) 以离散的方式处理, 变量为反应分子的个数, 反应过程表示为随机过程, 这一方法在处理所含分子数目较低的系统时就显得尤为重要, 是本文研究的重点, 在剩余的小节中给出。

2.1 连续确定性动力学模型

当生物反应系统的反应分子的数目足够大, 其随机性可以忽略时, 其数目可以认为是与时间有关的连续变量。我们可以用反应物的浓度作为变量的微分方程来刻画基因调控网络的动力学性质

其中 c_k , $k=1, \dots, M$ 为随机反应速率常数, 它只与分子的物理性质和系统的反应温度以及体积有关。若记 h_k 为反应通道 R_k 中反应分子的表达式, $a_k(X(t))$ 为通道 R_k 的反应倾向函数, 则有表达式 $a_k(X(t)) = c_k h_k(X(t))$ 。此式的含义将在后面给出。反应通道 R_k 的反应类型如下表所示:

表 2.1 反应通道的类型以及 h_k 的表达式

R_k	h_k
$S_1 + S_2 \xrightarrow{c} \text{Product}$	$x_1 \cdot x_2$
$S_1 + S_2 + S_3 \xrightarrow{c} \text{Product}$	$x_1 \cdot x_2 \cdot x_3$
$S_1 \xrightarrow{c} \text{Product}$	x_1
$lS_1 \xrightarrow{c} \text{Product}$	$\begin{pmatrix} x_1 \\ l \end{pmatrix}$
$S_1 + 2S_2 \xrightarrow{c} \text{Product}$	$x_1 \cdot \frac{1}{2} \cdot x_2 \cdot (x_2 - 1) = x_1 \cdot \binom{x_2}{2}$
$\emptyset \xrightarrow{c} \text{Product}$	1

其中 x_i 为分子 S_i 的分子数目。

给定 $X(s) = x$, 在无穷小时间区间 $[s, s + ds)$ 内, 通道 R_k 发生反应的概率为 $a_k(X(s))ds + o(ds)$ 。即

$$P(RR_k(s + ds) - RR_k(s) = 1 | X(t), t \leq s) = a_k(X(s))ds + o(ds)$$

并且两个及以上的通道发生反应的概率为 $o(ds)$ 。从而有下式成立

$$P(X, t + dt) = P(X) + \sum_{j=1}^M a_j(X - v_j)P(X - v_j, t)dt - \sum_{j=1}^M a_j(X)P(X, t)dt + o(dt)$$

当 $dt \rightarrow 0$ 时, 这就变成化学主方程 (Chemical Master Equation, CME):

$$\frac{\partial P(X, t)}{\partial t} = \sum_{j=1}^M a_j(X - v_j)P(X - v_j, t) - \sum_{j=1}^M a_j(X)P(X, t).$$

CME 的直接求解非常困难, 因为方程的维数等于可能的分子状态数。于是我们通过逼近方法来研究主方程的性质。其中福克-普朗克逼近 (Fokker-Planck approximation) 是一种常用的方法。该方法是在状态 x 附近

对方程进行泰勒展开, 然后从二阶导数后截断, 从而得到不同状态之间概率流的一种连续逼近。另外 Brian 等人^[28]提出了有限状态影射算法 (Finite state projection algorithm, FSP) 对状态截断来逼近 CME。对于 CME 的维数不是很大时, FSP 方法能够提供精确的解。当 CME 的维数很大时, FSP 方法能够提供保证精度的解。虽然该算法能够有效地解一些简单的生物化学反应系统, 但是对于复杂的生物化学反应系统仍然是不可解的。许多蒙特卡罗算法被提出用来克服解 CME 的困难。下面两小节给出一些重要的算法。

2.3 精确随机模拟算法

为了模拟生物化学反应系统的轨迹, Gillespie^[29,30]提出了直接反应法 (direction reaction method, DM) 和第一反应法 (first reaction method, FRM)。在 2000 年, Gibson 等人^[31]在第一反应法的基础上给出了下一反应法 (next reaction method, NRM)。在 2007 年, David^[24]发展了改进下一反应法 (modified next reaction method, MNRM)。这些随机算法的解等价于 CME 的解, 因而是精确随机模拟算法。我们首先给出 4 种算法, 然后对 4 种算法做出点评。

对于任何精确随机模拟算法, 在每一个时间步长上, 系统只有一个状态。在给定状态 $X(t) = x$ 下, 必须通过回答下述两个问题完成反应系统的模拟: 1) 什么时候发生下一个反应? 2) 下一个反应是哪个通道? 具体地, 若 τ 与 j 分别表示下一反应发生的时间和发生反应的通道, 精确随机模拟算法通过描述下述事件 E 而完成模拟整个过程:

E: 时间区间 $[t, t+\tau)$ 内不发生任何反应, 且 R_j 将在无穷小区间 $[t+\tau, t+\tau+d\tau)$ 发生一次反应。

$P(\tau, j)d\tau$: 事件 E 发生的概率;

Gillespie 给出了概率密度 $P(\tau, j)$:

$$P(\tau, j) = a_j \exp\left(-\tau \sum_i a_i\right) \quad (2.3.1)$$

由方程 (2.3.1), 分别对 j 求和以及对 τ 积分, 可以得到下一反应 R_j 发生的概

率分布和下一反应发生的时间 τ 的概率密度:

$$P_{\tau}(\tau) = a_0(x) \exp(-a_0(x)\tau), \quad \tau > 0$$

$$P_j(j) = \frac{a_j(x)}{a_0(x)}$$

其中 $a_0(x) = \sum_{i=1}^M a_i(x)$ 。从 $U(0,1)$ 内独立产生 2 个随机数 r_1, r_2 ，根据上述概率密度函数，可以得出

$$\tau = -\ln(r_1)/a_0(x), \quad (2.3.2)$$

而 j 为满足下述不等式的最小正整数

$$\sum_{i=1}^j a_i(x) > r_2 a_0(x) \quad (2.3.3)$$

DM 算法总结如下:

步骤 1 初始化系统状态 $X(0)$ ，设定反应初始时刻 $t_s = 0$ ，结束时间 t_f ；

步骤 2 根据系统状态 $X(t)$ 计算每个反应的倾向函数 a_{μ} ($\mu=1, \dots, M$) 和 a_0 ；

步骤 3 从 $U(0,1)$ 产生随机数 r_1, r_2 ；

步骤 4 根据式 (2.3.2) 和 (2.3.3) 计算下一反应通道 μ 及其发生时间

隔 Δt ；

步骤 5 更新反应时间和系统状态: $t = t + \Delta t$, $X = X + \nu_{\mu}$ ；

步骤 6 如果 $t < t_f$ ，返回到步骤 2 循环，否则结束。

DM 算法在每一次迭代中需要两个随机数，更新 a_{μ} 的时间开销与 M 成正比，且计算 $\sum_j a_j$ 的时间也与 M 成正比。同一篇文章里，Gillespie 给出第一反应方法。

$$\tau_i = -\ln(r_1)/a_0(x), \quad (2.3.4)$$

这个式子表示只有第 i 个反应通道发生下一反应的时间,因此设 μ 是预期反应时间最短的那个反应通道,设 τ 是那个最短的预期反应时间 τ_μ ,FNM算法如下:

- 步骤 1 初始化系统状态 $X(0)$, 设定反应初始时刻 $t_s = 0$, 结束时间 t_f ;
- 步骤 2 根据系统状态 $X(t)$ 计算每个反应的倾向函数 $a_\mu (\mu=1, \dots, M)$;
- 步骤 3 从 $U(0,1)$ 产生 M 个均匀分布的随机数 r_1, \dots, r_M ;
- 步骤 4 根据 (2.3.4) 式计算每一个反应通道的预期时间 τ_i ;
- 步骤 5 查找最小的 τ_i , 设为 τ_μ ;
- 步骤 6 更新反应时间以及系统状态: $t = t + \Delta t$, $X = X + v_\mu$;
- 步骤 7 如果 $t < t_f$, 返回到步骤 2 循环, 否则结束。

可以看出,第一反应方法在每次迭代时要用 M 个随机数,此算法用于更新 a_μ 的时间开销与 M 成正比,用于寻找最小的 τ_μ 的时间开销也与 M 成正比。由于第一反应方法在每次循环中要使用 M 个随机数,而直接法每次迭代却只使用 2 个随机数,为此 Gibson 在第一反应方法的基础上给出下一反应方法。David 在下一反应法的基础上提出了改进下一反应法。改进下一反应法与下一反应法差别仅仅是在形式上的,这两种算法不仅在每次循环中只需要一个随机数,而且还可以被拓展到与时间有关的生化反应系统上,特别容易扩展到时滞生物化学反应系统,具体含义在第四章给出。为方便起见,只给出 MNRM 算法,在本节的以后讨论中,NRM 和 MNRM 同等对待。下面介绍 MNRM 算法。

若记 $RR_k(t)$ 为通道 R_k 在时间区间 $[0, t)$ 内发生的次数,系统的状态方程可写为

$$X(t) = X(0) + \sum_{k=1}^M RR_k(t)v_k \quad (2.3.5)$$

给定 $X(s) = x$, 在无穷小区间 $[s, s+ds)$ 内, 通道 R_k 发生反应的概率为 $a_k(X(s))ds + o(ds)$. 即

$$P(RR_k(s+ds) - RR_k(s) = 1 | X(t), t \leq s) = a_k(X(s))ds + o(ds)$$

并且两个通道及以上的通道发生反应的概率为 $o(ds)$ 。

设 $Y_k(\bullet)$ 为独立的单位速率泊松过程：

$$P(Y_k(T+\Delta T)-Y_k(T)=1)=\Delta T+o(\Delta T)$$

由于 $a_k(X(s))$ 在下一反应发生之前，保持不变，因此 (2.3.5) 式可以写成

$$P(Y_k(\int_0^{s+\Delta s} a_k(X(t))dt)-Y_k(\int_0^s a_k(X(t))dt)=1 | X(t), t \leq s) = a_k(X(s))\Delta s + o(\Delta s)$$

记 $RR_k = Y_k(\int_0^s a_k(X(t))dt)$ ，系统状态方程可以改写成：

$$X(t) = X(0) + \sum_{k=1}^M Y_k(\int_0^t a_k(X(s))ds) v_k \quad (2.3.6)$$

记 $T_k(t) = \int_0^t a_k(X(s))ds$ 是单位速率泊松过程 Y_k 的内部时间， t 为系统的绝对时间。

为了精确模拟系统，必须对下列两个问题做出回答：

1) 随机过程 $Y_k(\int_0^s a_k(X(t))dt)$ 下一反应的绝对时间是多少？

2) 在下一反应的绝对时间知道的情况下，哪个通道将发生反应？

在没有发生其他反应之前，通道 R_k 下一反应的时间为具有参数 $a_k(X(t))$ 的指数分布。记 $P_k = \min\{s > T_k | Y_k(s) > Y_k(T_k)\}$ 。假定系统的状态为 $X(t) = x$ 时，假设知 P_k, T_k 。通道 R_k 下一次发生的时间间隔为 $\Delta t_k = \frac{P_k - T_k}{a_k(x)}$ 。选择最小的时间间隔

$\Delta t_i = \min_{k \in M} \{\Delta t_k\}$ ，则系统下一反应的时间间隔为 Δt_i ，发生反应的通道为 R_i 。更新系统：

$t = t + \Delta t_i$ ， $T_k = T_k + a_k(x)\Delta t_i$ 和 $x = x + v_i$ 和对于通道 R_i 更新

$P_i = P_i + \ln(1/r)$ ，其中 $r \in U(0,1)$ 。重复上述过程，得到 MNRM 算法。

MNRM 算法如下：

步骤 1 初始化系统状态，初始反应时刻 $t_i = 0$ ，结束时间为 t_f ，

$$P_k = T_k = 0, k = 1, \dots, M ;$$

步骤 2 根据系统状态 $X(t)$ 计算每个反应的倾向函数 a_k ($k=1, \dots, M$)；

步骤 3 从 $U(0,1)$ 产生 M 个独立变量 r_k ;

步骤 4 对每一个 k , 设 $P_k = \ln(\frac{1}{r_k})$;

步骤 5 对每一个 k , 设 $\Delta t_k = \frac{P_k - T_k}{a_k}$;

步骤 6 设 $\Delta t_i = \min_{k \in M} \{\Delta t_k\}$, 更新系统: $t = t + \Delta t_i$, $T_k = T_k + a(x)\Delta t_i$ 和 $x = x + v_i$,

和对于通道 R_i 更新 $P_i = P_i + \ln(1/r)$, 其中 $r \in U(0,1)$;

步骤 7 更新倾向函数 a_k ($k=1, \dots, M$);

步骤 8 如果 $t < t_f$, 返回到步骤 2 循环, 否则结束。

通常情况下 NRM 和 MNRM 要比其它两种算法要快一些, 然而有些情况使用 DM 是最快的, 因为 NRM 和 MNRM 虽然每次迭代只需用一个随机数, 但是它查找最小时间, 也需要消耗一定的 CPU 时间, 具体讨论参见 [32]。

2.4 快速的随机模拟算法

上述算法为精确模拟算法, 由于精确地模拟每一个反应事件导致计算量很大, 因而对于模拟复杂的生化系统是低效的。为了加快模拟的速度, Gillespie 等人提出了 τ -Leap 算法^[33, 34]。在给定时间区间 $[t, t+\tau)$ 内系统状态改变很小时, 该算法利用泊松变量生成每一个反应通道在该区间内的反应次数。该类算法能在给定的时间区间 $[t, t+\tau)$ 上发生多次反应, 故对于分子数目很大的系统是比较有效的。对某类多通道分子反应系统, 在整个反应过程中, 其中一些反应通道发生反应的频率与另外一些反应通道发生反应的频率相比, 要大几个数量级, 并且一些分子参加反应频率比另外一些分子参加反应的频率要大得多。系统的状态在短时间内, 主要由反应频率高的反应通道刻画, 在长时间内, 主要由反应频率低的反应通道来刻画。这种现象叫做多时间尺度现象 (简称多尺度现象), 在数学上称做“刚性”。许多算法被提出用来克服系统的刚性, 将在下一章给出。这一小节, 我们介绍模拟分子数目多的系统的算法。首先我们介绍 Gillespie 的 τ -Leap 算法, 该类算法允许反应通道在给定时间区间 $[t, t+\tau)$ 内

发生多次反应，但必须满足 Leap 条件：系统的反应倾向函数几乎不变。精确的表述如下：给定时间区间 $[t, t + \tau)$ 和系统状态 $X(t) = x$ ， R_j 通道反应的次数由泊松随机变量 $K_j(\tau; x) = Poisson(a_j(x)\tau)$ 逼近。在时间区间 $[t, t + \tau)$ 内的系统状态改变量为：

$$X(t + \tau) - x = \wedge(\tau; x) = \sum_{j=1}^M K_j(\tau; x) v_j \quad (2.4.1)$$

满足 Leap 条件为：

$$|a_j(x + \wedge(\tau; x)) - a_j(x)| \leq \varepsilon a_0(x), (j=1, \dots, M) \quad (2.4.2)$$

其中 ε 为预设误差参数 ($0 < \varepsilon \ll 1$)。Gillespie 等人^[4]给出了满足 (2.4.2) 式的最大 τ 值：

$$\tau = \min_{j \in M} \left\{ \frac{\varepsilon a_0(x)}{|\mu_j(x)|}, \frac{\varepsilon^2 a_0^2(x)}{|\sigma_j^2(x)|} \right\} \quad (2.4.3)$$

其中

$$\mu_j(x) = \sum_{j'=1}^M f_{j'}(x) a_{j'}(x) \quad (2.4.5)$$

$$\sigma_j^2(x) = \sum_{j'=1}^M f_{j'}^2(x) a_{j'}(x)$$

$$f_{j'}(x) = \sum_{i=1}^N \frac{\partial a_j(x)}{\partial x_i} v_{j'}^i, j, j' = 1, \dots, M \quad (2.4.6)$$

利用上述选取时间步长 τ 的方法，我们可以得到 τ -Leap 算法

步骤 1 初始化系统状态，初始反应时刻 $t_s = 0$ ，结束时间为 t_f

步骤 2 在时刻 t 系统状态为 X ，计算所有反应的倾向函数以及它们的和

$$a_0(x) = \sum_{j=1}^M a_j;$$

步骤 3 根据式 (2.4.3) 计算 τ ；

步骤 4 如果步骤 2 中得到的时间步长 τ 小于 $1/a_0$ 和某个数（不妨假设是 10 的乘积，则放弃这个时间步长，转而接连执行一定个数（不妨设为 100）的精确随机模拟 (SSA) 过程，然后再尝试获取新的时间步长 τ 。

反之，如果步骤 2 中得到的时间步长 τ 大于 $1/a_0$ 和那个数的乘积，则接受这个时间步长 τ ，向下进行到步骤 4；

步骤 5 对于每个反应 j ($j=1, \dots, M$)，产生一个均值为 $a_j(x)\tau$ 的 Poisson 随机变量 K_j ，代表反应 j 在时间步长 τ 内发生的次数；

步骤 6 更新 $t=t+\tau$ ， $x=x+\sum_{j=1}^M K_j v_j$ ；

步骤 7 如果 $t < t_f$ ，返回到步骤 2 循环，否则结束。

当 M, N 很大时，式 (2.4.5) 和 (2.4.6) 的计算量为 $O(M^2N+2M^2)$ ，为了有效地降低其计算量，Cao 等人^[37]提出了对于 τ -Leap 算法的有效的步长选择算法。该算法是基于反应分子的相对变化的。对于上述的 ε 有

$$\tau = \min_{i \in I_{\varepsilon}} \left\{ \frac{\max\{\varepsilon X_i(t)/g_i, 1\}}{|\bar{\mu}_i(X(t))|}, \frac{(\max\{\varepsilon X_i(t)/g_i, 1\})^2}{\bar{\sigma}_i^2(X(t))} \right\} \quad (2.4.7)$$

其中

$$\begin{aligned} \bar{\mu}_i(X(t)) &= \sum_{j=1}^M a_j(X(t))v_{ij}, \quad i \in I_{\varepsilon} \\ \bar{\sigma}_i^2(X(t)) &= \sum_{j=1}^M a_j(X(t))v_{ij}^2, \quad i \in I_{\varepsilon} \end{aligned} \quad (2.4.8)$$

I_{ε} 为反应分子集合，详细请参考文献[37]，(2.4.8)式的计算量为 $2M^2$ 。故当 N 很大时，能够有效地降低计算量，提高速度。

上述的算法都是假定在给定时间区间 $[t, t+\tau)$ 内系统状态改变很小时，该算法利用泊松随机数确定每一个反应通道在该区间内的反应次数。由于泊松数的无界性，有可能产生的反应次数导致某些分子变成负的，但是负的分子数不符合现实。为了避免负分子数的出现，Tian 等人^[7]和 Chatterjee 等人^[8]独立地提出了二项 τ -Leap 算法。因为二项分布产生的随机数是有界的，因此二项 τ -Leap 算法分布可以避免负分子数目的产生。Cai 等人^[39]提出的 K-Leap 算法能够减少产生负分子数目的概率。Cao 等人^[38]发展了一种避免负分子数产生的算法。该算法将反应通道分成容易产生负分子数的关键通道和不容易产生负分子数的非关键通道。用标准的 Gillespie 随机算法模拟关键通道，用 τ -Leap 算法模拟非关键通道。这样大大地降低了产生负分子数的概率，但该类算法在确定了时间步长 τ 后，就不能再进行调整了，属于先验 τ -Leap 算法。最近 David

提出了将后验的 Leap 检查点植入 τ -Leap 算法^[40], 称之为后验 τ -Leap 算法。该算法利用单位速率泊松过程可以自动地调节时间步长 τ , 使得精确地满足 Leap 条件, 从而产生负分子数的概率为 0, 从而可以保证精度。下面我们介绍后验 τ -Leap 算法。

单位速率泊松过程 Y_k 具有以下两种性质:

- a) 对于 $T_1 \geq T_2 \geq T_k(t)$, 增量 $Y_k(T_2) - Y_k(T_1) = Y_k(T_2 - T_1)$ 是与系统状态 $X(t)$ 是独立的参数为 $T_2 - T_1$ 的泊松分布;
- b) 给定 $0 \leq s < u < t$, $Y_k(s), Y_k(t)$ 。 $Y_k(u) - Y_k(s)$ 为二项分布 $(Y_k(t) - Y_k(s), r)$, 其中 $r = \frac{u-s}{t-s}$ 。

在先验 τ -Leap 算法里, 只是对 a_k 进行逼近, 而在后验 τ -Leap 里, 对通道的反应次数也进行逼近。利用单位速率泊松随机过程的性质 a, b, 提出了后验 τ -Leap 算法。算法归纳如下:

给定时刻 t 、系统状态 $X(t) = x$ 、每一个通道 R_k 的倾向函数 $a_k(X(t))$ 、内部时间 $T_k = T_k(t) = \int_0^t a_k(X(s))ds$ 以及到绝对时间为 t 时发生次数 $C_k = Y_k(T_k)$ 。假定对每一个通道 R_k 已经存储 $T_1^k, \dots, T_{end}^k, Y_k(T_1^k), \dots, Y_k(T_{end}^k)$ 并且具有 $T_{end}^k > \dots > T_1^k > T_0^k = T_k$ 。对于步长 τ , 通道 R_k 的内部时间为 $T_k + a_k(X(t))\tau$ 时, 发生的反应次数 Y_k 概率分布分两种情况:

如果存在 $i \in \{0, \dots, end - 1\}$ 使得 $T_i^k \leq T_k + a_k(X(t))\tau < T_{i+1}^k$ 成立, 根据性质 b 知

$$Y_k(T_k + a_k(X(t))\tau) = Binomial(Y_k(T_{i+1}^k) - Y_k(T_i^k), r) + Y_k(T_i^k)$$

其中

$$r = \frac{T_k + a_k(X(t))\tau - T_i^k}{T_{i+1}^k - T_i^k}.$$

如果 $T_k + a_k(X(t))\tau \geq T_{end}^k$

$$Y_k(T_k + a_k(X(t))\tau) = Poisson(T_k + a_k(X(t))\tau - T_{end}^k) + Y_k(T_{end}^k)$$

则在时间区间 $[t, t + \tau)$ 发生的反应次数 $N_k = Y_k(T_k + a_k(X(t))\tau) - C_k$ 代入状态方程 (2.4.1) 得到状态 $X(t + \tau)$, 检查是否满足 Leap 条件。如满足则接受 τ , 更新系统。如不满足, 则将 $Y_k(T_k + a_k(X(t))\tau)$, $T_k + a_k(X(t))\tau$ 按升序插入到已存储的

信息里。用更小的 τ 重复上述过程，直到找到符合条件的 τ 。

后验 τ -*leap*算法：

步骤1 初始化：初始时间及初始分子数目 $t=0, X(t)=X$ ，结束时间 t_f ，对每一个 k ， $T_k=C_k=0$ ， $Q_k=[0,0]$ ，计算 a_k ，设 $0 < q < 1$ ；

步骤2 while $t \leq t_f$

(a) 根据 τ -*leap*算法，计算 τ ；

(b) 对每一个 k ，做下面的计算

(a1) 设 B_k 为 Q_k 的行数；

(a2) 如果 $T_k + a_k \tau \geq Q_k(B_k, 1)$

$$N_k = \text{Poisson}(T_k + a_k \tau - Q_k(B_k, 1)) + Q_k(B_k, 2) - C_k$$

$$row_k = B_k;$$

(a3) 否则

找到指标 i ，满足不等式 $Q_k(i-1, 1) \leq T_k + F_k < Q_k(i, 1)$

$$r = \frac{T_k + F_k - Q_k(i-1, 1)}{Q_k(i, 1) - Q_k(i-1, 1)}$$

$$N_k = \text{Binomial}(Q_k(i, 2) - Q_k(i-1, 2), r) + Q_k(i-1, 2) - C_k$$

$$row_k = i - 1;$$

步骤3 根据得到的 N_k ，带入状态方程检查 Leap 条件；

步骤4 如不符合 Leap 条件

(a) 更新 Q_k

将 $[T_k + a_k \tau, C_k + N_k]$ 插入到 row_k 和 $row_k + 1$ 之间，如果

$row_k + 1 > B_k$ ，增加新一行到 Q_k 的最后；

(b) 令 $\tau = q\tau$ ，回到2(b)；

步骤 5 如符合 Leap 条件

(a) 更新每一个 Q_k

删除小于等于 row_k 的行, 增加新的一行 $[T_k + F_k, C_k + N_k]$ 作为 Q_k 的第一行;

(b) 设 $t = t + \tau$, $X = X + \sum_{k=1}^M N_k v_k$;

(c) 更新每一个 $T_k = T_k + a_k \tau$ 和 $C_k = C_k + N_k$;

(d) 计算 a_k , 回到 2(a)。

第三章 刚性生化系统的随机模拟算法

所谓的刚性生物化学反应系统，是指在整个反应过程中，其中一些反应通道发生反应的频率与另外一些反应通道发生反应的频率相比，要大几个数量级，并且一些分子参加反应频率比另外一些分子参加反应的频率要大得多。系统的状态在短时间内，主要由反应频率高的反应通道刻画，在长时间内，主要由反应频率低的反应通道来刻画。近年来，许多算法用来解决刚性问题。隐 τ -Leap 算法^[7]解决了由分子数目引起刚性的系统。但该算法在时间步长较大时，不能保证精度。Rao 等人^[16]和 Ethan 等人^[11]把反应分子分成两种：中间分子（intermediate）和慢分子。在拟稳态(quasi-steady-state, QSS)假设下把中间分子从 CME 中消除，得到只含有慢分子状态的 CME。该算法的缺点是在现实的反应系统里，难以确定中间分子和慢分子。Hasetltine 等人^[17]把反应通道分成两种：快反应通道和慢反应通道，用确定性方程或郎之万方程模拟快通道，SSA 模拟慢通道。但该算法难以推广到复杂的非线性反应系统中。Cao 等人提出了多尺度算法（MSSA）^[19]。该算法把反应通道分成快反应通道和慢反应通道，把参加任一快反应通道的分子叫做快分子，其它的叫慢分子。快反应通道及其快分子组成快—子系统，而慢反应通道及参加其反应的分子组成慢—子系统。MSSA 在随机的部分平衡假设下(stochastic partial equilibrium assumption, SPEA)，用快—子系统状态的稳定概率分布或者用其一阶两阶矩逼近慢—子系统的倾向函数，然后用 SSA 模拟慢—子系统的系统状态。该算法既有坚实的理论基础也便于应用到复杂的分子反应系统中，是模拟刚性系统的有效算法之一。

当系统的慢—子系统中分子数目较多时，在给定的时间区间 $[t, t+\tau]$ 内慢反应通道能发生多次反应，而系统的慢倾向函数改变很小，MSSA 由于在慢—子系统上模拟每一个反应事件从而不能有效地模拟整个系统。为了提高模拟速度，我们基于 MSSA 算法和 τ -Leap 算法提出了改进多尺度算法（Improved Multiscale stochastic simulation with τ -Leap, IMSS τ ）。这种算法充分考虑了快—子系统和慢—子系统的相互作用，用 τ -Leap 算法模拟慢—子系统，

从而在牺牲一定精度的条件下，达到了提高模拟速度的目的。

本章首先介绍 MSSA 算法，然后给出 IMSS τ 算法，最后给出两者的数值比较。

3.1 MSSA 算法

考虑具有刚性的生化系统。反应通道集合 R 可以根据反应通道的倾向函数的大小分成快反应通道集合 $R^f = \{R_1^f, \dots, R_{M^f}^f\}$ 和慢反应通道集合 $R^s = \{R_1^s, \dots, R_{M^s}^s\}$ ，其中 $M^f + M^s = M$ 。把参加快反应通道的分子定义为快分子 S^f ，其余的分子为慢分子 S^s 。从而分子 S 可以分成快分子集合 $S^f = \{S_1^f, \dots, S_{N^f}^f\}$ 和慢分子集合 $S^s = \{S_1^s, \dots, S_{N^s}^s\}$ ，其中 $N^f + N^s = N$ 。快—子系统是快分子和快分子反应通道组成的，而慢—子系统由慢反应通道和其涉及的分子组成的。特别注意快分子的数目也可以被慢反应通道改变，但慢分子不能被任何快反应通道改变其数量。同样对状态向量分成 $X(t) = (X^f(t), X^s(t))$ ，简写成 $x = (x^f, x^s)$ 。对应的快倾向函数和慢倾向函数分别记为：

$$\begin{aligned} a_j^f(x) &= a_j^f(x^f, x^s), \quad j=1, \dots, M^f \\ a_j^s(x) &= a_j^s(x^f, x^s), \quad j=1, \dots, M^s \end{aligned}$$

对应快状态和慢状态改变向量为：

$$\begin{aligned} v_j^f &= (v_{1j}^f, \dots, v_{N^f j}^f), \quad j=1, \dots, M^f \\ v_j^s &= (v_{1j}^s, \dots, v_{N^f j}^s, v_{1j}^{ss}, \dots, v_{N^s j}^{ss}), \quad j=1, \dots, M^s \end{aligned}$$

由定义知 $v_y^{sf} = 0$ ，我们把 0 元素直接从 v^f 中去掉。设慢—子系统上的时间刻度为 t_s ，在 SPEA 下，快分子的状态的极限概率分布是平衡分布，

$$\lim_{t \rightarrow \infty} P(X^f(t) | x, t_s) = P(x^f | x, t_s)$$

快分子达到平衡状态的时间 τ_{eqm} 与 t_s 相比，前者是可以忽略的。记 x^f 为慢—子系统在时间刻度 t_s 处于平衡时的快分子状态值。在时间刻度 t_s 时，受快—子系统影响后的慢—子系统的倾向函数 $\bar{a}_j^s(x) = \sum_{x^f} a_j^s(x^f, x^s) P(x^f | x, t_s)$ 。通常有如下形式：

$$\bar{a}_j^s(x) = \begin{cases} a_j^s(x^s), & \text{if } a_j^s(x) = a_j^s(x^s) \\ c_j^s \langle x_i^{f'} \rangle, & \text{if } a_j^s(x) = a_j^s x_i^f \\ c_j^s x_i^s \langle x_i^{f'} \rangle, & \text{if } a_j^s(x) = c_j^s x_i^s x_i^f \\ \frac{c_j^s \langle x_i^{f'} (x_i^{f'} - 1) \rangle}{2} & \text{if } a_j^s(x) = \frac{c_j^s x_i^{f'} (x_i^{f'} - 1)}{2} \\ c_j^s \langle x_i^{f'} x_i^{f'} \rangle & \text{if } a_j^s = c_j^s x_i^f x_i^f \end{cases}$$

其中 $\langle g(x^{f'}) \rangle$ 定义如下:

$$\langle g(x^{f'}) \rangle = \sum_{x^{f'}} g(x^{f'}) P(x^{f'} | x)$$

MSSA 算法如下:

给定开始时间 t_0 , 开始状态 x_0 和结束时间 t_f .

步骤 1 计算快—子系统的部分平稳状态, 更新快分子变量 $x^f = x^{f'}$;

步骤 2 对每一个 $j=1, \dots, M^s$, 计算 $\bar{a}_j^s(x)$ 和 $\bar{a}_0^s(x) = \sum \bar{a}_j^s(x)$;

步骤 3 从 $U(0,1)$ 中产生两个随机数 r_1, r_2 . 下一次发生反应的时刻为 $t+\tau$, 其中 τ 由下式给定

$$\tau = \frac{1}{\bar{a}_0^s(x)} \ln\left(\frac{1}{r_1}\right)$$

下一个慢反应通道的指标 j 由满足下式的最小正整数决定

$$\sum_{j=1}^J \bar{a}_j^s(x) > r_2 \bar{a}_0^s(x);$$

步骤 4 如果 $t+\tau > t_f$, 结束. 否则, 更新时间 $t = t+\tau$ 和状态 $x = x+v_j$, 返回步骤 1.

当快分子数目较大时, 可以把快分子的高阶矩近似成一阶矩的函数. 用

$$\bar{a}_j^s(x) = \frac{c_j^s}{2} (\langle x_i^{f'} \rangle^2 - \langle x_i^{f'} \rangle), \bar{a}_j^s(x) = c_j^s \langle x_i^{f'} \rangle \langle x_i^{f'} \rangle$$

代替

$$\bar{a}_j^s(x) = \frac{c_j^s}{2} \langle x_i^{f'}(x_i^{f'} - 1) \rangle, \bar{a}_j^s(x) = c_j^s \langle x_i^{f'} x_i^{f'} \rangle$$

设 $\langle x_i^{f'} \rangle = h_i(x^f)$ ，一阶矩 $h_i(x^f)$ 可以用 SPEA 和分子守恒律得出，3.1 小节给出具体讨论。另外，由于快分子数目较大而慢一子系统一次反应至多改变 3 个快分子，模拟的时候可以省去步骤 1。如果想得到快分子的状态，可以在快一子系统上用 SSA 模拟得到，详见参考文献[18, 19]。

3.2 IMSS τ 算法

IMSS τ 算法核心是找到满足慢一子系统上 Leap 条件的 τ' 最大值的表达式。

满足 Leap 条件

$$|\bar{a}_j^s(h(x^f + \wedge^f), x^s + \wedge^s) - \bar{a}_j^s(h(x^f), x^s)| \leq \varepsilon \bar{a}_0^s(h(x^f), x^s), \quad j=1 \cdots M^s \quad (3.2.1)$$

其中

$$\bar{a}_0^s(h(x^f), x^s) = \sum_{j=1}^{M^s} \bar{a}_j^s(h(x^f), x^s)$$

$$\wedge_i^s = \sum_{j=1}^{M^s} P(\bar{a}_j^s(h(x^f), x^s), \tau) \nu_{ij}^{ss}, \quad i=1, \dots, N^s \quad (3.2.2)$$

$$\wedge_i^f = \sum_{j=1}^{M^s} P(\bar{a}_j^s(h(x^f), x^s), \tau) \nu_{ij}^{ff}, \quad i=1, \dots, N^f \quad (3.2.3)$$

如果下面的函数知道，那么可以找到满足 (3.2.1) 的最大 τ' ：

$$\bar{\mu}_j(x) = \sum_{j'=1}^M \bar{f}_{j'}(x) \bar{a}_{j'}^s(x)$$

$$\bar{\sigma}_j^2(x) = \sum_{j'=1}^M \bar{f}_{j'}^2(x) \bar{a}_{j'}^s(x)$$

$$\bar{f}_{j'}(x) = \sum_{i=1}^{N^f} \frac{\partial \bar{a}_j^s(h(x^f), x^s)}{\partial x_i^f} \nu_{ij'}^{ff} + \sum_{i=1}^{N^s} \frac{\partial \bar{a}_j^s(h(x^f), x^s)}{\partial x_i^s} \nu_{ij'}^{ss}, \quad j, j'=1, \dots, M^s \quad (3.2.4)$$

从而得到

$$\tau' = \min_{j \in M^s} \left\{ \frac{\varepsilon \bar{a}_0^s(h(x^f), x^s)}{\bar{\mu}_j(x)}, \frac{\varepsilon^2 \bar{a}_0^s{}^2(x)}{\bar{\sigma}_j^2(x)} \right\} \quad (3.2.5)$$

计算 τ' 的关键就是怎样计算 (3.2.4) 式右边的第一项, 我们在 3.3 小节给出答案。我们先给出 IMSS τ 算法如下:

给定开始时间 t_0 , 开始状态 x_0 和结束时间 t_f ;

步骤 1 计算快一子系统的部分平稳状态, 更新快分子变量 $x^f = x^f$;

步骤 2 对每一个 $j=1, \dots, M^f$, 计算 $\bar{a}_j^f(x)$ 和 $\bar{a}_0^f(x) = \sum_j \bar{a}_j^f(x)$;

步骤 3 根据 (3.2.5) 计算步长 τ' , 然后根据方程 (3.2.3) 和 (3.2.4) 产生两个随机数 \wedge^s, \wedge^f ;

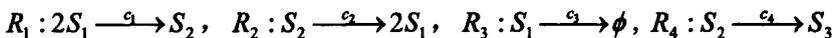
步骤 4 如果 $t + \tau' > t_f$, 结束。否则, 更新时间 $t = t + \tau'$ 、状态 $x = (x^f + \wedge^f, x^s + \wedge^s)$ 及 $h(x^f)$, 返回步骤 1。

当快分子数目较多时, 我们可以省去步骤 1。如果需要快分子状态, 可以在快一子系统上用 SSA 模拟。

3.3 IMSS τ 算法的实现

3.3.1 计算 $h_i(x^f)$

不失一般性, 考虑一个降解聚合系统来阐明 $h_i(x^f)$ 的计算过程。该模型有 3 种分子, 4 个反应通道。模型如下:



在这些反应中, 分子 S_1 以速率 c_3 降解, 以速率 c_1 二聚成不稳定的分子 S_2 , 而分子 S_2 以速率 c_2 分解成分子 S_1 以速率 c_4 转化成稳定分子 S_3 。该模型作为刚性的模型已在 [16, 33, 34] 中研究过。现在我们从多尺度角度来研究。反应速率分别取值为 $c_1 = 200$, $c_2 = 1000$, $c_3 = 1$, $c_4 = 0.1$ 。初始状态为 $X(0) = (4000, 7980, 0)$ 。由不等式 $c_1, c_2 \gg c_3, c_4$ 知, 快一子系统由通道 R_1, R_2 及分子 S_1, S_2 组成。

根据 SPEA 知

$$c_1(h_1^2(x_1, x_2) - h_1(x_1, x_2))/2 = c_2 h_2(x_1, x_2) \quad (3.3.1.1)$$

根据分子守恒律知

$$h_1(x_1, x_2) + 2h_2(x_1, x_2) = x_1 + 2x_2 \quad (3.3.1.2)$$

我们有两种方法可从上述方程解出 $h_1(x_1, x_2), h_2(x_1, x_2)$ 。一种是用非线性

方法例如牛顿迭代法得到数值解。另一种是得到显式表达式解。因为守恒律产生线性方程，而 SPEA 通常产生一次和两次方程，最多是三次方程，这些方程都能用符号计算软件例如 MAPLE, MATLAB 得到显式解

$$h_1(x_1, x_2) = \frac{1}{2} - \frac{c_2}{2c_1} + \frac{1}{2} \sqrt{1 - \frac{2c_2}{c_1} + \left(\frac{c_2}{c_1}\right)^2 + 8\frac{c_2}{c_1}x_1 + 4\frac{c_2}{c_1}x_2}$$

$$h_2(x_1, x_2) = -\frac{1}{4} + \frac{c_2}{4c_1} - \frac{1}{4} \sqrt{1 - \frac{2c_2}{c_1} + \left(\frac{c_2}{c_1}\right)^2 + 8\frac{c_2}{c_1}x_1 + 4\frac{c_2}{c_1}x_2} + \frac{1}{2}x_1 + x_2$$

3.3.2 计算 $\frac{\partial \bar{a}_j^s(h(x^f), x^s)}{\partial x_i^f}$

根据微分链式法则，可得

$$\frac{\partial \bar{a}_j^s(h(x^f), x^s)}{\partial x_i^f} = \sum_{k=1}^{M^f} \frac{\partial \bar{a}_j^s(h(x^f), x^s)}{\partial h_k(x^f)} \frac{\partial h_k(x^f)}{\partial x_i^f}, i=1, \dots, N^f, j=1, \dots, M^s \quad (3.3.2.1)$$

由倾向函数 $\bar{a}_j^s(h(x^f), x^s)$ ，可以计算 $\frac{\partial \bar{a}_j^s(h(x^f), x^s)}{\partial h_k(x^f)}$ 。因此式 (3.3.2.1) 的关键是计算 $\frac{\partial h_k(x^f)}{\partial x_i^f}$ 。

如果 $h_k(x^f)$ 是由显式表达式得到的解，那么可以直接对 $h_k(x^f)$ 求偏导，可以得到 $\frac{\partial h_k(x^f)}{\partial x_i^f}$ 的解。如果 $h_k(x^f)$ 是由非线性数值方法得到的解，可以用隐式微分法则，得到含 $\frac{\partial h_k(x^f)}{\partial x_i^f}$ 的线性方程组。不失一般性，我们以求 $\frac{\partial h_1(x_1, x_2)}{\partial x_2}$, $\frac{\partial h_2(x_1, x_2)}{\partial x_2}$ 为例来说明计算过程，对方程 (3.3.1.1) 和

(3.3.1.2) 的 x_2 进行求导

$$h_1(x_1, x_2) \frac{\partial h_1(x_1, x_2)}{\partial x_2} - \frac{1}{2} \frac{\partial h_1(x_1, x_2)}{\partial x_2} - \frac{c_2}{c_1} \frac{\partial h_2(x_1, x_2)}{\partial x_1} = 0$$

$$\frac{\partial h_1(x_1, x_2)}{\partial x_2} + 2 \frac{\partial h_2(x_1, x_2)}{\partial x_2} = 2$$

可以解上述线性方程组得到 $\frac{\partial h_1(x_1, x_2)}{\partial x_2}$, $\frac{\partial h_2(x_1, x_2)}{\partial x_2}$ 数值解。

3.4 数值例子

我们通过 2 个具体的数值实例来验证 IMSS τ 算法的性能。在模拟计算中, 分别运行 MSSA 和 IMSS τ 相同的次数, 从而用不同算法得到慢分子的频率分布, 并以此近似为慢分子的概率分布。由于系统的慢反应通道的倾向函数与快反应通道的倾向函数相比, 两者至少相差两个数量级。我们可以认为 MSSA 算法得到概率分布为精确分布。我们通过 IMSS τ 和 MSSA 的频率直方图作比较, 可以给出直觉的印象。均值、方差可以衡量算法的精度。另外慢分子的轨迹图, 也可作为衡量算法精度的方法之一。为了考察算法的模拟速度, 我们考虑两种算法所用 CPU 的时间。这里所有的模拟工作均是用 MATLAB6.0 实现的, 其运行环境为 WinXP 系统, 1.90 CPU 与 256Mb 内存。

3.4.1 降解聚合系统

该模型及其参数值已在第 3.3 节给出, 慢一子系统由通道 R_1, R_2 以及慢分子 S_3 和快分子 S_1, S_2 组成。我们分别运行 IMSS τ ($\varepsilon = 0.01$) 和 MSSA 10000 次, 得到 S_3 的直方图 (见图 3.1), 从图中看出 IMSS τ (“o” 表示) 与 MSSA (“*” 表示) 的直方图, 前者的高频率的部分比后者稍微向左平移了一些, 低频率部分几乎重合。两种算法得到的 S_3 的轨迹图 (见图 3.2) 几乎重合。在表 3.1, 我们分别给出了 S_3 均值、方差和所用 CPU 时间。可以看出 IMSS τ 模拟得到均值、方差分别为 5771.2、2183.7, MSSA 模拟得到的 5764.8, 2252.5, 相对误差分别为 0.11% 和 3.05%。IMSS τ 和 MSSA 所用 CPU 的时间分别为 2871 (s) 和 65246 (s), 前者比后者快 22 倍。

表 3.1 降解合成系统中 IMSS τ 与 MSSA 的性能比较

	均值	方差	CPU 时间 (s)
MSSA ($X_3(10)$)	5764.8	2252.5	65246
IMSS τ ($X_3(10)$)	5771.2	2183.7	2871

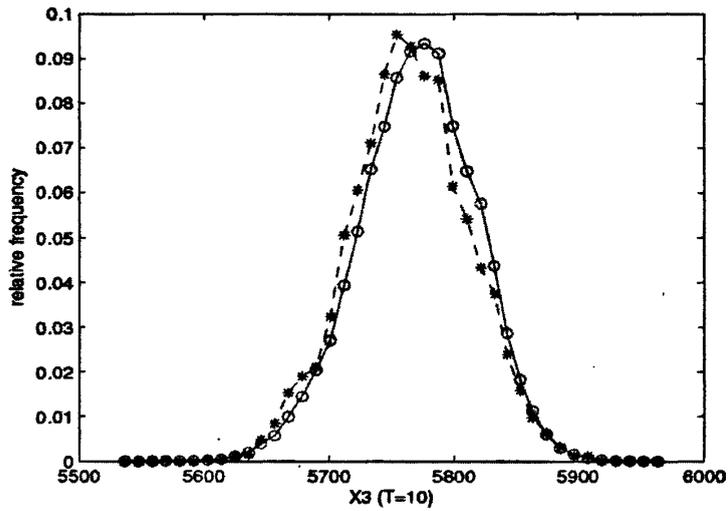


图 3.1 IMSS τ (“o”表示)和 MSSA (“*”表示)得到的降解-合成模型中的 S_3 的直方图。

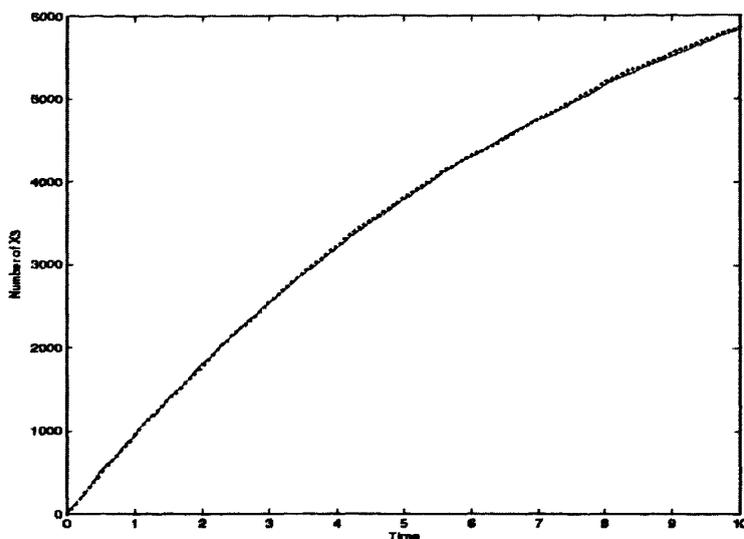
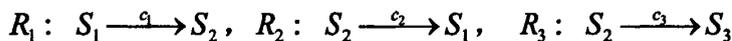


图 3.2 IMSS τ (“.”表示) 和 MSSA (实线表示) 得到的降解-合成模型中的 S_3 的轨迹图。

3.4.2 线性聚合反应系统

该系统具有 3 个反应通道



单体分子 S_1 以速率 c_1 转化成不稳定分子 S_2 , 不稳定分子 S_2 以速率 c_2 转化成 S_1 , 以速率 c_3 转化成稳定分子 S_3 。在模拟中, 我们取初始分子数目为 $X_1(0)=5000, X_2(0)=5000, X_3(0)=0$; 反应速率常数分别为 $c_1=1000, c_2=200, c_3=0.1$ 。结束时间 $T=10$; 由于 $c_1, c_2 \gg c_3$, 我们可以认为 R_1, R_2 及 S_1, S_2 组成快—子系统。慢—子系统由通道 R_3 以及慢分子 S_3 和快分子 S_2 组成。我们分别运行 IMSS τ 和 MSSA 10000 次, 得到各自 S_3 的直方图 (见图 3.3), 从图中可以看出 IMSS τ (“o”表示) 与 MSSA (“*”表示) 的直方图没什么差异。两种算法得到的 S_3 的轨迹图 (见图 3.4) 也几乎重合。在表 2, 我们分别给出了 S_3 均值、方差和所用 CPU 时间。可以看出 IMSS τ 模拟得到均值、方差分别为 5657.4、2575.5, MSSA 模拟得到的 5654.5, 2511.5, 相对误差分别为 0.51% 和 2.57%。IMSS τ 和 MSSA 所用 CPU 的时间分别为 1577(s) 和 33241(s), 前者比后者快 21 倍。

表 3.2 线性合成模型中 IMSS τ 与 MSSA 的性能比较

	均值	方差	CPU 时间 (s)
MSSA ($X_3(10)$)	5654.5	2511.5	33241
IMSS τ ($X_3(10)$)	5657.4	2575.5	1577

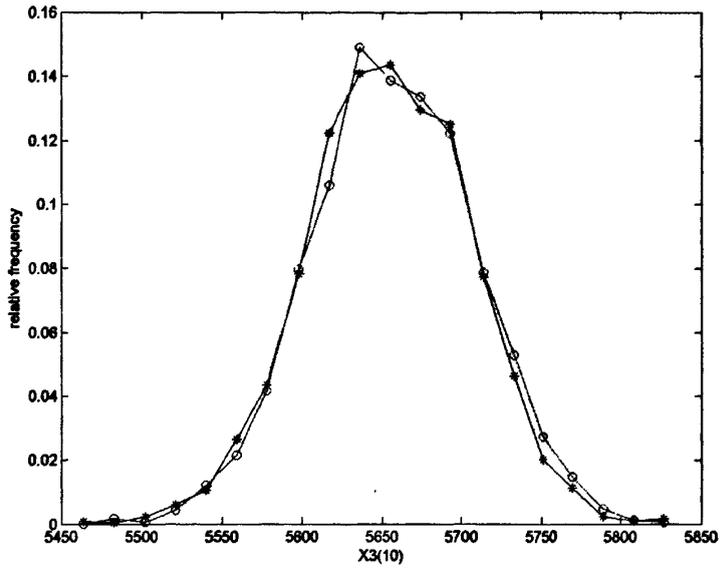


图 3.3 IMSS τ (“o” 表示) 和 MSSA (“*” 表示) 得到的线性合成模型中的 S_3 的直方图。

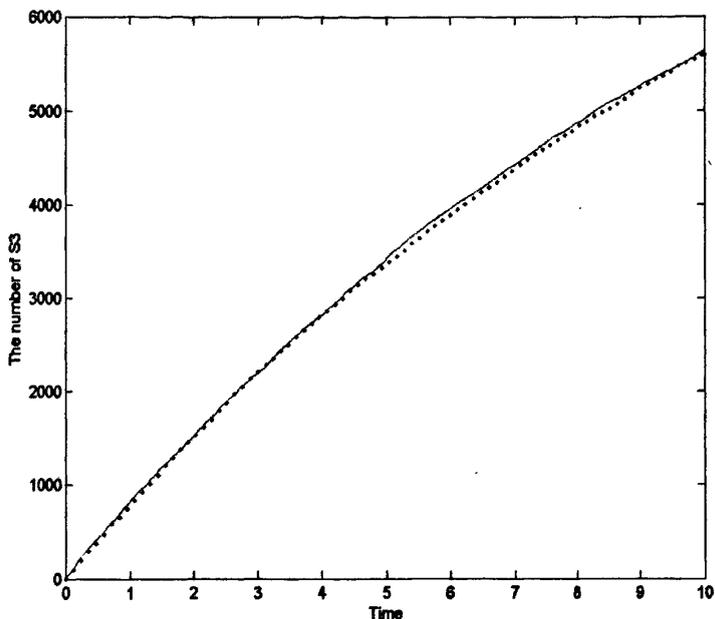


图 3.4 IMSS τ (“.”表示)和 MSA (实线表示)得到的线性合成系统中的 S_3 的轨迹图。

3.5 结语

Cao 等人提出的 MSA 算法即具有坚实的理论基础,又便于应用,是模拟复杂刚性的非线性化学反应系统的有效算法之一。但是当慢一子系统的分子数目较大时,该算法由于模拟慢一子系统的每个事件,因而导致该算法效率较低。本文在 MSA 的基础上发展了一种快速近似算法 IMSS τ 。因充分研究了快一子系统和慢一子系统的相互作用,故对于慢一子系统的分子数目较大的刚性系统,IMSS τ 比 MSA 更能有效地模拟。数值实验表明 IMSS τ 算法能够在损失较小的精度下有效地提高了模拟速度。

第四章 时滞生化系统的随机模拟算法

上述的精确算法及其加速算法均是针对不具有时滞的化学反应系统设计的。然而，现实中存在许多反应系统，其反应具有明显的时滞。目前有三种精确的模拟该类系统的算法。Barrio 等人提出的拒绝算法^[22]。该算法虽然计算下一反应的时间简单，但不能有效利用随机数。Cai 提出的时滞化学反应系统的直接法^[23]虽然能够有效地使用随机数，但是计算下一发生反应事件的时间比较复杂。David 提出模拟时滞反应系统的改进下一反应法^[24] (DMNRM) 即有效利用了随机数又使得计算下一反应的时间简单。与无时滞的反应系统一样，该算法模拟每一个反应事件从而导致了模拟复杂的时滞反应系统时速度低下。

本章首先介绍了时滞生化反应系统的改进下一反应法接着提出了一个模拟时滞化学反应系统的自适应 τ -Leap 算法 (DA τ -Leap)，最后对两种算法做了比较。

4.1 时滞生化反应系统的精确随机模拟算法

考虑反应通道中存在部分反应通道具有时滞性，用 R_D 表示具有时滞的反应通道子集，对于 $R_i \in R_D$ ，其时滞为 d_i ，即通道 R_i 发生反应时，其反应产物在 d_i 时间后才能生成。通常 d_i 既可以是一个确定的数，也可以是随机数。在本文中假设 d_i 是确定的数。在时滞化学反应系统中，存在三类反应通道：没有时滞的反应通道集合记为 R_{ND} ，未完成反应的反应物可以参加到其他新反应中的时滞反应通道集合记为 R_{ICD} ，未完成反应的反应物不参加到其他新反应中的时滞反应通道集合记为 R_{CD} ，即 $R = R_{ND} \cup R_D = R_{ND} \cup R_{CD} \cup R_{ICD}$ 。当属于 R_{ICD} 的反应通道发生反应时，反应物的分子数目立即更新，而属于 R_{CD} 的反应通道发生反应时，反应物的分子数目在其时滞时间之后才更新。对于时滞化学反应系统的反

应通道 R_k 在无穷小区间 $[t, t + \Delta t)$ 发生反应的概率仍然为 $a_k(X(t))\Delta t + o(\Delta t)$ 。这与无时滞化学反应系统的反应通道发生概率是一样的。因此在时间区间 $[0, t)$ 内反应通道 R_k 发生的反应次数仍然为 $Y_k(\int_0^t a_k(X(s))ds)$ 。为方便起见, 在本章中将状态改变向量分成生成向量 v_j 和消耗向量 v'_j 。假定系统在时刻 t , $X(t) = x$, 存在 N_d 个未完成的时滞反应事件, 这些事件将在 L_1, \dots, L_{N_d} 时刻完成, T_k, P_k 与无时滞的化学反应系统表示的意义一样。则没有反应开始或者反应结束的情况下, 通道 R_k 发生下一反应事件的时间间隔仍然为 $\Delta t_k = \frac{P_k - T_k}{a_k(X(t))}$ 。DMNRM 与 MNRM

唯一不同的是跟踪和记录时滞通道的完成时间。我们用一个二元结构体 W 来存储 (L_i, μ_i) , 其中 μ_i 表示时滞反应通道 R_{μ_i} , W 按 L_i 大小升序存储。因此系统状态发生改变的绝对时间间隔为 $\Delta_j = \min_{i \in \{1, \dots, M\}} \{\Delta t_i, W(1,1) - t\}$ 。更新 $T_k = T_k + a(X(t))\Delta_j$, 如果 $\Delta_j = \Delta t_i$, 更新 $P_i = P_i + \ln \frac{1}{r}$, $r \in U(0,1)$, 状态改变需要分类考虑 1) $i \in ND$, $x = x + v_i - v'_i$ 2) $i \in CD$, 状态不更新。3) $i \in ICD$, $x = x - v'_i$; 如果 $\Delta_j = W(1,1) - t$, 状态更新需要分两类考虑: 1) $W(1,2) \in ICD$, 更新系统 $x = x + v_{W(1,2)}$, 2) $W(1,2) \in CD$, 状态更新为 $x = x + v_{W(1,2)} - v'_{W(1,2)}$ 。然后对存储向量进行更新。这就是 DMNRM 算法。

4.2 模拟时滞化学反应系统的自适应 τ -Leap 算法

假设系统在时刻 t 有 N_d 组未完成的时滞事件, 这些事件将分别在 L_1, \dots, L_{N_d} 时刻完成, 其中第 j 组事件为通道 R_{μ_j} 在时刻 t 前面发生 K_{μ_j} 个时滞事件 ($j = 1, \dots, N_d$)。用一个三元结构体 S 存储 (L_j, μ_j, K_{μ_j}) 。不是一般性, 假设完成时间为 $t = L_0 < L_1 \leq L_2 \leq \dots \leq L_{N_d} < L_{N_d+1} = \infty$ 。为方便后文讨论起见, 假设在 Leap 时间区间 $[t, t + \tau)$ 内除了由于那些将在该时间段内完成的时滞反应事件组将引起倾向函数的改变之外, 新发生的反应事件对倾向函数的改变均可忽略不

计。事实上，这一假设与无时滞化学反应系统中的 Leap 条件一致。

在满足 Leap 条件下的时间区间 $[t, t+\tau)$ 内系统的状态方程可以写成：

$$\Delta X(t, \tau) = \sum_{k \in ND} (v_k - v'_k) N_k - \sum_{k \in ICD} v'_k N_k + \sum_{j=1}^{N_d} \text{sgn}(t+\tau - T_j)_+ v_j^* K_{\mu_j} \quad (4.2.1)$$

$$\text{其中, } u_+ = \max\{u, 0\}, \quad v_j^* = \begin{cases} v_j & , j \in ICD \\ v_j - v'_j & , j \in CD \end{cases} .$$

在时间区间 $[t, t+\tau)$ 内通道 R_k 发生的反应次数可以近似为：

$$N_k = Y_k \left(\int_0^{t+\tau} a_k(x(s)) ds \right) - Y_k \left(\int_0^t a_k(x(s)) ds \right) = Y_k \left(\int_t^{t+\tau} a_k(x(s)) ds \right) \approx \text{Poisson}(F_k(t, \tau))$$

其中 $F_k(t, \tau) = \sum_{i=1}^j a_j(X(L_{i-1}))(L_i - L_{i-1}) + a_j(X(L_j))(t+\tau - L_j), L_j \leq t+\tau < L_{j+1}$ 。

为了将后验 τ -Leap 算法应用到时滞化学反应系统，我们需要给出计算 $F_k(t, \tau)$ 和状态方程(4.2.1)的有效算法。但由于状态方程(4.2.1)和 $F_k(t, \tau)$ 的复杂性，我们需要给出一个比较合适的初值：先验 τ 。下面分别给出先验 τ 和 $F_k(t, \tau)$ 以及状态方程(4.2.1)的算法。

4.2.1 计算先验 τ

为了保证在时间区间 $[t, t+\tau)$ 内满足 Leap 条件，反应物分子变化量 $\Delta X_k(t, \tau)$ 须满足

$$\Delta X_k(t, \tau) \leq \max \left\{ \frac{\varepsilon X_k(t)}{g_k}, 1 \right\} \quad (4.2.2)$$

由于 $\Delta X_k(t, \tau)$ 是随机数，用 $\Delta X_k(t, \tau)$ 的均值和方差代替不等式(4.2.2)的 $\Delta X_k(t, \tau)$ ，有

$$|E(\Delta X_k(t, \tau))| \leq \max \left\{ \frac{\varepsilon X_k(t)}{g_k}, 1 \right\} \quad (4.2.3)$$

$$\text{var}(\Delta X_k(t, \tau)) \leq \max^2 \left\{ \frac{\varepsilon X_k(t)}{g_k}, 1 \right\} \quad (4.2.4)$$

分别计算 $|E(\Delta X_k(t, \tau))|$ 与 $\text{var}(\Delta X_k(t, \tau))$ 得

$$E(\Delta X_k(t, \tau)) = \sum_{i \in ND} (v_{ik} - v'_{ik}) F_i(t, \tau) - \sum_{i \in ICD} v'_{ik} F_i(t, \tau) + \sum_{i=1}^{N_i} \text{sgn}(t + \tau - L_i) v'_{ik} K_i \quad (4.2.5)$$

$$\text{var}(\Delta X_k(t, \tau)) = \sum_{i \in ND} (v_{ik} - v'_{ik})^2 F_i(t, \tau) + \sum_{i \in ICD} (v'_{ik})^2 F_i(t, \tau) \quad (4.2.6)$$

由于 $F_k(t, \tau)$ 和状态方程 (4.2.1) 的复杂和 Leap 条件要求倾向函数 $a_k(X(t))$ 几乎保持不变以及我们还可以在 DA τ -leap 算法中调整 τ 的大小。故我们可以对先验 τ 计算中用 $\tau a_k(X(t))$ 代替 $F_k(t, \tau)$

$$\tilde{\mu}_k(X(t)) = \sum_{i \in ND} (v_{ik} - v'_{ik}) a_i(X(t)) - \sum_{i \in ICD} v'_{ik} a_i(X(t)) \quad (4.2.7)$$

$$\tilde{\sigma}_k^2(X(t)) = \sum_{i \in ND} (v_{ik} - v'_{ik})^2 a_i(X(t)) + \sum_{i \in ICD} (v'_{ik})^2 a_i(X(t)) \quad (4.2.8)$$

从而我们可以的得出先验 τ 的计算公式:

$$\tau = \min_{i \in I_n} \left\{ \frac{\max\{\varepsilon X_i(t)/g_i, 1\}}{|\tilde{\mu}_i(X(t))|}, \frac{(\max\{\varepsilon X_i(t)/g_i, 1\})^2}{\tilde{\sigma}_i^2(X(t))} \right\} \quad (4.2.9)$$

4.2.2 计算 $F_k(t, \tau)$ 、 $VX(t, \tau)$ 和指标 i

$VX(t, \tau)$ 表示在时间区间 $[t, t + \tau)$ 内没有通道发生新反应事件时在时刻 $t + \tau$ 的系统状态, 对每个通道 R_k , 我们用三元结构体 D_k 存储 (VX, F_k, L_i) , 其中 $VX = VX(t, L_i - t)$, $F_k = F_k(t, \tau)$ 并且按 L_i 升序排列。

算法 1: 计算 $F_k(t, \tau)$, $VX(t, \tau)$, 指标 i

(a) 给定 τ , 设 W_k 为 D_k 的行数;

(b) 如果 $t + \tau \geq D_k(W_k, 3)$;

(c) 设 W_k^1 为 $S(W_k : \text{end}, :)$ 的行数

$$i = W_k + 1; F_k = D(W_k, 2), VX = D(W_k, 1)$$

While $t + \tau \geq S(i, 3)$ 且 $i \leq W_k^1$

$$F_k = F_k + (S(i, 1) - S(i - 1, 2)) a_k(VX), \quad VX = VX + v_{S(i, 2)}^* S(i, 3)$$

将 $F_k, VX, S(i, 3)$ 按升序存储到 D_k 中

End while

$$F_k = F_k + (t + \tau - D_k(\text{end}, 3)) a_k(VX);$$

(d) 否则选择 i 满足 $D_k(i, 3) \leq t + \tau < D_k(i + 1, 3)$

$$F_k = D_k(i, 2) + (t + \tau - D_k(i, 3)) a_k(D_k(i, 1))$$

$$VX = D_k(i,1).$$

4.2.3 DA τ -Leap 算法

综上所述, DA τ -Leap 算法如下:

步骤 1 初始化: 初始时间及初始分子数目 $t=0, X(t)=X$, 时滞通道 R_{μ} 的时滞时间为 d_{μ} , 结束时间 t_f , $S=[0,0,0]$, 对每一个 k , $T_k = C_k = 0$, $Q_k = [0,0]$, $D_k = []$ 计算 a_k , 设 $0 < q < 1$;

步骤 2 while $t \leq t_f$

(a) 根据式 (4.2.9), 计算, 令每一个 $D_k = [x,0,t]$, $F_k = 0$;

(b) 根据算法 1, 计算 VX 、指标 d 及每一个 F_k ;

(c) 对每一个 k , 做下面的计算:

(a1) 设 B_k 为 Q_k 的行数;

(a2) 如果 $T_k + F_k \geq Q_k(B_k,1)$

$$N_k = \text{Poisson}(T_k + F_k - Q_k(B_k,1)) + Q_k(B_k,2) - C_k$$

$$\text{row}_k = B_k;$$

(a3) 否则

找到指标 i , 满足不等式 $Q_k(i-1,1) \leq T_k + F_k < Q_k(i,1)$

$$r = \frac{T_k + F_k - Q_k(i-1,1)}{Q_k(i,1) - Q_k(i-1,1)}$$

$$N_k = \text{Binomial}(Q_k(i,2) - Q_k(i-1,2), r) + Q_k(i-1,2) - C_k$$

$$\text{row}_k = i - 1;$$

步骤 3 根据得到的 N_k , 计算 $VX = VX + \sum_{i \in ND} (v_i - v'_i) N_i - \sum_{i \in ICD} v'_i N_i$ 检查 Leap 条件;

步骤 4 如不符合 Leap 条件

(a) 更新 Q_k

将 $[T_k + F_k, C_k + N_k]$ 插入到 row_k 和 $\text{row}_k + 1$ 之间, 如果

$\text{row}_k + 1 > B_k$, 增加新一行到 Q_k 的最后;

(b) 令 $\tau = q\tau$, 回到 2(b);

步骤 5 如符合 Leap 条件

(a) 更新每一个 Q_k

删除小于等于 row_k 的行，增加新的一行 $[T_k + F_k, C_k + N_k]$ 作为 Q_k 的第一行；

(b) 设 $t = t + \tau$, $X = VX$;

(c) 更新 S

删除小于等于 d 的行，对于所有的 $[t + d_{\mu_i}, \mu_i, v_{\mu_i}^* N_{\mu_i}]$ 按升序依次插入到 S 中，其中 $\mu_i \in R_D$;

(d) 更新每一个 $T_k = T_k + F_k$ 和 $C_k = C_k + N_k$;

(e) 计算 a_k ，回到 2(a);

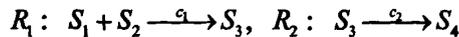
步骤 6 结束 while。

4.3 数值模拟

本节通过两个常见的时滞化学反应系统证实 $DA\tau$ -Leap 算法的模拟性能。因为 DMNRM 算法是精确算法，我们可以把它模拟得到的结果作为精确结果。在模拟中分别运行 DMNRM 算法与 $DA\tau$ -Leap 算法相同的次数，其中 $DA\tau$ -Leap 算法中分别取 ε 不同值，从而可以得到每一组实验的关于每一种分子的频率直方图。然后计算 ε 不同值下的 $DA\tau$ -Leap 算法下的结果与 DMNRM 算法下的结果的直方图距离。直方图距离^[42]是有效刻画一个随机模拟算法精度的方法之一。同时，为了说明算法的模拟速度，我们也考虑了直方图距离与 CPU 时间的关系。这里所有的模拟工作均是用 MATLAB6.5 实现的，其运行环境为 WinXP 系统, 1.90GHzCPU 与 256Mb 内存。

4.3.1 双通道系统

该系统有两个反应通道，四种分子：



在该系统中， R_2 为无时滞反应通道， $R_1 \in R_{CD}$ ，即当 R_1 发生反应事件时，系统立即更新 $X_1 = X_1 - 1, X_2 = X_2 - 1$ ，但 X_3 只能在时滞 d_1 之后才能更新，实验所用的参数值为：

$$c_1 = 0.001, c_2 = 0.01, X_1(0) = X_2(0) = 4000, X_3(0) = 5000,$$

时滞时间为 $d_1 = 0.1$ ，且模拟时间从 $t_s = 0$ 到 $t_f = 1$ 结束。分别运行 DMNRM 和 DA τ -leap 算法 10000 次。得到各自结果的直方图，并计算不同 ε 值下的 DA τ -Leap 算法的直方图与 DMNRM 算法的直方图的距离。

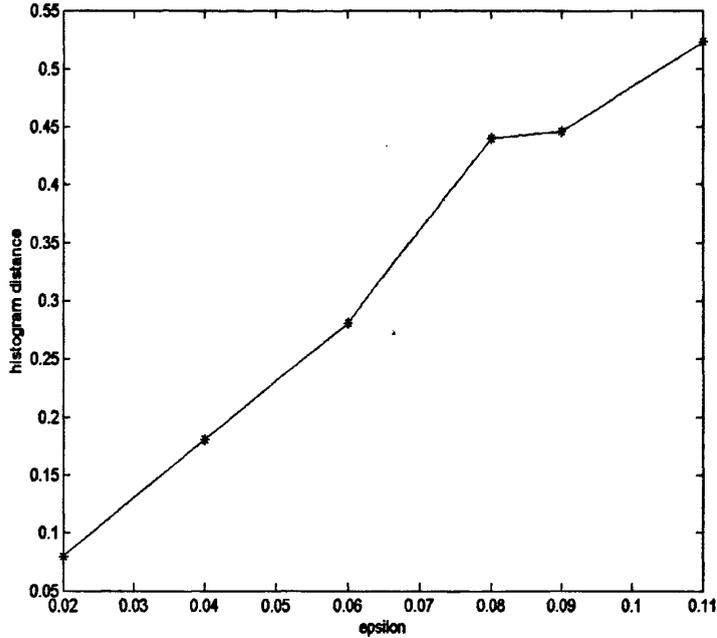


图 4.1 双通道模型的不同 ε 值下的 $X_3(1)$ 的直方图距离

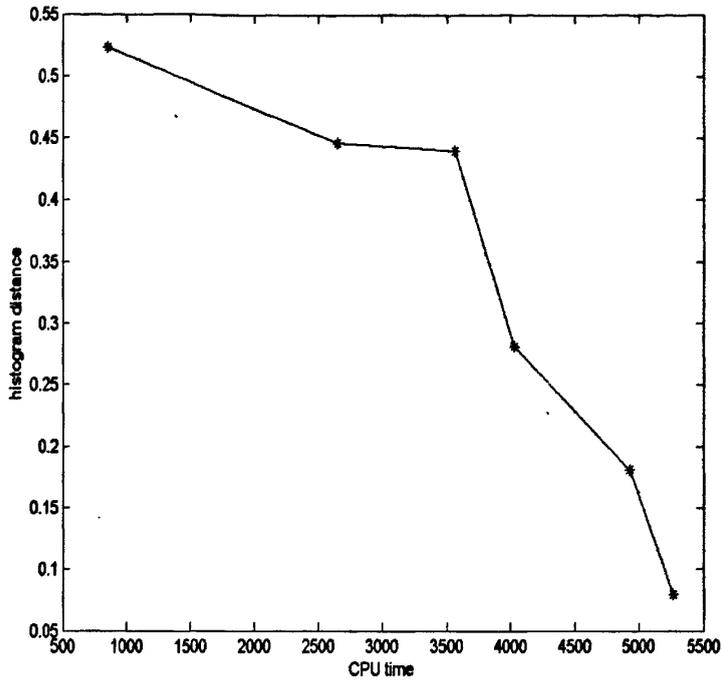


图 4.2 双通道模型的 $X_3(1)$ 的直方图距离与 1×10^4 次运行 CPU 时间的关系，运行 DMNRM 1×10^4 次的 CPU 时间为 44850 秒。

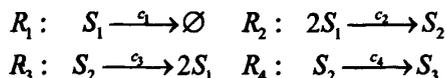
图 4.1 给出了 DMNRM 算法与不同 ε 值下的 $DA\tau$ -Leap 算法的 $X_3(1)$ 的直方图距离。可以看出随着 ε 值的变小，直方图距离也随着变小。当 ε 值很小时， $DA\tau$ -Leap 算法可以得到很好的模拟效果。图 4.2 给出了 $X_3(1)$ 的直方图距离与计算机模拟的 CPU 时间之间的单调递降关系。显然随着 CPU 时间的增加，二者的直方图距离均减少。反之随着直方图的距离的增加，CPU 时间均降低。表 4.1 给出了不同 ε 值下的 $X_3(1)$ 的直方图距离与 CPU 时间。

表 4.1 双通道模型中不同 ε 值对应的 $DA\tau$ -Leap 算法性能比较

ε	0.02	0.04	0.06	0.08	0.09	0.11
Hist. D($X_3(1)$)	0.0800	0.1808	0.2808	0.4400	0.4460	0.5232
CPU time	5258.4	4927.3	4030.7	3557.4	2650	850.8

4.3.2 降解聚合系统

常见的聚合反应系统有 4 种分子和 4 个反应通道组成，如下表示：



在该系统中，单体分子 S_1 聚合得到一个不稳定形式 S_2 ，该合成过程具有逆反性， S_2 可以转化成稳定分子 S_3 ，同时 S_1 可以降解。假设 $R_4 \in R_{CD}$ ，其余的通道都不存在时滞性。模拟计算中，取反应参数分别为 $c_1 = 0.001, c_2 = 0.01, c_3 = 0.5, c_4 = 0.04$ ，初始分子数目分别为 $X_1(0) = 10000, X_2(0) = 3000, X_3(0) = 0$ 。通道 R_4 的时滞时间为 $d_4 = 0.1$ 。反应开始时间为 $t_s = 0$ ，结束时间为 $t_f = 10$ 。

图 4.3 给出了 DMNRM 算法与不同 ε 值下的 DA τ -Leap 算法的 $X_3(10)$ 的直方图距离。可以看出随着 ε 值的变小，直方图距离也随着变小。当 ε 值很小时，DA τ -Leap 算法可以得到很好的模拟效果。图 4.4 给出了 $X_3(10)$ 的直方图距离与计算机模拟的 CPU 时间之间的反比关系。显然随着 CPU 时间的增加，二者的直方图距离均减少。反之随着直方图的距离的增加，CPU 时间均降低。表 4.2 给出了不同 ε 值下的 $X_3(10)$ 的直方图距离与 CPU 时间。

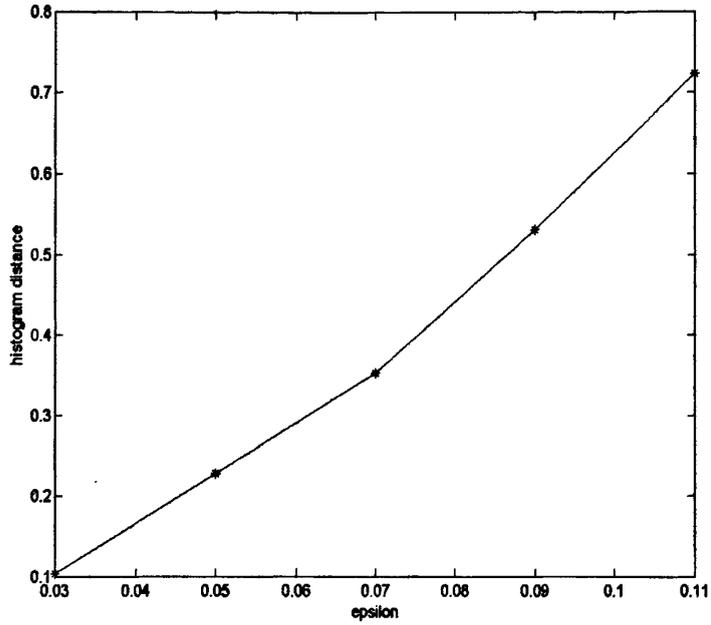


图 4.3 降解聚合模型的不同 ε 值下的 $X_3(10)$ 的直方图距离

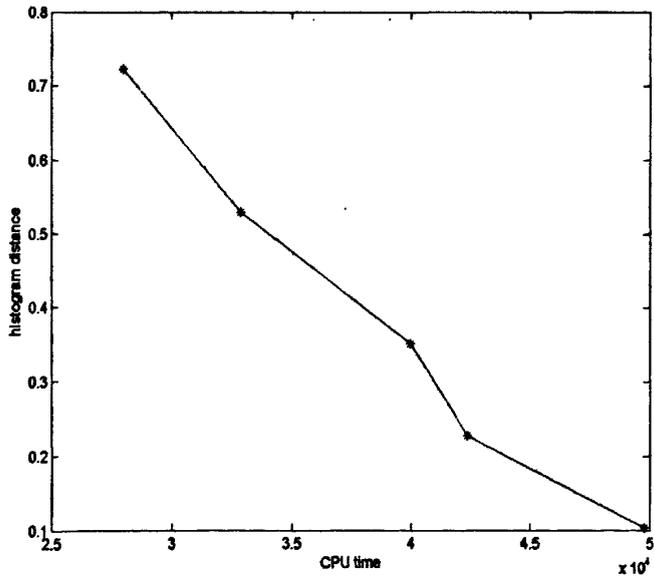


图 4.4 双通道模型的 $X_3(10)$ 的直方图距离与 1×10^4 次运行 CPU 时间的关系，运行 DMNRM

1×10^4 次的 CPU 时间为 59821 秒

表 4.2 降解聚合系统中不同 ε 值下的的 DA τ -Leap 算法性能比较

ε	0.03	0.05	0.07	0.09	0.11
Hist. D($X_3(10)$)	0.1032	0.2274	0.3522	0.5301	0.7230
CPU time	49785	42367	39870	32890	27970

4.4 结语

本文提出的 DA τ -Leap 算法是模拟时滞化学反应系统的快速近似算法。该算法考虑了时滞的特性，给出了时滞状态下的近似状态方程，能够在给定的时间区间 $[t, t+\tau)$ 内，并且精确地满足 Leap 条件下发生多次反应事件，避免了负分子数目的产生，从而保证了算法的精度。该算法在牺牲可容忍的精度条件下，能够快速模拟分子数目较大的时滞化学反应系统。数值模拟实验证实了 DA τ -Leap 算法的性能。

第五章 结论与展望

5.1 结论

对基因调控网络动力学性质的研究所涉及的范围非常广泛，以下三个方向引起了国内外学者的极大兴趣，这三个方向分别是（1）基因调控网络中的随机性；（2）基因调控网络中的时滞性；（3）基因调控网络中的空间特性。本文以基因调控网络模拟算法为研究对象，对基因调控网络中的随机模拟算法进行了一些探索性的工作。主要取得了如下成果：

- 1) 总结归纳了基因调控网络的动力学模型以及详细介绍了基因调控网络的几种精确随机模拟算法及快速近似算法。
- 2) 对于刚性基因调控网络算法的进展进行了介绍，给出了 Cao 等人提出的 MSSA。MSSA 即具有坚实的理论基础，又便于应用，是模拟复杂刚性的非线性化学反应系统的有效算法之一。但是当慢—子系统的分子数目较大时，该算法由于模拟慢—子系统的每个事件，因而导致该算法效率较低。本文在 MSSA 的基础上发展了改进的多尺度算法 (IMSS τ)。因充分研究了快—子系统和慢—子系统的相互作用，故对于慢—子系统的分子数目较大的刚性系统，IMSS τ 比 MSSA 更能有效地模拟。数值实验表明 IMSS τ 算法能够在损失较小的精度下有效地提高了模拟速度。
- 3) 对时滞基因调控网络模拟算法进行了介绍和研究。介绍了模拟时滞基因调控网络的改进下一反应法，该算法是精确模拟时滞化学反应系统的算法。但是对于复杂的时滞化学系统，效率仍旧低下。本文发展了一种模拟具有时滞化学反应系统的自适应 τ -Leap 算法 (DA τ -Leap)。该算法利用单位泊松随机过程的性质，将后验 τ -Leap 算法应用到了时滞化学反应系统，能够自动调节 τ 使得在时间区间 $[t, t+\tau)$ 内发生多次反应事件并且精确地满足 Leap 条件，从而避免了负分子数目的产生，较大地提高了模拟速度。数值实验证实了 DA τ -Leap 算法的优点。

5.2 展望

通过本文对基因调控网络的模拟算法的总结和研究,作者认为在该领域还存在在以下几个方面值得进一步研究。

- 1) 时滞基因调控网络里同样存在刚性问题,但是由于时滞的影响,使得基因调控的动力学性质更加复杂。虽然,在模拟无时滞的基因调控网络随机算法里,已有许多有效算法克服刚性问题,但是如何将这些算法推广到时滞基因调控里仍值得深入研究。
- 2) 本文研究的基因调控网络是齐性空间,即分子在反应过程中分布始终是保持均匀的。但在现实中,由于重力,电荷里,细胞结构等各种物理因素的存在,分子在空间分布一般来说是不均匀的。因此需要一类算法模拟非齐性空间的基因调控网络系统。虽然已发展了一些算法^[43-46]针对模拟非齐性空间的基因调控网络,但是由于非齐性空间的基因调控网络仍然存在刚性问题、时滞现象以及两者的综合影响,已有的算法仍然不能有效模拟这类网络。因此模拟非齐性网络的算法仍是以后研究的重点之一。

综上所述,对基因调控网络的数值模拟有助于了解生命体内组分的时空演化。虽然我们现在对基因调控已经有了一定的研究成果,但是离模拟接近真实的细胞生化网络仍有很长的路要走。我们相信随着许多算法工作者的重视和研究,会设计出更加有效的算法,从而为深入研究基因调控网络提供便利,这将有助于我们更加接近生命系统的本质。

参考文献

- [1] 彭司华, 周洪亮. 系统生物学的分析与建模. 信息与控制[J], 2005, 33(4): 457-462.
- [2] 石铁流, 李亦学. 系统生物学的现状与展望. 中国科学基金[J], 2005, 19(5): 282-286.
- [3] 林辉. 系统生物学及其研究进展. 重庆医学[J], 2005, 34(9): 1416-1417.
- [4] 焦传珍. 系统生物学及其研究进展. 系统生物学及其研究进展[J], 40 (12) 3-4
- [5] 翟友刚, 史朋, 谢忠平. 系统生物学的现状与展望. 医学研究通讯, 34(3) 2005, 61-62.
- [6] 马宏宾, <http://complex.amss.ac.cn/resource/talks/Regulation.ppt>
- [7] Elowitz M.B., Levine, A.J. and Siggia, E.D, et.al. Stochastic gene expression in a single cell. Science [J], 2002, 297:1183-1186.
- [8] Raser J.M. and Shea E.K. Control of stochasticity in eukaryotic gene expression. Science [J] 2004, 304:1811-1814.
- [9] Kaern M, Elston T. and Blake W.J, et al. Stochasticity in gene expression: from theories to Phenotypes [J]. Nature Reviews Genetics, 2005, 6: 451-464.
- [10] Rao C.V., Wolf D.M. and Arkin A.P. Control, exploitation and tolerance of intracellular noise [J]. Nature, 2002, 420:231-237.
- [11] Paulsson, J. Summing up the noise in gene networks [J]. Nature, 2004, 427:415-418.
- [12] Thattai M. and Oudenaarden A. Intrinsic noise in gene regulatory networks [J]. Proc. Natl Acad. Sci. USA, 2001, 98:8614-8619.
- [13] Kepler T.B and Elston, T.C. Stochasticity in transcriptional regulation: origins, consequences, and mathematical representations [J]. Biophys. J., 2001, 81:3116-3136.
- [14] Rao, C.V., and Arkin, A.P. Stochastic chemical kinetics and the quasi-steady-state assumption: Application to the Gillespie algorithm [J]. Chem. Phys., 2003, 118:4999-5010.
- [15] Rathinam M, Petzold L.R and Cao Y, et al. Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method.[J]. Chem. Phys, 2003, 119:12784-12794.
- [16] Christopher V.R and Adam P.A. Stochastic chemical kinetics and the quasi-steady-state assumption: Application to the Gillespie algorithm.[J]. Chem Phys, 2003, 118:4999-5010.

- [17] Eric L.H and James B.R, Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics.[J]. Chem Phys,2002,117:6959-6979.
- [18] Cao Y., Gillespie D.T and Petzold L.R, The slow-scale stochastic simulation algorithm,[J]. Chem. Phys., 2005,122: 014116.
- [19] Cao Y, Gillespie D.T and Petzold L.R, Multiscale Stochastic Simulation Algorithm with Stochastic Partial Equilibrium Assumption for Chemically Reacting Systems,[J].Comput. Phys. 2005,206:395-411.
- [20] Bratsun D.A., Volfson D.N. and Hasty J, et al. Non-Markovian Processes in Gene Regulation [R]. Proceedings of SPIE, 2005, 5848:210-219.
- [21] Bratsun D.A., Volfson, D.N. and Tsinring L.S, et al. Delay-induced stochastic oscillations in gene regulation [J]. Proc. Natl Acad. Sci. USA, 2005, 102(41):14593-14598.
- [22] Barrio M, Burrage K and Leier A, et al. Oscillatory regulation of Hes1: discrete stochastic delay modeling and simulation. PLOS Comp Biol,2006,2:1017-1039.
- [23] Cai X.D. Exact stochastic simulation of coupled chemical reactions with delays. [J]. Chem Phys, 2007,126:124108.
- [24] David F. A modified next reaction method for simulating chemical systems with time depend propensities and delays. J Chem Phys 2007, 127: 214107.
- [25]Keizer J. Statistical thermodynamics of nonequilibrium processes [M]. Berlin: Springer-Verlag, 1987.
- [26] Yi T, Yu T. J and Gregory T. Stochastic fluctuations in gene expression far from equilibrium: O expansion and linear noise approximation [J]. Chem. Phys, 2005, 122:124108.
- [27] Gillespie D.T. A rigorous derivation of the chemical master equation.[J]. Physics A, 1992, 188:404.
- [28] Munsky B and Khammash M. The finite state projection algorithm for the solution of the chemical master equation.[J] Chem.Phys.,124,044104.
- [29] Gillespie D.T. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions [J]. Comp.Phy.,1976,22 (4): 403-434.
- [30] Gillespie D.T. Exact stochastic simulations of coupled chemical reactions[J].Phys chem,

- 1977,81(25):2340-2361.
- [31] Gibson M and Bruck J. Efficient exact stochastic simulations of coupled chemical reactions.[J].Phys Chem ,A, 2000,104:1876-1889.
- [32] 刘祥, 基因调控网络的空间模型及其算法研究[D], 上海: 上海大学, 2007.
- [33] Gillespie D.T. Approximate accelerated stochastic simulation of chemically reacting systems.[J].Chem Phys, 2001,115(4):1716-1733.
- [34] Gillespie D.T and Petzold L.R. Improved leap-size selection for accelerated stochastic simulation. [J] Chem Phys,2003,119:8229-8234.
- [35] Tian T.H and Burrage K. Binomial leap methods for simulating stochastic chemical kinetics. [J]. Chem Phys,2004,121:10356-10364.
- [36] Chatterjee A, Vlachos D.G and Katsoulakis M.A. Binomial distribution based τ - Leap accelerated stochastic simulation. [J] Chem Phys,2005,122:024112.
- [37] Cao Y, Gillespie D.T and Petzold L.R. Efficient step size selection for the tau-leaping simulations method.[J] Chem Phys, 2006, 124:044109.
- [38] Cao Y, Gillespie D.T and Petzold L.R, Avoiding Negative Populations in Explicit Tau Leaping, [J]. Chem. Phys.2005,123: 054104 .
- [39] Cai X.D and Xu Z.Y. K-leap method for accelerating stochastic simulation of coupled chemical reactions. [J]. Chem Phys, 2007, 126:074102.
- [40] David F. Incorporating postleap checks in tau-leaping. [J]. Chem Phys 2008, 128:054103.
- [41] Ethan A, Eric L and James B. Two classes of quasi-steady-state model reductions for stochastic kinetics. [J]. Chem Phys, 2007, 127:094106.
- [42] Cao Y. and Petzold L.R. Accuracy Limitations and the Measurement of Errors in the Stochastic Simulation of Chemically Reacting Systems, [J]. Comp Phy, 2006,212(1) 6-24.
- [43] Elf J, Doncic A. and Ehrenberg M. Mesoscopic reaction-diffusion in intracellular signaling[C]:Fluctuations and noise in Biological, Biophysical and Biomedical Systems, 2003,5110:114-124.
- [44]Elf J and Ehrenberg M. Spontaneous separation of bi-stable biochemical systems into spatial domains of opposite phases [J]. Syst Biol (Stevenage), Vol. 1, No. 2. (Dec 2004),

230-236.

[45]David B. Simulating mesoscopic reaction-diffusion systems using the Gillespie algorithm

[J]. Phys. Rev. E 71, 041103 (2005).

[46]刘祥, 周文, 闫正楼, 王翼飞.一种改进的非齐性空间反应—扩散主方程的随机模拟算

法[J]. 应用科学学报.2007,651-656.

作者在攻读硕士学位期间公开发表的论文

- [1] 闫正楼, 彭新俊, 周文, 刘祥, 王翼飞. 改进的多尺度算法. 北京: 计算机与应用化学(已投).
- [2] 闫正楼, 彭新俊, 周文, 刘祥, 王翼飞. 模拟时滞化学反应系统的自适应 τ -Leap算法. 上海: 应用数学与计算数学
- [3] 周文, 彭新俊, 刘祥, 闫正楼, 王翼飞. 加速随机模拟化学反应系统的“最后所有可能的步进”方法. 上海: 应用数学和力学
- [4] 刘祥, 周文, 闫正楼, 王翼飞. 一种改进的非齐性空间反应-扩散主方程的随机模拟算法. 上海: 应用科学学报.

致谢

在论文完成之际，我的学习生涯也告一段落。

首先感谢我的导师王翼飞教授对我的学习生活上的帮助，感谢他给了我一个温馨的学习环境。当我浮躁迷惘时，给了鼓励和安慰；当我犯错时，给了谅解。在此，向王老师说一声：辛苦了！

感谢马和平、李常品、贺国强教授等三年来的指导和帮助。

感谢师兄张家军、刘祥、彭新俊和师姐周文、师妹刘焕对我学业上的指导和帮助。

感谢冯铁男、张玉滨和孟炜等实验室的所有同学三年来的共同学习。

感谢席艳玲、靳庆芳、杜丽娟和雒晓良等山西老乡的关心和帮助。

感谢亲如兄弟的室友：杨振新、玄其飞和赵振刚三年来的风雨共济。

感谢鞠瑞亮、汪卫和徐从祥等一块踢球的同学们，是你们给了我生活上的乐趣。

感谢大学老师何松年等教授的培养和关心。

感谢我的大学同学：赵彦、李荣平、徐劲传等三年来的物质和精神上的支持。

感谢我的父母和4个兄长对我20来年的养育，没有他们的默默付出和支持，就没有我的今天。

最后，感谢上大的优美环境，让我在这里愉快地度过了人生重要的三年。