§1.2 随机模拟算法与 tau-leaping 方法（原文摘录）

随机模拟算法与方法§１．２ｔａｕ－ｌｅａｐｉｎｇ随机模拟算法ＳＳＡ是随机模拟空间均匀化学系统的一种精确策略它（）由ＤａｎｉｅｌＴ．Ｇｉｌｌｅｓｐｉｅ首次提出ＳＳＡ利用蒙特卡洛方法来近似模拟随机化学系统的时间演化．对于当前时刻ｔ，系统状态Ｘ⑷％且系统具有Ｋ个反应通道，每个通道在当前时刻的反应速率为为实现状态向量的更新算法需.，要确定两个关键问题即下一个反应发生的时间ｔ＋Ｔ与该时刻发生何种反应％?．为此，ＤａｎｉｅｌＴ．Ｇｉｌｌｅｓｐｉｅ证实可以通过生成两个随机数将这两个随机数带入与倾向函数相关的方程来确定ｔ＋ｒ与大量实验表明ＳＳＡ可以精确的完成化学反应系统的随机模拟．随后，科学家们利用此方法对生化系统进行了大量模拟［２（＾２２１．为使得ＳＳＡ更加有效的应用到各种系统中科学家们对ＳＳＡ作出了诸多改进例,，如为了减少运算量，ＭｉｃｈａｅｌＡ．Ｇｉｂｓｏｎ等人提出了ＮＲＭ法，排序直接法等［２３￣２６１，为提高精确度ＨａｌｕｋＲｅｓａｔ提出了具有概率权重的ＰＳＳＡ算法等虽然ＳＳＡ精，确地模拟了离散随机微观模型但由于ＳＳＡ需要跟踪每一个反应事件这对于含有,，大量分子物种与反应通道的系统来说模拟是非常耗时的因而无法应用到实际生,，活当中．ＤａｎｉｅｌＴ．Ｇｉｌｌｅｓｐｉｅ意识至ＩＪ可以采用一个近似方案，以确定的时间步长来推进该过程，而不是对每个反应进行采样．为了加快离散随机模拟的速度，ＤａｎｉｅｌＴ．Ｇｉｌｌｅｓｐｉｅ提出了ｔａｕ－ｌｅａｐｉｎｇ方法［２８１通过设定一些跳跃条件来选定时间步长ｔ由，ｔ确定跳跃区间然后根据每个反应发生的倾向函数与选定的时间步长ｔ来产生.，泊松随机数利用这些随机数来近似跳跃区间内可能发生的化学反应次数这样就,．可以跳过许多反应而不需要跟踪每一个反应事件．在某些情况下，ｔａｕ－ｌｅａｐｉｎｇ方法可以在模拟速度上产生显著的增益，但在精度上有可接受的损失．ｔａｕ－ｌｅａｐｉｎｇ方法提供了从离散随机状态下的ＳＳＡ到动力学状态下的化学朗之万方程和连续确定性状态下的反应速率方程的天然桥梁％为了使这种加速策略成功应用到化学反应系．统的模拟中，ＡｂｈｉｊｉｔＣｈａｔｔｅｒｊｅｅ等人对原始的ｔａｕ－ｌｅａｐｉｎｇ方法做出了改进，提出了更有效，更稳健的跳跃步长选择策略，如：二项ｔａｕ－ｌｅａｐｉｎｇ方法、Ｒ－ｔａｕ－ｌｅａｐｉｎｇ方法、Ｋ－ｔａｕ－ｌｅａｐｉｎｇ方法等．针对刚性系统ＭｕｒｕｈａｎＲａｔｈｉｎａｍ、ＹａｎｇＣａｏ等人还提出了隐式ｔａｕ－ｌｅａｐｉｎｇ算法、自适应ｔａｕ选择方法等等［２９__气ＤｅｓｍｏｎｄＪ．Ｈｉｇｈａｍ等人将这种方法与计算机软件相结合编译出了模拟随机化学反应系统的ＭＡＴＬＡＢ代码，极大提高了计算效率％．这种ｔａｕ跳跃方法在化学动力学和系统生物学领域中发挥重要作用科学家们将其应用到随机系统的模拟算法中进行加速对于由化,．学朗之万方程（ＣＬＥ）生成的一类ＳＤＥ，这种基于ｔａｕ－ｌｅａｐｉｎｇ的模拟方法会加速处理此类随机微分方程更加便捷有效的获取各类统计数据近些年来一些数学家,.，们针对不同的随机系统在ｔａｕ－ｌｅａｐｉｎｇ方法的基础上进行推广提出了不同的跳跃,，方法，如：ＡｂｄｕｌｌｅＡｓｓｙｒ等人提出了ｔａｕ－Ｒｏｃｋ方法［３８１对于带有离散噪声的刚性随机系统，ｔａｕ－Ｒｏｃｋ方法可以显著的加速．ＸｉｎｊｕｎＰｅｎｇ等人提出了加快时滞随机微分方程（ＤＳＳＡ）的ＤｔＬ方法和ＤＫ－Ｌｅａｐ算法［３９，４Ｑ１．对于具有快速和慢速时间尺度为特征的随机系统，ＣｈｉｈｅｂＢ．Ｈａｍｍｏｕｄａ提出了多级漂移__隐式ｔａｕ－ｌｅａｐｉｎｇ方法［４１彳将ｔａｕ－ｌｅａｐｉｎｇ方法应用到了随机反应网络（ＳＲＮ）－－某些特定的连续时间马尔可夫

链当中为了有效地估计ＳＲＮ的各种统计量特别是罕件事件概率他.（），们提出了基于ｔａｕ－ｌｅａｐｉｎｇ的近似方案来提高蒙特卡洛估计器的效率.

§2.3 tau-leaping 原理（原文摘录）

§．３ｔａｕ－ｌｅａｐｉｎｇ原理考虑在搅拌均匀的化学系统中具有ｎ种反应物种汾Ｊ的化学反应系统，{,，该系统具有Ｍ种化学反应通道系统的状态久⑷三其中足⑴表示物种民在ｔ时刻的分子数．反应焉的动力学由倾向函数％和状态改变量％三（ｉｉｊ,... ｉＶｊ.）描述:％．〇ｒ）出表示对于给定的初始状态Ｘ⑷＝〇:，反应尽在下一个时间段［ｔ，ｉ＋刹发生的概率,％表示物种氏发生尽反应的改变量．系统的动力学行为遵循化学主方程ＣＭＥ（）：ｄＰｘ＾＾＾°＝＾［ａ＾ｘ－Ｖｊ）Ｐ（ｘ－Ｖｊ，ｔ＼ｘ〇，ｔ〇）－ａｊ（ｘ）Ｐ（ｘ，ｔｘ〇，ｔ〇)]，其中尸表示对给定初始时刻的初始状态文⑷＝邱在（时刻文⑷＝￥，的概率然而除了一些简单的反应系统之外化学主方程在计算上是很难处理的因.,．此数学家们采用随机模拟算法ＳＳＡ来近似模拟系统的状态演化，（）：首先令，Ｍａ０（｜）＝〉：Ｑｊ（｜），ｊ＝ｌ下一个反应发生的时间Ｔ是期望值为的指数型随机变量反应尽发生的概，率为为了模拟系统在时间ｔ从状态ｚ发生的改变ＳＳＡ生成两个均匀分布的随机,，数ｎ和？′２．ｎ用来产生下一个反应的发生时间｜＋Ｔ，ｔ＝—ａ厂＾＾ｎ（—ｎ),（２．１３）〇（）用来确定发生何种反应ｒ２＾２ａｊ（ｘ）＞ｒ２〇＞ｏ（＾）－（２．１４）Ｊ＝ｉ随后，系统更新状态＋ｔ）＝ａ：＋％．ＳＳＡ模拟对每一个单独反应的细致跟踪让我们可以观察到尤的完整变化但，是对于现实中存在大量化学反应的随机系统来说这种构造通常是非常耗时的因,．此，Ｇｉｌｌｅｓｐｉｅ提出了ｔａｕ－ｌｅａｐｉｎｇ思想，他将时间轴分为一组连续的子区间，在这些子区间中一些反应的发生对系统的改变是不明显的在这种情况下如果可以确定.，每个子区间每个反应发生的真实次数就可以放弃了解这些反应发生的精确时刻,，进而从一个子区间跳跃到下一个子区间而不是从一个反应事件跳到下一个反应事，件．假设令＆表示在固定的时间段［Ｍ

＋ｒ］内，发生巧反应的真实次数．对于任意的Ｔ直接计算是非常困难的因此可以选择合适的Ｔ使其满,.，足跳跃条件：跳跃条件这里我们要求Ｔ足够小以至于在＋内倾向函数不会发生明显变:，化？令??（＃（０;），Ｔ）为／＾（Ｔ；Ｘ，ｔ）的近似，其中尸（ａ，Ｔ）是带有均值ｏｒｒ的Ｐｏｉｓｓｏｎ随机变量反应次数的近似平均值为本文的附录中将给出证明则尤在此，（），时间段内发生的变化期望值为ｎ。三。（Ｘ，Ｔ）＝土（尤）１＞】土＝ｊ＝ｌ其中办）＝Ｅ？＝ｉ疔０）１土，＜＝土"？对于Ｔ的选择主要有两种方法一种是对每个时间段内的倾向函数的变化,：ｇ（ｘ＋Ｇ，ｙｔ（ａ：＋ｉｊ））－咖，＃（３:)），ｊ＿＝进行检查，若差异太大，则尝试较小的ｔ但这个过程可能会比较耗时对于第二种方法我们要求倾向函数在时间,.，[ｔ，ｔ＋Ｔ）内的变化差值｜ｇ（；ｒ＋ｔｔ;，ｙ＾＝（：ｃ＋ａ;））－ｇ（ａ:，ｙｊｔ（ａ;)），ｊ＿＝ｌ,... ｎ，由所有倾向函数的和的某个特定分数倍界定即ｅ，ｑ（ｘ＋ｕｊ，ｙｆ（ｘ＋ｕｊ））－ｑ（ｘ，ｙｆ（ｘ））＜ｅｑ〇（ｘ），ｊ＝１,... ｎ.（２．１５）在等式左侧使用泰勒展开式ｑ（ｘ＋ｕ，ｙｆ（ｘ＋ｕｊ））－ｑ（ｘ，ｙｆ（ｘ））＼＾０－Ｖｑｆ｛ｘ）－ＯＸｉ＝ｉｉ定义唠⑷三与＾，由（３．１）可得ｎＴ⑷唁⑷仝喻⑷．Ｉｉ＝ｌ因此ｔ的选择为ｎＴ｛叫。⑷／

（２－！ ６）４＝１利用等式｜２．１６｜选择ｒ显然需要耗费一定的计算精力，因为这里需要计算２ｒｉ个和＆（Ｘ）＆＃（〇：)？选定Ｔ之后，由于在［Ｍ＋Ｔ］内反应次数呈泊松分布［１９１，因此可以通过使用泊松随机数模拟一段时间内的反应次数，进而可以跳过许多反应．令尸（ａｊ（：Ｃ），Ｔ）作为的近似，其中＾Ｐ（ａ，Ｔ）是带有均值ａＴ的ｐｏｉｓｓｏｎ随机变量？所以，对于ｔ时刻的状态Ｘ⑴＝％我们选择合适的ｔ满足跳跃条件，即可更新＋ｒ）处的状态：Ｍ＾Ｘ（ｔ＋ｔ）＝ｘ＋ｙｊＶｊＰ（ａｊ（ｘ），Ｔ），（２．１７）ｊ＝ｉ其中对于每一个表示均值为ａｊ（ａ；）Ｔ的独立泊松随机变量．这样的一个计算过程被称为ｔａｕ－ｌｅａｐｉｎｇ近似．１１

§3.1 基于空间离散格式的 tau-leaping 算法（原文摘录）

§．１ｔａｕ－ｌｅａｐｉｎｇ算法考虑带有加法噪声的维随机微分方程ＴＺｄＸｔ＝＾（Ｘｔ）ｄｔ＋ａｄＷｔ，（３．１）其中Ｗ＊是ｎ维的白噪声，：Ｒ＂—■Ｒｎ和ａ：Ｒ？４吧１分别为漂移项和常系数扩散项．ＳＤＥ（３．１）的解｛Ｘ山ｅ［Ｑ，Ｔ］是一个随机过程，其在时间连续下的空间离散过程可以看作是质点粒子从初始位置出发，在ｌ〇，Ｔ］内的运动过程．对于Ｎａｗａｆ Ｂｏｕ－Ｒａｂｅｅ和Ｅｒｉｃ Ｖａｎｄｅｎ－Ｅｉｊｎｄｅｎ等人提出的空间离散ｇ方法，考虑在连续时间上的空间离散化质点尤在ｎ维网格空间中进行随机游走在ｔ时刻所处的位置为，，尤＝（尤，，．．．尤?；）．下一时刻质点按照Ｑ算法进行跳跃，在任一维方向中选择前进或后退一步然而质点每跳跃一步算法都需要重新计算倾向函数和跳跃时间跳跃．，，，方向等我们的模拟需要长时间观察解的动力学行为这对模拟工作来说是非常耗．，时的因此我们借助化学反应系统中的随机模拟方法将ｎ维网格空间看成是包含．，．ｎ种化学物种＆＆的化学系统具有加个反应通道｛，???｝，：每次只有一个物种发生反应．在每一个反应时刻，第ｉ个物种选择或者ＡＣ反应通道进行反应．从这个观点出发，借助ＤａｖｉｄＨ．Ｇｉｌｌｅｓｐｉｅ提出的随机模拟化学反应系统的加速ｔａｕ－ｌｅａｐｉｎｇ算法．我们提出了基于＜５方法的加速策略．首先，根据空间离散＜３算法计算出各个方向的倾向函数ｇ〇ｒ，＝１，，…ｎ），咖為土⑷⑷表示对于给定的初始状态文⑷＝ａ；，在下一个时间段［匀ｔ＋ｄｔ］，质点由ｚ跳跃到＃㈦的概率．计算状态改变量％三（％，．．．％；），其中％表示质点跳跃到＃后，第ｉ维的状态改变量．选择满足跳跃条件的时间步长Ｔ．将时间轴分为多个连续的子区间使得在时间子区间＋内一些跳跃的发生对系统的改，，变是不明显的在这种情况下如果我们可以确定每个区间段内在各个方向的近．，，似跳跃次数我们就可以放弃了解这些反应发生的精确时刻进而从一个时间子区，，间跳跃到下一个子区间计算多次跳跃的结果而不是每次都进行跳跃这样就可以，，．大大加快模拟的速度．令Ｋｊｆｒｒｊ）表示在固定的时间段＋内，发生跳跃的真实次数对于时间步长ｔ直接计算是非常困难的我们知道在每个时间．，．区间内质点的跳跃次数是呈泊松分布［２８］的且泊松分布的参数为⑷＊ｔ，，）（证明见附录）．因此，我们将通过使用泊松随机数＆（Ｔ；：ｒ，ｉ）来近似一段时间内的跳跃次数，这样将会减少计算次数，提高随机模拟的速度？然后，用七＋４＝Ａ＋巧％来近似计算这段时间内的实际位移ｌ（ｔ＋Ｔ）ｊ＝．更新状态和时间之后继续进行下一时间段内的跳跃．算法的主要步骤为：步骤１：选择符合跳跃条件的Ｔ，生成平均值为咖，＃（〇；））的ｐｏｉｓｓｏｎ随机数Ａ＾，（ｉ＝

１，... ｎ），用来近似在时间［Ｍ＋爿内系统在第《个位置前进或后退的次数其中下一状态的倾向函数，：Ｑｃ？咖＃））＝咎ｅｘｐ１＾，（）＾ｑ（ｘ，ｙｒ（ｘ））＝＾ｅＷ（－），Ｑｕ？咖队＋⑷＝糾⑷ｖ〇＋令，）｜（（）），Ｑ（ｘ，ｙｉ（ｘ））＝＾（－（／ｉｉ（ｘ）Ｖ０）＋

＾），步骤更新状态：＋Ｔ）＝Ｘ⑷＋Ａ＊／ｉ＝Ｘ⑷＋（Ａ＾－入＾"－Ａｆ，… ４－ＡＳ）＊心（３．２）步骤３：更新时间：ｔｔ＋ｔ，统计各个网格点处的数据，继续下一次循环．步骤４生成个各网格点处的概率密度函数：．在计算机中实现的主要代码如算法所示１：Ａｌｇｏｒｉｔｈｍ１ｎ维ＳＤＥ数值平稳密度的ｔａｕ－ｌｅａｐｉｎｇ算法输入：状态下界ｌｏｗｘ，样本容量Ｓｐ，网格数量Ｎ，停止时间ｔ＊，初始状态文〇，零矩阵ｄａｔａ＾ｗ，…ｘＡｒ计数ｎｕｍ，输出：不变概率密度ｄａｔａ１：ｆｕｎｃｔｉｏｎｔａｕ－ｌｅａｐｉｎｇ（ｌｏｗｘ，Ｓｐ，Ｎ，ｔ＊，Ｘ〇）２：ｗｈｉｌｅｔ＜ｔ＊ｄｏ３：ｆｏｒｊ：１—＞ｎｄｏ４利用Ｑ方法计算＜和：；５：生成泊松随机数？ｐｏｉｓｓｏｎ＾ｑ＾Ｔ）；６：更新状态变化量？＜— 文＋Ａｊ Ｖｊ）；７：记录质点落入的网格点：ｇｊｒｏｕｎｄ（（文ｔｊ－（ｌｏｗｘ）』＋ｈ／２．０）／ｈ）；８：ｅｎｄｆｏｒ９：ｉｆｇｘ＞１ａｎｄｇｌ＜＝Ｎ，ｇ２＞１ａｎｄｇ２＜＝Ｎ，...ｇｎ＞１ａｎｄｇ？＜＝Ｎｔｈｅｎ１０：ｄａｔａ（ｇｌ５ｇ２，... ｇｎ）＾ｄａｔａ（ｇ１，ｇ２，... ｇｎ）＋１；１１：ｎｕｍ？＜—ｎｕｍ＋１；１２：ｅｎｄｉｆ１３：ｔｔ＋ｒ；：ｅｎｄｗｈｉｌｅ１５：计算概率密度函数ｄａｔａ—ｄａｔａ／（ｈｎｘｎｕｍ）；１６：ｅｎｄｆｕｎｃｔｉｏｎ

§3.3 中点格式的 tau-leaping 算法（原文摘录）

§．３ｔａｕ－ｌｅａｐｉｎｇ在第２．３节中，我们给出了选择Ｔ的跳跃条件，该条件要求倾向函数在时间Ｔ内无明显变化但是在实践中我们发现如果我们要采取较大的跳跃来产生比ｇ算，，法更快的模拟总是会产生一些倾向函数的变化这些变化将会不可避免的引入一，，些误差．当我们用简单的欧拉方法数值求解形如ｄＸ／ｄｔ＝／｜Ｘ｜的常微分方程时，会出现类似的困难．其中尤根据＝Ｘ⑷⑴）Ａｔ沿着时间轴跳跃时，只要函数／在Ａｔ增量期间发生变化，就会产生误差．减少这种误差的一种方法是使用估计中点的程序改进：厶１文＝１／（；＾））？，令；＾＋？）＝￥（）＋／（文（〇＋０１文）？．为了使这种估计中点策略适应ｔａｕ－ｌｅａｐｉｎｇ方法我们将状态的预期变化量Ｇ，类似欧拉增量／＾又，在跳跃过程中取：ｃ＋向／２］作为估计的中点状态，因此我们在时刻ｉ，从状态ｚ跳跃的中点估计方法如下：步骤１：对于符合跳跃条件的Ｔ，计算在ｋ，ｔ＋Ｔ）状态的预期变化量Ｄ步驟２：令壬＝ｘ＋［ｉＤ／２］．生成平均值为斤⑷＊１■的ｐｏｉｓｓｏｎ数年步骤３：计算实际状态变化量ｗ＝＆__更新状态：又（ｔ＋ｒ）＝Ｘ⑷＝ｘ＋ｃｊ．步骤４：更新时间：ｔｔ＋Ｔ，统计各个网格点处的数据，继续下一次循环．步骤生成个各网格点处的概率密度函数：．表３．６：ＳＤＥ（３．３）在不同参数下Ｑｃ－ｔａｕ－ｌｅａｐｉｎｇ与中点格式的Ｑｃ－ｔａｕ－ｌｅａｐｉｎｇ方法的Ｐ误差及ＣＰＵ运行时间的比较．丁ｔｓｔ〇ｐＮＭｅｔｈｏｄＰ－ｅｒｒｏｒＣＰＵｔｉｍｅ（ｓｅｃ）０．０１２０００００３０Ｑｃ－ｔａｕ－ｌｅａｐｉｎｇ０．０１０１９．９５０．０１２０００００３０Ｑｃ－ｍｉｄ－ｔａｕ－ｌｅａｐｉｎｇ０．００８４１１．３４．０１２０００００６０Ｑｃ－ｔａｕ－ｌｅａｐｉ

ｎｇ０．００７４１２．２１０．０１２０００００６０Ｑｃ－ｍｉｄ－ｔａｕ－ｌｅａｐｉｎｇ０．００５９１６．２８０．０５２０００００３０Ｑｃ－ｔａｕ－ｌｅａｐｉｎｇ０．０４２９１．３１０．０５２０００００３０Ｑｃ－ｍｉｄ－ｔａｕ－ｌｅａｐｉｎｇ０．０２６８１．９２０．０５２０００００６０Ｑｃ－ｔａｕ－ｌｅａｐｉｎｇ０．０３２８３．２００．０５２０００００６０Ｑｃ－ｍｉｄ－ｔａｕ－ｌｅａｐｉｎｇ０．０１２８４．７８〇°－３＊４ｒ

ＡＡ__ｔ■ｉ?? 迷＾

車ｔＱＱｒｕｃｃｅｔｍ

ａｓｕｉｏｄｌｕｌｅｔｔａａｉｏｕｐｎｉｌｎｅｇａｐｉｎｇ

＼°—°５ｙＯＨｔＳ－ｅＣｙ－３－２－１０１２３Ｘ图３．６：ＳＤＥ（３．３）基于中点格式的不变概率密度函数．对于．节中的带加法噪声的一维立方振荡器（.），为了减小ｔａｕ－ｌｅａｐｉｎｇ算法的模拟误差，我们采用了本节提出的中点格式的ｔａｕ－ｌｅａｐｉｎｇ方法．这里我们仍在基础上进行ｔａｕ－ｌｅａｐｉｎｇ模拟模拟在７Ｖ＝１〇〇的网格点上进行状态空间,，Ｄ＝卜３，礼模拟的空间步长是均匀的ｈ＝０.，停止时间＝２０００００，初始状态＝２，选取跳跃步长Ｔ＝０．０５．图３．６展示了中点格式算法的不变概率密度函数与原来的Ｑｃ－ｔａｕ－ｌｅａｐｉｎｇ方法的比较，其中带星号的曲线为中点算法，带圆圈的曲线为（５ｃ－ｔａｕ－ｌｅａｐｉｎｇ方法，红色曲线为真实的平稳密度曲线．由图可见，中点算法曲线更加贴合真实的平稳密度曲线．在不同参数下，我们与Ｑｃ－ｔａＵ－ｌｅａｐｉｎｇ算法进行了１００次数值模拟比较．表．２１中的数据表明中点格式的方法明显缩小了误差但在模拟速度上由于中点格式的算，法增加了计算步骤，因此运行时间会比Ｑｅ－ｔａｕ－ｌｅａｐｉｎｇ方法长．结合表３．１、３．２可看出中点格式的Ｑｃ－ｔａｕ－ｌｅａｐｉｎｇ方法仍然比方法要快．２２

§3.4 带拒绝机制的 tau-leaping 算法（原文摘录）

§．４ｔａｕ－ｌｅａｐｉｎｇ我们的算法在进行步长跳跃时要求在＋内倾向函数不会发生明显变化,．然而在实际的模拟实验中往往会出现Ｐｏｉｓｓｏｎ跳跃次数过大的情况此时显然不,，满足跳跃条件此外跳跃次数过大也会使得质点跳跃时跑出边界从而跳跃无效.,,,．在数值模拟化学系统时也出现过类似的问题，为此ＤａｖｉｄＦ．Ａｎｄｅｒｓｏｎ等人提出了跳跃后检查的机制受此启发为保证算法有效性我们提出了带有拒绝机制的,，ｔａｕ－ｌｅａｐｉｎｇ算法．算法将基于前一次跳跃的成功或失败来自适应的选择ｔ．首先，明确时间步长ｔ计算各个方向的倾向函数其次我们将通过使用泊松随,，机数Ｐｊ（ｒ；〇:，ｔ)？ｐｏｉｓｓｏｎ（％＊ｔ）来近似一段时间内的跳跃次数，这里我们对每个区间段＋的跳跃次数加以上界限制上界的选择我们是根据对质点轨迹的多丨.，次观察得到的．假如生成的ｐｏｉｓｓｏｎ跳跃数小于上界，我们将进行正常的ｔａｕ－ｌｅａｐｉｎｇ跳跃．用知＋１＾＝尤』＋巧.％来近似计算这段时间内的实际位移文（ｔ＋４＝４？假如ｐｏｉｓｓｏｎ跳跃数超过上界我们将选择一个更小的时间步长Ｔ′＜Ｔ使其在＋,，内跳跃次数不会太大并尝试在这个较短的时间段内进行跳跃已知在＋内,.，泊松跳跃的次数为巧．因此，想要计算在＋的跳跃次数片，应该以巧为条件．引理３．４．１令ｙ⑷是一个带有参数Ａ的ｐｏｉｓｓｏｎ过程，且则在已知Ｙ⑷和：Ｋ⑷的条件下，－１"⑷具有二项式分布__

Ｙ（ｓ），ｒ），其中ｒ＝（ｉｔ＿ｓ）／（ｔ＿ｓ）．由引理３．１（证明可见附录），我们可知ｐ｜的分布为二项分布ｂｉｎｏｍｉａｌ＾,？－），其中ｒ＝ｔ′／ｔ．跳跃之后继续更新状态＋巧％，进行下一次循环__因此，对于当前时刻ｔ，质点位置尤＝〇;，上界ｂｏｕｎｄ，我们的方法如下：步骤１：选择满足跳跃条件的ｔ．步驟２：由Ｑ算法计算各个方向的倾向函数（ａ:））．步骤３：生成泊松随机数Ａ’（Ｔ；ａ;，幻？ｐｏｉｓｓｏｎ（ｇ（：ｃ，ｙｆ（；ｒ））＊ｔ）．步骤４：如果，｜Ａ／－ＡＪ＜ｂｏｕｎｄ，更新状态４＝＆＋（Ａ／＊＜＋Ａｆ＊ｆ）；否贝Ｉｊ，ｔ′＝ｍａｘ（ｌ／Ａｆ，ｌ／Ａｆ），ｔ＝ｔ’，生成二项分布的随机数ｆ，Ａｆ？ｂｉｎｏｍａ＾Ａ＾＾Ｔ′／Ｔ），更新状态４＝心＋（Ａｆ＊＜＋Ａｆ＊ｆ）．步骤５：更新时间：ｉｔ＋ｒ，统计各个网格点处的数据，继续下一次循环．表３．７：ＳＤＥ（３．３）在不同参数下ｔａｕ－ｌｅａｐｉｎｇ与带拒绝机制的Ｑｆ－ｔａｕ－ｌｅａｐｉｎｇ的Ｐ误差及ＣＰＵ运行时间的比较．ｒｔｓｔ〇ｐＮＭｅｔｈｏｄ＾－ｅｒｒｏｒＣＰＵｔｉｍｅ（ｓｅｃ）０．０１２０００００３０Ｑ＾ｚ－ｔａｕ－ｌｅａｐｉｎｇ０．０７０９１２．８９０．０１２０００００３０Ｑ＾－ｔａｕ－ｌｅａｐｉｎｇ－ｒｅｊｅｃｔ０．０４５８９．３３０．０１２０００００６０Ｑ＾－ｔａｕ－ｌｅａｐｉｎｇ０．０３００４８．９００．０１２０００００６０Ｑ＾－ｔａｕ－ｌｅａｐｉｎｇ－ｒｅｊｅｃｔ０．０２６８１１．７６０．０５２０００００３０Ｑ＾－ｔａｕ－ｌｅａｐｉｎｇ０．０６７７２．０３０．０５２０００００３０Ｑ＾－ｔａｕ－ｌｅａｐｉｎｇ－ｒｅｊｅｃｔ０．０６９６１．１００．０５２０００００６０ｇ＾ｚ－ｔａｕ－ｌｅａｐｉｎｇ０．０７０９１２．８９０．０５２０００００６０Ｑ＾ｚ－ｔａｕ－ｌｅａｐｉｎｇ－ｒｅｊｅｃｔ０．０４９１４．０５步骤生成个各网格点处的概率密度函数:．０．４０°３５／＼０．３０／＼

，．２５＿／＼〇３０．２０－－－ｔｒｕｅｓｏｌｔｕａｔｕｉｏｎｌｅａｐｉｎｇｗｉｔｈｒｅｊｅｃｔｄｉｓｔｒｉｂｕｔｉｏｎ′／＼．０５￣／＼’、.—３—２—１０１２３Ｘ图３．７：基于带有拒绝机制的Ｑｆｔａｎ－ｌｅａｐｉｎｇ方法进行模拟的不变概率密度函数．我们仍然以带加法噪声的一维立方振荡器模型为例，用ｐｙｔｈｏｎ运行３．１节提出的Ｑｆ－ｔａｕ－ｌｅａｐｉｎｇ方法时我们发现此时质点大部分将会超出边界此时无法进，，行数值模拟．因此，我们采用了本节提出的带有拒绝机制的Ｑｕ－ｔａｕ－ｌｅａＰｉｎｇ方法，成功实现了模拟运行．模拟在ｉＶ＝３０的网格点上进行，状态空间Ｄ＝［－３，］，模拟的空间步长是均匀的／ｉ＝０．２，初始状态＝０．我们取＝２０００００，Ｔ＝０．０５．如图３．７，我们给出了ＳＤＥ（３．３）基于带有拒绝机制的Ｑｆ－ｔａｕ－ｌｅａｐｉｎｇ方法进行模拟的不变概率密度函数曲线与真实解的平稳密度曲线其中黄色曲线为真实解．的不变概率密度函数蓝色线为基于带有拒绝机制的Ｑｆ－ｔａｕ－ｌｅａｐｉｎｇ方法进行模，拟的不变概率密度函数．由图可见我们的算法成功避免了Ｑｆ－ｔａｕ－ｌｅａｐｉｎｇ方法所出现的问题．为了更好的比较误差，我们取了１００次平均，由表格３．７可见，在误差可以接受的范围内，运行时间将Ｑｆ－ｔａｕ－ｌｅａｐｉｎｇ的运行时间将大大缩减．对于其他空间离散算法在运行过程中也会出现相同的问题我们提出的带拒，，绝机制的空间离散算法可以很好的避免此问题．２５

§4.1 扩散项为小参数的 tau-leaping 方法：主要算法（原文摘录）

第四章ＳＤＥ扩散项为小参数的ｔａｕ－ｌｅａｐｉｎｇ方法传统的ｇ方法不适应于扩散项为小参数ｅ的ＳＤＥ问题，针对此类ＳＤＥ，我们提出了尺度化的空间离散ｔａｕ－ｌｅａｐｉｎｇ方法，并且对二维Ｒｉｎｇｄｅｎｓｉｔｙ模型进行了应用由多次试验表明我们的算法对于模拟此类随机微分方程是非常有效的且大大．提高了模拟效率．此外，这种改进的ｔａｕ－ｌｅａｐｉｎｇ算法同样还适应于吸引域比较大的随机扰动Ｌｏｒｅｎｚ振子模型我们在第三节中成功模拟出了此类模型的不变概率密，度函数解决了空间离散ｇ算法不适用的问题，．主要算法§４．１考虑动力系统的长时间行为或非正常行为时往往需要考虑ＳＤＥ扩散项为ｅ，小参数问题假设带有加法噪声的ｎ随机微分方程．ｄＸｔ＝ｆｉ（Ｘｔ）ｄｔ＋ａｄＷｔ，（４．１）其中Ｗ＊是ｎ维的白噪声，：Ｒ＂４和ａ：Ｒ＂４分别为漂移项和常系数扩散项，（Ｔ《１为小参数．在空间离散方法中方法是二阶精度的但是当ａ非常小时方法显然不，，，再适用对于方法计算精度又达不到我们的要求因此我们考虑使用Ｑｆ方.，.，法．为保证Ｑ严方法是二阶精确的，需要满足ａ２－｜／／（〇；）叫＞０．因此，要使得Ｑｆ满足二阶精度相应的需要模拟的网格长度非常小这就会导致运算速度减慢本，，．节我们提出使用ｔａｕ－ｌｅａｐｉｎｇ方法选定合适的ｔ通过生成泊松数计算一段时间内，，的近似跳跃次数来实现多步位移进而提高运算速度算法的主要思想和第三章中，．提出的ｔａｕ－ｌｅａｐｉｎｇ相似主要区别在于空间步长的选择以及在不变概率密度函数，，的实现上有所不同．例如：对于当前时刻ｔ，质点位置尤＝（；ｒ，ｙ），ｎ＝２．步骤１：选择符合跳跃条件的时间ｔ，根据算法计算（４．１）的风，然后在加细网格＝ｈ／ｓｃａ〖ｅ上生成各方向的倾向函数：，吆〇ｒ））＝ｌ／／ｉ＿（ｍａｘ（＂ｉ，０）＋ｑ（Ｘｉ，ｙｊ（ｘ））＝ｌ／／ｉｎｅｕ）（－ｍｍ（／ｉｉ，０）＋Ｍｉ／ｈｎｅｗ），９（工ｉ，Ｚ／ｊ（尤））＝ｑ｛ｘｉ，ｙｊ（ｘ））＝ｌ／Ｚｉｎｅ＾＾ｍｉｎ＾ｊＯ）＋Ｍｊ／ｈｎｅｗ）．步骤２：生成ｐｏｉｓｓｏｎ随机数Ａ?? ｐｏｉｓｓｏｎＷ；＾,％.)！＂），ｎ＝（ｌ，２，３，４)？步驟３：更新状态：又（Ｚ＋Ｔ）＝文⑷＋（Ａｌ—Ａ２＋■＾３—■＾４）＊／ｉ．步驟４：更新时间：ｔ＝ｉ＋Ｔ．步骤５：在原来的网格／ｉ上统计数据，生成近似的不变概率密度函数．Ａｌｇｏｒｉｔｈｍ２扩散项为小参数的二维ＳＤＥ的Ｑｆ－ｔａｕ－ｌｅａｐｉｎｇ算法输入：状态下界ｌｏｗｘ，ｌｏｗｙ，样本容量Ｓｐ，网格数量Ｎ，停止时间ｔ＊，步长ｔ，零矩阵ｄａｔａ２ｘ２计数ｍｉｍ，输出：不变概率密度ｄａｔａ１：ｆｕｎｃｔｉｏｎ２Ｄ－ｔａｕ－ｌｅａｐｉｎｇ（ｌｏｗｘ，ｌｏｗｙ，Ｓｐ，Ｎ，ｔ＊，ｘ〇，ｙ〇，ｒ）２：ｔ？＜—０，ｈ－＜—Ｓｐ／Ｎ，ｈ＊＜—ｈ／ｓｃａｌｅ，ｘｎ？＜—０，ｙｎ？＜—０；３：ｘ—ｌｏｗｘ－ｈ＊／２．０＋ｒｏｕｎｄ（（ｌ－ｌｏｗｘ＋ｈ＊／２．〇）／ｈ＊）ｈ＊；４：ｘ＜ｒ－ｌｏｗｙ－ｈ＊／２．０＋ｒｏｕｎｄ（（ｌ－ｌｏｗｙ＋ｈ＊／２．０）／ｈ＊）ｈ＊；５：ｗｈｉｌｅｔ＜ｔ＊ｄｏ利用Ｚ方法计算６：Ｑ〗ｑｉ，ｑ２，；７：０．５ｍａｘ（ｅｐｓ－ａｂｓ（／ｉｉ）ｈ＊，０），ｉ１，２；８:—ｒ（士ｍａｘＯ＾Ｏ）／!! ＊＋Ｍｌｌｙ／（ｈ＊ｈ＊））；９：ｑ３，ｑ４—ｔ（士ｍａｘ（＂２，０）／ｈ＊＋Ｍ２２＂ｈ＊ｈ＊））；１０：计算四个方向的泊松随机数：Ａｉｔｐｏｉｓｓｏｎｈ），（ｉ—１，２，３，４）；１１：更新状态：ｘ？＜—ｘ＋（Ａｉ—Ａ２）ｈ＊；ｙ？＜—ｙ＋（Ａ３—Ａ４）ｈ＊；１２：记录质点落入的网格点：ｘｎ—ｒｏｕｎｄ（（ｘ－ｌｏｗｘ＋ｈ／２．０）／ｈ）；１３：ｙｎ＜ｒ－ｒｏｕｎｄ（（ｙ－ｌｏｗｙ＋ｈ／２．０）／ｈ）；１

```
4：ｉｆｘｎ＞＝１ａｎｄｘｎ＜＝Ｎａｎｄｙｎ＞＝１ａｎｄｙｎ＜＝Ｎｔｈｅｎ１５：ｄａｔａ（ｘｎ，ｙｎ）—ｄａｔａ（ｘｎ，ｙｎ）＋1；１６：ｎｕｍ?＜—ｎｕｍ＋１；１７：ｅｎｄｉｆ１８：ｔ—ｔ＋ｔ；１９：ｅｎｄｗｈｉｌｅ２０：计算概率密度函数ｄａｔａ—ｄａｔａ／（ｈｘｈｘｎｕｍ）．２１：ｅｎｄｆｕｎｃｔｉｏｎ２７
```