

On the stochastic simulation of the Master Equation

Elena Grecu

18 February 2024

1 Introduction

In physics and chemistry, a Master Equation (ME) is a set of differential equations that describe the time evolution of a system defined on a discrete set of states. These equations describe the probabilities of the associated system of transitioning from a given state into another. They are used to model various phenomena, such as chemical reactions, population dynamics, etc. While in principle, all ordinary differential equations (ODEs) are equivalent to a ME, and solvable through probabilistic methods specific to MEs, the benefits of such solutions are limited mainly to ODEs describing the evolutions of populations. In this paper, we show how to cast an ODE into a ME framework and solve it probabilistically (when beneficial). We provide a simple but intuitive proof of why the two are equivalent, and why the associated probabilistic solutions work. Most of the initial work on probabilistic methods to solve the ME was carried out by physicist Daniel Gillespie in the 1970s. His studies, while rigorous, made use of physical arguments (borrowed from molecular physics) to prove the validity of the approach. In this note, we exclusively rely on mathematical arguments. We also provide a simple example of a population dynamics ODE and its equivalent ME, and solve the ME using probabilistic methods.

2 The Master Equation

Let us assume a system of equations that describes the chemical reactions of two species, A and B, that can react to form a third species, C. The system of equations is given by:

$$\frac{d[A]}{dt} = -k_1[A][B] + k_2[C] \tag{1}$$

$$\frac{d[B]}{dt} = -k_1[A][B] + k_2[C] \tag{2}$$

$$\frac{d[C]}{dt} = k_1[A][B] - k_2[C] \tag{3}$$

where k_1 and k_2 are the rate constants of the reactions, and $[A]$, $[B]$, and $[C]$ are the concentrations of the three species. The above system of equations can be solved using standard ODE methods. However, it is beneficial to analyze the system in a probabilistic framework. Specifically, given that a molecule of C is formed when a molecule of A and

a molecule of B react, we can calculate the probability of the system transitioning from a state with n_A molecules of A and n_B molecules of B to a state with $n_A - 1$ molecules of A, $n_B - 1$ molecules of B, and $n_C + 1$ molecules of C. This is the essence of the Master Equation. Formally, the Master Equation is given by:

$$\frac{dP(S)}{dt} = T(S) \cdot P(S) \quad (4)$$

where $P(S)$ is the probability of the system being in a given state S , and $T(S)$ is a matrix that describes the transition probabilities. While straightforward to define for the above system of equations, matrix $T(S)$ can be quite complex for more complicated systems. Therefore, instead of explicitly defining $T(S)$, we will outline a simulation algorithm, called the Stochastic Simulation Algorithm (SSA), that enables the derivation of a numerical solution of the Master Equation.

3 The Stochastic Simulation Algorithm

For simplicity let us assume that we have a system with only one species that undergoes a process resulting in a concentration change. The process can be described by the following ODE:

$$\frac{d[A]}{dt} = -k[A] \quad (5)$$

where k is the rate constant of the process and $[A]$ is the concentration of the species. Assuming that the concentration $[A]$ is defined as the number of moles of A divided by the volume of the system V , then the number of molecules of A in the system is given by $n_A = N_A \cdot [A] \cdot V$, where N_A is Avogadro's number. We can then calculate the probability of the system transitioning from a state with n_A molecules of A to a state with $n_A - 1$ molecules of A by using the following formula:

$$P(n_A \rightarrow n_A - 1) = k \cdot n_A \cdot \Delta t \quad (6)$$

In the original formulation of Gillespie [1], the probability of transition $k \cdot n_A \cdot \Delta t$ was formulated as a hypothesis supported by physical arguments. However, one can calculate the expected number of molecules at time $t + \Delta t$ from the number of molecules at time t and the transition probability $p = n_A \cdot k \cdot \Delta t$ and show that the expected number of molecules at time $t + \Delta t$ is numerically consistent with a finite difference approximation of the ODE. Specifically,

$$E[n_A(t + \Delta t)] = n_A(t) \cdot (1 - p) + (n_A(t) - 1) \cdot p \quad (7)$$

where $E[n_A(t + \Delta t)]$ is the expected number of molecules at time $t + \Delta t$, and $n_A(t)$ is the number of molecules at time t . The above equation can be expanded to:

$$E[n_A(t + \Delta t)] = n_A(t)(1 - k \cdot \Delta t) \quad (8)$$

which is equivalent to finite difference approximation of 5. Therefore, the transition probability $k \cdot n_A \cdot \Delta t$ is consistent with the associated ODE. This is a simple but intuitive proof of why the Master Equation and the ODE are equivalent. In addition to the probability of transition, to be able to simulate the time evolution of Eq. 5, one needs to determine the time to the next transition, τ . Variable τ is a random variable, and generating values for it requires knowledge of its associated probability distribution. Since we do not know whether τ is small or large, we can split the time interval $[0, \tau]$ into N small intervals of length $\frac{\tau}{N}$, such as the probability of the system not undergoing a transition within a $\frac{\tau}{N}$ interval is $1 - k \cdot n_A \cdot \frac{\tau}{N}$. Then, the probability of the system transition happening at time $t + \tau$ is given by:

$$P(n_A \rightarrow n_A - 1) = (1 - k \cdot n_A \cdot \frac{\tau}{N})^{N-1} \cdot k \cdot n_A \cdot \frac{\tau}{N} \quad (9)$$

For large N , the above equation can be approximated by:

$$P(n_A \rightarrow n_A - 1) = e^{-k \cdot n_A \cdot \tau} \cdot k \cdot n_A \cdot \frac{\tau}{N} \quad (10)$$

because

$$\lim_{N \rightarrow \infty} (1 - k \cdot n_A \cdot \frac{\tau}{N})^{N-1} = e^{-k \cdot n_A \cdot \tau} \quad (11)$$

The value in Eq. 10 is the probability of the system transition in the time interval $[t + (N - 1) \cdot \frac{\tau}{N}, t + \tau]$. If we divide it by $\frac{\tau}{N}$, we get the probability the density of probability of the transition happening within $\frac{\tau}{N}$ from time $t + \tau$, i.e.:

$$p(n_A \rightarrow n_A - 1) = e^{-k \cdot n_A \cdot \tau} \cdot k \cdot n_A \quad (12)$$

This is a probability density function of the type $p(x) = \lambda \cdot e^{-\lambda x}$, where $\lambda = k \cdot n_A$, which is generally known as an exponential distribution [2]. The cumulative distribution function of the exponential distribution is given by [2]:

$$F(x) = 1 - e^{-\lambda x} \quad (13)$$

To generate a value for τ , we can generate a random number r from a uniform distribution between 0 and 1, and solve Eq. 13 for τ , i.e $r = 1 - e^{-\lambda \tau}$, which gives $\tau = -\frac{1}{\lambda} \cdot \ln(1 - r)$.

Equation 12 may be generalized to systems of equations involving multiple species. Before we do that, let us first define the concept of propensity function. The propensity function is a function that describes the probability of a reaction happening in a very small given time interval dt normalized by dt , Gillespie [3]. For example, for equation (6), the propensity function is $k \cdot n_a$, while for the discrete equations associated with the system of given by Eqs. (1-3), the propensity function of the first reaction is given by $k_1 \cdot n_A \cdot n_B$, and the propensity function of the second reaction is given by $k_2 \cdot n_C$. For a general system with multiple propensity functions, the probability of the system having a transition in the interval $[t + (N - 1) \frac{\tau}{N}, t + \tau]$, where τ and N are defined as in the single species case, is equal to

the probability of the system not having any of the possible transitions in the first $N - 1$ intervals, and having a transition in the last interval, i.e.

$$P(S \rightarrow S') = \prod_{i=1}^M (1 - \alpha_i \cdot \frac{\tau}{N})^{N-1} \cdot \sum_{i=1}^M \alpha_i \cdot \frac{\tau}{N} \quad (14)$$

where α_i is the propensity function of the i -th reaction out of M possible reactions, and S and S' are the states of the system before and after the transition. When N is large, the above equation can be approximated by:

$$P(S \rightarrow S') = e^{-\sum_{i=1}^M \alpha_i \cdot \tau} \cdot \sum_{i=1}^M \alpha_i \cdot \frac{\tau}{N} \quad (15)$$

Therefore, τ is an exponentially distributed random variable with parameter $\lambda = \sum_{i=1}^M \alpha_i$. We can, therefore, as explained above, generate a value for τ by generating a random number r from a uniform distribution between 0 and 1, and solving Eq. 13 for τ using $\lambda = \sum_{i=1}^M \alpha_i$. However, merely generating a value for τ is not enough to simulate the time evolution of the system. We also need to determine which transition happens at time $t + \tau$. This can be done by generating an additional random number r from a uniform distribution between 0 and 1, and determining which transition happens based on the magnitude of the propensity functions. Specifically, we can calculate $f_j = \frac{\sum_{i=1}^j \alpha_i}{\sum_{i=1}^M \alpha_i}$, and then determine the minimum value of j such that $r < f_j$. Since $f_N = 1$, there is always a value of j such that the above condition is satisfied. We can then determine that the transition that happens is the j -th transition. The SSA algorithm can be summarized as follows:

1. Initialize the system at time $t = 0$.
2. Generate a uniformly distributed random number r_1 between 0 and 1 and determine τ from $F(\tau) = r_1$, where $F(\tau)$ is the cumulative distribution function of the exponential distribution with parameter $\lambda = \sum_{i=1}^M \alpha_i$.
3. Generate a (uniformly distributed) value for r_2 and determine which transition takes place from $r_2 < f_j$.
4. Update the system by applying the transition that happens at time $t + \tau$.
5. $t = t + \tau$.
6. Repeat steps 2-5 until the desired time is reached.

4 Example

We will now provide an example of the SSA algorithm. We will consider a system of equations known as the Lotka-Volterra equations, which describe the dynamics of a predator-prey

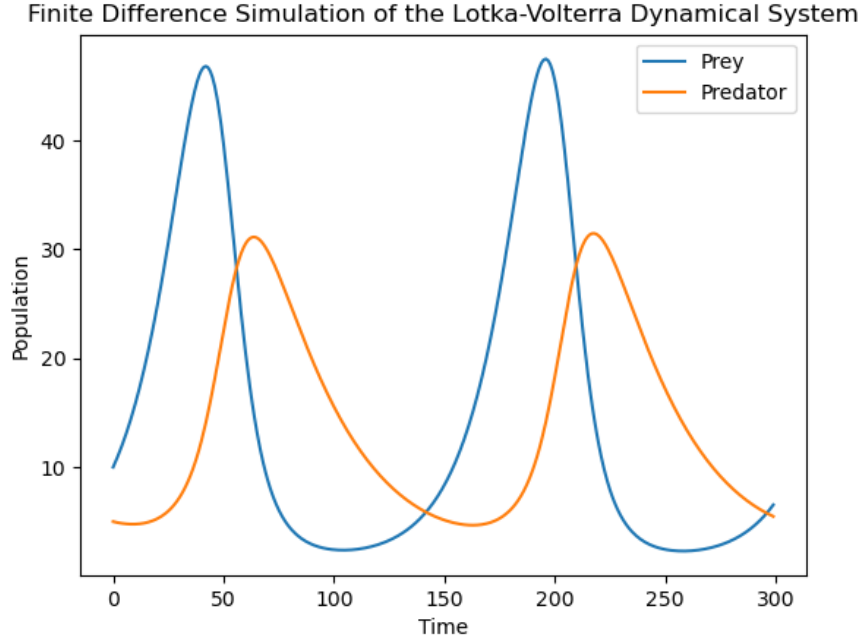


Figure 1: A simulation of the Lotka-Volterra equations using a finite-difference solution.

system. The equations are given by:

$$\frac{dX}{dt} = \alpha X - \beta XY \quad (16)$$

$$\frac{dY}{dt} = \delta XY - \gamma Y \quad (17)$$

where X and Y are the populations of the prey and predator, and α , β , δ , and γ are the rate constants of the system. We will assume that the system is initialized with $X = 10$ and $Y = 5$. We will also assume that $\alpha = 0.07$, $\beta = 0.005$, $\delta = 0.002$, and $\gamma = 0.03$. A simulation of the system using a finite-difference solution of the system of equations is shown in Fig. 1. Before presenting the SSA implementation and the results, it is worth noting that the SSA algorithm is not likely to be accurate for problems involving small populations. This is because as the population decreases, the discrete decrement of the algorithm may result in completely extinct species, while as apparent in the solution shown in Fig. 1, neither species goes extinct, although their populations can become very low. To circumvent this situation, we apply the SSA algorithm to a transformed, but mathematically equivalent to the original, system of Lotka-Volterra equations. Specifically, we transform the original system into one of the type:

$$\frac{dX'}{dt'} = \alpha' X' - \beta' X'Y' \quad (18)$$

$$\frac{dY'}{dt'} = \delta' X'Y' - \gamma' Y' \quad (19)$$

where X' and Y' are scaled up populations of the prey and predator, and α' , β' , δ' , and γ' are the associated rate constants of the system. That is $X' = X \cdot s_f$ and $Y' = Y \cdot s_f$, where s_f is a scaling factor. We use $s_f = 100$. To make the system equivalent to the original, we need to set $\alpha' = \alpha/s_f$, $\beta' = \beta/s_f^2$, $\delta' = \delta/s_f^2$, $\gamma' = \gamma/s_f$, and $t' = t \cdot s_f$. Another observation is that a pray-predator interaction event has different impacts on the pray and predator populations. Therefore, we use $\beta' \cdot X' \cdot Y'$ as propensity function in the system and update the system using $X' = X' - 1$ and $Y' = Y' + \frac{\delta}{\beta}$ to make the transitions consistent with the original system of equations. The SSA algorithm is implemented in the following Python code:

```
def get_tau_and_transition(X,Y,alpha,beta,gamma):
    tot_prop=alpha*X/s_f+beta*X*Y/s_f**2+gamma*Y/s_f
    r=np.random.rand()
    tau=-np.log(r)/tot_prop
    r=np.random.rand()
    if r<alpha*X/s_f/tot_prop:
        return 1,tau
    elif r<(alpha*X/s_f+beta*X*Y/s_f**2)/tot_prop:
        return 2,tau
    else:
        return 3,tau

def update_state(X,Y,alpha,beta,gamma,delta):
    r,tau=get_tau_and_transition(X,Y,alpha,beta,gamma)
    if r==1:
        return max(2,X+1),max(2,Y),tau
    elif r==2:
        return max(2,X-1),Y+1*delta/beta,tau
    else:
        return X,max(2,Y-1),tau
```

The outcome of a simulation using the SSA is shown in Fig. 2. As apparent, the SSA simulation is in good agreement with that produced by the finite-difference solution. Nevertheless, it should be mentioned that the SSA algorithm is computationally more expensive than the finite-difference solution, and is not the best option for problems that do not involve conservation of populations or mass. Specifically, the SSA simulation may drift away from true solution of the system without any objective to determine whether a drift occurred. In systems involving conservation of populations or mass, drifts cannot occur as the number of individuals or their mass is preserved. In such cases, the SSA algorithm may be a better option than numerical deterministic solutions, especially if the interactions among species are complex and involve some randomness. An example of such a system is the stochastic coalescence of cloud droplets, Gillespie [4]. This is a system of equations that describes the time evolution of the number of cloud droplets of different sizes. The system is a good candidate for the SSA algorithm because the total mass is conserved, i.e. two cloud droplets of mass m_i and m_j can coalesce to form a cloud droplet of size $m_i + m_j$. Deterministic solutions of the system are possible but complicated as they involve the discretization of the

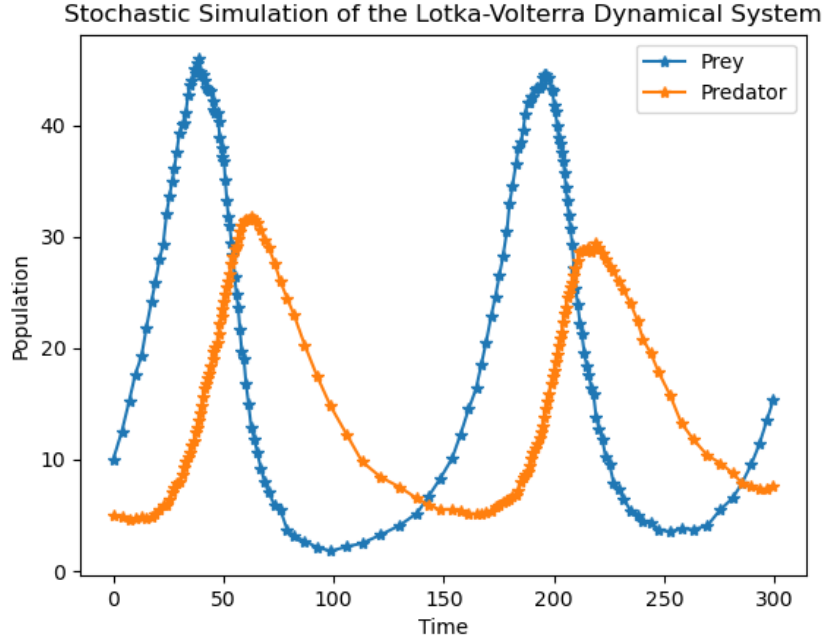


Figure 2: A simulation of the Lotka-Volterra equations using the SSA algorithm.

mass of the cloud droplets in a large number of bins and then accounting for the interactions among the bins. The complication stems in the fact that the truncation errors may lead to unphysical solutions (e.g. negative mass in some bins). Moreover, additional processes such as a fragmentation of cloud droplets, which are not easy to model deterministically, can be easily incorporated in the SSA algorithm.

5 Conclusions

In this note, we described the Master Equation associated with some types of ODEs and a Stochastic Simulation Algorithm to solve them. We provided a simple but intuitive proof of why the Master Equation and the ODE are equivalent, and why the associated probabilistic solutions work. We also provided a simple example of a population dynamics system of ODEs, and solved it using the SSA.

References

- [1] Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. The Journal of Physical Chemistry, 81(25), 2340-2361.
- [2] Dekking FM, Kraaikamp C, Løpuhaä HP, Meester LE. A Modern Introduction to Probability and Statistics: Understanding why and how. London: springer; 2005.

- [3] Gillespie DT. Stochastic simulation of chemical kinetics. *Annu. Rev. Phys. Chem.*. 2007 May 5;58:35-55.
- [4] Gillespie, D. T., 1975: An Exact Method for Numerically Simulating the Stochastic Coalescence Process in a Cloud. *J. Atmos. Sci.*, 32, 1977-1989, [https://doi.org/10.1175/1520-0469\(1975\)032](https://doi.org/10.1175/1520-0469(1975)032).