CONDITIONAL MONTE CARLO FOR REACTION NETWORKS

DAVID F. ANDERSON* AND KURT W. EHLERT†

Abstract. Reaction networks are often used to model interacting species in fields such as biochemistry and ecology. When the counts of the species are sufficiently large, the dynamics of their concentrations are typically modeled via a system of differential equations. However, when the counts of some species are small, the dynamics of the counts are typically modeled stochastically via a discrete state, continuous time Markov chain.

A key quantity of interest for such models is the probability mass function of the process at some fixed time. Since paths of such models are relatively straightforward to simulate, we can estimate the probabilities by constructing an empirical distribution. However, the support of the distribution is often diffuse across a high-dimensional state space, where the dimension is equal to the number of species. Therefore generating an accurate empirical distribution can come with a large computational cost.

We present a new Monte Carlo estimator that fundamentally improves on the "classical" Monte Carlo estimator described above. It also preserves much of classical Monte Carlo's simplicity. The idea is basically one of conditional Monte Carlo. Our conditional Monte Carlo estimator has two parameters, and their choice critically affects the performance of the algorithm. Hence, a key contribution of the present work is that we demonstrate how to approximate optimal values for these parameters in an efficient manner. Moreover, we provide a central limit theorem for our estimator, which leads to approximate confidence intervals for its error.

Key words. Monte Carlo, continuous time Markov chain, chemical master equation, nonparametric density estimation, reaction networks

AMS subject classifications. 65C05, 60J28, 62G07

1. Introduction.

Systems of interacting species appear often in nature. To better understand the dynamics of such systems, we can model them as reaction networks with deterministic or stochastic dynamics [6, 20, 27, 49]. If the counts of the constituent species is high, then the dynamics are commonly modeled by a system of differential equations [6, 16, 49]. However, if the count of any species is small, then a stochastic model with a discrete state space is more appropriate [5, 6, 34, 41, 46, 49].

Since the amount of each species is necessarily nonnegative and discrete, the state space of the stochastic process is a subset of $\mathbb{Z}^d_{\geq 0}$, where d is the number of species types. Let ν be the distribution of the initial state, and suppose we are interested in the distribution of the state of the process at some fixed time $t > 0$. That is, if $X(t)$ is the state of the process at time t, then we would like to know the value of

$$p_t^\nu(x) \stackrel{\text{def}}{=} P_\nu(X(t) = x),\ x \in \mathbb{Z}^d_{\geq 0}.$$

In general, finding the exact values of $p_t^\nu(\cdot)$ is extremely difficult. More precisely, the authors are not aware of any general class of models for which p_t^ν can be solved for explicitly, with the exception of linear, or first-order, models [25] or models that admit a special Poisson structure [7]. However, there are many numerical methods that give an estimate. One type of approach is to approximately solve Kolmogorov's forward equation, which is called the chemical master equation (CME) in much of the biology and chemistry literature. The CME can be written as

$$\frac{d}{dt}p_t^\nu(x) = \sum_{r=1}^{R}\left[p_t^\nu(x-\zeta_r)\lambda_r(x-\zeta_r) - p_t^\nu(x)\lambda_r(x)\right],\ x \in \mathbb{Z}^d_{\geq 0}, \tag{1.1}$$

*Department of Mathematics, University of Wisconsin-Madison (anderson@math.wisc.edu)
†Department of Mathematics, University of Wisconsin-Madison (kehlert@math.wisc.edu)

where R is the number of reactions in the system, $\lambda_r : \mathbb{Z}^d_{\geq 0} \to \mathbb{R}_{\geq 0}$ is the intensity (or propensity) function for the rth reaction, $\zeta_r \in \mathbb{Z}^d$ gives the net change in the counts of the species due to an occurrence of the rth reaction, and the initial distribution $p^\nu_0(\cdot)$ is given by ν. See section 2 for the precise specification of the model, including terminology.

For most models of interest, solving (1.1) entails solving a high-dimensional (often infinite-dimensional) system of linear ordinary differential equations. Solving such a system directly is almost always very difficult, so there has been a considerable amount of research devoted to the development of fast and accurate approximate algorithms. The general approach for many such algorithms is to first truncate the state space of the system to a smaller subset. This truncation makes solving the problem computationally feasible, at the cost of introducing a controllable error to the solution. After truncation, the new system of ODEs must be solved.

There is currently a wide variety of methods for performing both the truncation step and solution step. In particular, there is the finite state projection algorithm [36, 47], the uniformization method [13], sliding window methods [24, 50], the sparse grid method [23], the radial basis function approximation [29], a class of spectral methods [15, 26], and methods that specialize to systems with multiple scales [9, 12, 31, 32, 39]. Moreover, there are tensor methods [28, 44, 48] that represents the truncated CME with tensors.

As an alternative to approximating (1.1) directly via the methods above, we can take a Monte Carlo approach. That is, we can generate n independent and identically distributed (i.i.d.) realizations of the process X, denoted by $\{X_i\}_{i=1}^n$, and use the Monte Carlo estimator

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(X_i(t) = x) \approx \mathbb{E}_{\nu,0}\left[\mathbb{1}(X(t) = x)\right] = p^\nu_t(x), \tag{1.2}$$

where $\mathbb{E}_{\nu,0}$ is the expectation under the initial distribution ν and starting time of zero. By the strong law of large numbers, the approximation becomes an equality as n goes to infinity.

To utilize the above estimator, we need to simulate exact realizations of the process X over the time interval $[0, t]$, and there are many methods to choose from. In particular, there is the Gillespie algorithm [18], the next reaction method [17], and the modified next reaction method [1], which are all straightforward to implement and often have similar efficiency. For our numerical results in the later sections, we used the modified next reaction method.

One drawback of using the Monte Carlo estimator (1.2) to approximate the solution to the CME (1.1) is that huge numbers of simulations are generally required to achieve a high level of accuracy. That said, the Monte Carlo estimator has at least two distinct advantages when compared against the methods that approximately solve the CME directly: it is very simple to implement and it is substantially less sensitive to the dimension of the state space.

There are two natural ways to improve upon a Monte Carlo estimator. The first way is to decrease the time required to generate realizations of the random samples (i.e., the process X in our case). Lowering the time required to generate paths of the processes that we are interested in has been an active area of research for almost two decades [1, 17, 33, 35, 40, 45]. Moreover, researchers have also designed efficient algorithms that generate approximate paths that trade some accuracy for speed [2, 4, 10, 11, 14, 19, 22, 42].

The second way to improve upon a Monte Carlo estimator, and the focus of this article, is to instead lower the variance of the estimator itself. There are many broadly applicable variance reduction techniques, including coupling methods, control variates, stratified sampling, antithetic random variables, quasi-Monte Carlo, and conditional Monte Carlo [21, 38].

In this paper, we utilize a form of conditional Monte Carlo to reduce the variance. Define $\mathbb{E}_{\nu,s}[f(X(t)]$ as the expectation of $f(X(t))$ taken with respect to the initial state distribution ν and starting time $0 \leq s \leq t$. That is, $P(X(s) = x) = \nu(x)$. If ν is a point-mass distribution at $y \in \mathbb{Z}^d_{\geq 0}$, then we write $\mathbb{E}_{y,s}[f(X(t))]$. Fix $h \in [0, t]$, then

$$\begin{aligned} p^\nu_t(x) &= \mathbb{E}_{\nu,0}\left[\mathbb{1}(X(t) = x)\right] \\ &= \mathbb{E}_{\nu,0}\left[\mathbb{E}_{\nu,0}\left[\,\mathbb{1}(X(t) = x)|\, X(t-h)\right]\right] \\ &= \mathbb{E}_{\nu,0}\left[\mathbb{E}_{X(t-h),t-h}\left[\mathbb{1}(X(t) = x)\right]\right] && \text{(Markov property)} \\ &= \lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{X_i(t-h),t-h}\left[\mathbb{1}(X(t) = x)\right], \text{ a.s.} && \text{(strong law of large numbers)} \end{aligned} \tag{1.3}$$

where the $\{X_i(t-h)\}_{i=1}^n$ are i.i.d. realizations of $X(t-h)$. A natural estimator for the right hand side of the above equation is

$$\hat{p}^\nu_t(x; n, m, h) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m}\sum_{j=1}^{m}\mathbb{1}(X_{ij}(t) = x), \tag{1.4}$$

where we generate the X_{ij} in the following manner:

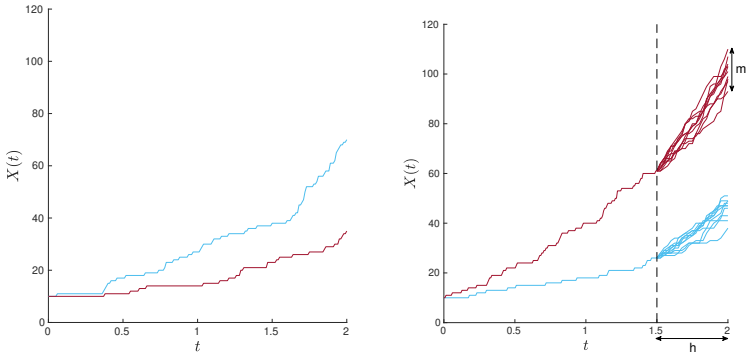
- simulate n independent realizations of the process X over the time interval $[0, t-h]$, each with an initial value determined by ν, and denote the ith realization by X_i,
- for each $i \in \{1, \ldots, n\}$, generate m conditionally independent realizations over the time interval $[t-h, t]$, each of which has initial state $X_i(t-h)$. Denote the jth such realization by X_{ij}.

Note that for each $j \in \{1, \ldots, m\}$, the process X_{ij} is equal to X_i over the interval $[0, t-h]$. See Figure 1.

Since $\{X_{i_1 j}(t)\}_{j=1}^m$ and $\{X_{i_2 j}\}_{j=1}^m$ are independent for $i_1 \neq i_2$, the strong law of law numbers implies that with probability one we have

$$\lim_{n\to\infty} \hat{p}^\nu_t(x; n, m, h) = \mathbb{E}_{\nu,0}\left[\frac{1}{m}\sum_{j=1}^{m}\mathbb{1}(X_{ij}(t) = x)\right] = p^\nu_t(x).$$

Hereafter we will refer to the original estimator (1.2) as *classical Monte Carlo*, and the new estimator (1.4) as *conditional Monte Carlo*. The conditional Monte Carlo estimator has two unspecified parameters, denoted m and h. The number of branches is determined by m, and the time at which branching occurs is controlled by h. If m and h are fixed, then the remaining parameter n is simply chosen large enough such that the estimator's variance is below some desired threshold. If $m = 1$, $h = 0$, or $h = t$, then the conditional and classical Monte Carlo estimators are the same. If $m > 1$ and $h \in (0, t)$, then for the same computational cost as classical Monte Carlo, the conditional Monte Carlo estimator obtains more observations of $X(t)$. We would like to choose the values of m and h such that, in some sense, our new estimator is more efficient than classical Monte Carlo. In section 3, we provide an algorithm for finding optimal values of m and h, which is the key contribution of this article.



(a) Two independent realizations of the process over the time interval $[0, 2]$.

(b) Two independent realizations of the process generated over $[0, 1.5]$. Each is then followed by m conditionally independent "branches" simulated over $[1.5, 2]$.

FIG. 1. *Paths generated for the birth model* $X \to 2X$.

The remainder of the article is organized in the following way. In section 2, we define the continuous time Markov chain model of reaction networks. Then in section 3, we present an algorithm for finding the optimal values of m and h, and also the full algorithm, Algorithm 3.2, for the implementation of the conditional Monte Carlo estimator. Next, in section 4, we give numerical results demonstrating the order of magnitude improvement that can be obtained with the use of conditional Monte Carlo in the current context. In section 5, we derive a central limit theorem for the error of the conditional Monte Carlo estimator and then test it on examples. Finally, in section 6, we summarize our results and suggest ideas for future work. The proofs of the main results are in Appendix A. The supplementary material contain more figures related to numerical results. An example MATLAB implementation of the conditional Monte Carlo algorithm is at https://github.com/kehlert/conditional_monte_carlo_example.

2. Mathematical model. Suppose our reaction network has d types of species and R reactions. For $1 \leq r \leq R$,

(i) we will denote by ζ_r the reaction vector for the rth reaction, meaning that if the rth reaction occurs at time t, and the process is currently in state $x \in \mathbb{Z}^d_{\geq 0}$, then the new state becomes $x + \zeta_r$;

(ii) we will denote by $\lambda_r : \mathbb{Z}^d_{\geq 0} \to [0, \infty)$ the intensity, or propensity, function of the rth reaction.

A standing assumption is that $\lambda_r(x) = 0$ if $x + \zeta_r \notin \mathbb{Z}^d_{\geq 0}$, which preserves the non-negativity of the components. We let X be a continuous time Markov chain (CTMC) whose transition rate from state x to x' is

$$q(x, x') = \sum_{r=1}^{R} \lambda_r(x) \mathbb{1}(x' - x = \zeta_r).$$

Hence, X is a Markov process with infinitesimal generator $Af(x) = \sum_{r=1}^{R} \lambda_r(x)(f(x + \zeta_r) - f(x))$. We will denote our process by X, so that $X(t) \in \mathbb{Z}^d_{\geq 0}$ is the vector whose ith component gives the count of species i at time $t \geq 0$.

The most common choice of intensity function is stochastic mass action kinetics. Suppose that we require y_i copies of species i for the rth reaction to occur. Then we say that λ_r has stochastic mass action kinetics if

$$\lambda_r(x) = \kappa_r \prod_{i=1}^{d} \frac{x_i!}{(x_i - y_i)!} \mathbb{1}(x_i \geq y_i), \tag{2.1}$$

for some $\kappa_r > 0$, which is called the rate constant of the reaction. For example, for the reaction $2A + B \to A + C$, where A, B, and C are the species types in our model system, the reaction vector is $(-1, -1, 1)^T$ and $y = (2, 1, 0)^T$, in which case $\lambda_r(x) = \kappa_r x_1(x_1 - 1)x_2$, where we have ordered the species alphabetically.

None of our theoretical results assume that the λ_r has the above mass action form, but the models we tested do use it unless otherwise noted.

One well–known representation for the stochastic process X is the *random time change representation* of Thomas Kurtz [5, 6, 30]

$$X(t) = X(0) + \sum_{r=1}^{R} Y_r \left(\int_0^t \lambda_r \left(X(s) \right) \, ds \right) \zeta_r, \tag{2.2}$$

where $X(0)$ is the initial state and the Y_r are independent unit-rate Poisson processes. We will make use of the above representation in some of our proofs.

2.1. Examples. In the subsequent sections, we intersperse numerical results, and below is a list of all the example models we used. The species to the left of the arrows are the reactants (giving the counts of the species consumed in the reaction), and those to the right are the products. The numbers above the arrows are the rate constants κ_r. Unless otherwise noted, for every model and reaction we define the intensities λ_r with (2.1).

(i) *Birth*
The initial state is $X(0) = 10$ and $t = 2$. The single reaction is

$$X \xrightarrow{1} 2X.$$

Following (2.1), the rate of the reaction is $\lambda(x) = x$.

(ii) *Birth–Death*
The initial state is $X(0) = 100$ and $t = 2$. There are two reactions

$$\emptyset \xrightarrow{50} X, \; X \xrightarrow{1} \emptyset.$$

Following (2.1), the rates of the reactions are $\lambda_1(x) = 50$, and $\lambda_2(x) = x$, respectively.

(iii) *Lotka–Volterra*
This model is also often called the predator-prey model. The initial state is $A(0) = 200$ and $B(0) = 100$. We set $t = 4$. The reactions are

$$A \xrightarrow{2} 2A, \; A + B \xrightarrow{0.01} 2B, \; B \xrightarrow{2} \emptyset.$$

Following (2.1), and after ordering the species as (A, B), the rates of the reactions are $\lambda_1(x) = 2x_1$, $\lambda_2(x) = 0.01x_1x_2$, and $\lambda_3(x) = 2x_2$, respectively.

(iv) *Dimerization*

In this model, $mRNA$ is translated into the protein P, which then dimerizes into D, and the dimer D accumulates over time. The initial state for every species is zero except for $G(0) = 1$. We set $t = 1$. The reactions are

$$G \xrightarrow{25} G + mRNA,\ mRNA \xrightarrow{100} mRNA + P$$
$$2P \xrightarrow{0.001} D,\ mRNA \xrightarrow{0.1} \emptyset,\ P \xrightarrow{1} \emptyset.$$

Following (2.1), and after ordering the species as $(G, mRNA, P, D)$, the rates of the reactions are $\lambda_1(x) = 25x_1$, $\lambda_2(x) = 100x_2$, $\lambda_3(x) = 0.001x_3(x_3 - 1)$, $\lambda_4(x) = 0.1x_2$, and $\lambda_5(x) = x_3$ respectively.

(v) *Toggle*
Each species represses the production of the other, which leads to a probability mass function that is multimodal. The initial state is $A(0) = B(0) = 0$. We set $t = 100$. The reactions are

$$\emptyset \to A,\ A \to \emptyset,\ \emptyset \to B,\ B \to \emptyset.$$

For this model, the first and third intensity functions are not chosen to be mass action. Specifically, we let

$$\lambda_1(x) = \frac{50}{1 + 2x_2},\ \lambda_2(x) = x_1,\ \lambda_3(x) = \frac{50}{1 + 2x_1},\ \lambda_4(x) = x_2,$$

where we again ordered the species as (A, B).

(vi) *Fast/Slow*
A and B quickly convert into one another, and B slowly turns into C. The initial state is $A(0) = B(0) = 100$ and $C(0) = 0$. We set $t = 10$. The reactions are

$$A \xrightarrow{10} B,\ B \xrightarrow{10} A,\ B \xrightarrow{0.1} C.$$

Following (2.1), and after ordering the species as (A, B, C), the rates are $\lambda_1(x) = 10x_1$, $\lambda_2(x) = 10x_2$, and $\lambda_3(x) = 0.1x_2$, respectively.

3. Determining the values of m and h via optimization. The conditional Monte Carlo estimator (1.4) is of little value without knowledge of which values of m and h to use. In this section, we will show that appropriate values can be found by numerically solving an easy optimization problem.

Recall that the distribution of the process is denoted by p_t^ν, and we denote an estimate of this distribution by $\hat{p}_t^\nu$. We will measure the quality of the estimation via the *mean integrated squared error* (MISE), which is

$$\text{MISE}(\hat{p}_t^\nu) \stackrel{\text{def}}{=} \mathbb{E}_{\nu,0}\left[\sum_{x \in \mathbb{Z}^d_{\geq 0}} \left(\hat{p}_t^\nu(x) - p_t^\nu(x)\right)^2\right]. \tag{3.1}$$

Note that if $\hat{p}_t^\nu$ is constructed via our conditional Monte Carlo estimator, then it, and by extension $\text{MISE}(\hat{p}_t^\nu)$, is a function of n, m, and h. Suppose we have a fixed computational budget, which we denote as b. We then want to choose the values of n, m, and h so that we minimize $\text{MISE}(p_t^\nu)$ subject to our budget constraint b.

3.1. Computational cost model. Assuming that our model is non-explosive [3, 37], the expected number of reactions required to generate $\{X_{1j}\}_{j=1}^m$ is given by

$$\underbrace{\mathbb{E}_{\nu,0}\left[\int_0^{t-h}\lambda_0(X(s))\,ds\right]}_{\text{expected \# of reactions in } [0,t-h]} + m\cdot\underbrace{\mathbb{E}_{\nu,0}\left[\int_{t-h}^{t}\lambda_0(X(s))\,ds\right]}_{\text{expected \# of reactions in } [t-h,t]},$$

where $\lambda_0(x)=\sum_{r=1}^R\lambda_r(x)$ (see Theorem A.1). Hence, the expected computational cost for our conditional Monte Carlo estimator is

$$n\cdot c\left(\mathbb{E}_{\nu,0}\left[\int_0^{t-h}\lambda_0(X(s))\,ds\right]+m\cdot\mathbb{E}_{\nu,0}\left[\int_{t-h}^{t}\lambda_0(X(s))\,ds\right]\right), \tag{3.2}$$

where $c>0$ is an unknown constant.

Since we cannot generally evaluate the expectations in the cost model (3.2), as this would be as difficult as the problem we are attempting to solve, we need to estimate them. To do so, fix a relatively small $\tilde{n}$ and simulate $\tilde{n}$ i.i.d. paths $\{X_i\}_{i=1}^{\tilde{n}}$. Then the expectations are approximately equal to

$$\frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}}\int_0^{t-h}\lambda_0(X_i(s))\,ds,\ \text{and}\ \frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}}\int_{t-h}^{t}\lambda_0(X_i(s))\,ds. \tag{3.3}$$

Importantly, for the fixed set of $\tilde{n}$ paths, the values (3.3) can be computed for a variety of different h values. The process X_i is piecewise constant, and therefore so is $\lambda_0(X_i)$. Thus, for any value of h, we can easily compute the integrals so long as we have stored the jump times of X_i and the value of $\lambda_0(X_i)$ at each jump.

3.2. Optimization problem. Given a reaction network, our goal is to find values of n, m, and h that minimize the mean integrated squared error (MISE) (3.1) for our conditional Monte Carlo estimator (1.4) while staying within our computational budget of b. More precisely, we want to solve the following optimization problem

$$\min_{n,m,h}\ \underbrace{\mathbb{E}_{\nu,0}\left[\sum_{x\in\mathbb{Z}^d_{\geq 0}}\left(\hat{p}_t^{\nu}(x;n,m,h)-p_t^{\nu}(x)\right)^2\right]}_{\text{mean integrated squared error (MISE)}}, \tag{3.4}$$

subject to

$$\begin{aligned} &n\cdot c\left(\mathbb{E}_{\nu,0}\left[\int_0^{t-h}\lambda_0(X(s))\,ds\right]+m\cdot\mathbb{E}_{\nu,0}\left[\int_{t-h}^{t}\lambda_0(X(s))\,ds\right]\right)\leq b \\ &n,m\in\mathbb{Z}_{\geq 1}\ \text{and}\ 0\leq h\leq t. \end{aligned} \tag{3.5}$$

The following theorem will allow us to transform the above optimization problem into a more solvable form.

THEOREM 3.1. *Suppose the process X is non-explosive. For any fixed $n,m\in\mathbb{Z}_{\geq 1}$*

and $h \in [0,t]$

$$\mathbb{E}_{\nu,0}\left[\sum_{x\in\mathbb{Z}^d_{\geq 0}}\left(\hat{p}^{\nu}_t(x;n,m,h)-p^{\nu}_t(x)\right)^2\right] = \frac{1}{n}\left[\frac{1}{m}+\left(1-\frac{1}{m}\right)P_\nu(X_{11}(t)=X_{12}(t))-\sum_{x\in\mathbb{Z}^d_{\geq 0}}p^{\nu}_t(x)^2\right].$$

The proof of Theorem 3.1 can be found in Appendix A.2.

If we allow n to be continuous, then we can use the constraint (3.5) to solve for n^{-1}, and subsequently eliminate the constraint by substitution. This leads to a simpler optimization problem. In particular, let

$$f(m,h) \stackrel{\text{def}}{=} \left(\frac{1}{m}\,\mathbb{E}_{\nu,0}\left[\int_0^{t-h}\lambda_0(X(s))\,ds\right]+\mathbb{E}_{\nu,0}\left[\int_{t-h}^{t}\lambda_0(X(s))\,ds\right]\right) \times\left(1+(m-1)P_\nu(X_{11}(t)=X_{12}(t))-m\sum_{x\in\mathbb{Z}^d_{\geq 0}}p^{\nu}_t(x)^2\right).$$

Then the original optimization problem (3.4) and (3.5) is equivalent to

$$\min_{m,h} f(m,h) \quad m\in\mathbb{Z}_{\geq 1}, 0\leq h\leq t. \tag{3.6}$$

Note that both c and b have dropped out of the optimization problem.

There are three terms in f that we must know, or be able to approximate, in order to solve (3.6).

- The expectations of the integrals. We discussed how to approximate these in subsection 3.1.
- The sum $\sum_x p^{\nu}_t(x)^2$. However, we note that $\sum_x p^{\nu}_t(x)^2$ is the probability that two independent paths end up in the same state at time t. For many models, including the ones we tested, that sum is extremely close to zero. Thus for our examples, we just replace this sum with zero, and make that our general recommendation.
- The term $P_\nu(X_{11}(t) = X_{12}(t))$, whose approximation is the subject of the next section.

3.3. Approximating the joint probability. In order to optimize the objective function $f(m,h)$ in (3.6), we need to know, or be able to quickly approximate, the term $P_\nu(X_{11}(t) = X_{12}(t))$. The following theorem, proven in Appendix A.3, will allow us to make a good approximation, without requiring any additional simulations.

THEOREM 3.2. *Let S be the $d\times R$ matrix whose rth column is ζ_r and let null(S) be the right nullspace of S restricted to integer values. Let X and Z satisfy*

$$X(t) = X(0)+\sum_{r=1}^{R}Y^X_r\left(\int_0^t\lambda_r(X(s))ds\right)\zeta_r,$$

$$Z(t) = X(0)+\sum_{r=1}^{R}Y^Z_r\left(\int_0^t\lambda_r(X(s))ds\right)\zeta_r,$$

where the Y_r^X and Y_r^Z are independent, unit-rate Poisson processes. Assume that X is non-explosive. For each $1 \le r \le R$ and $0 \le a \le b \le t$, denote

$$\Lambda_r^{a,b} = \int_a^b \lambda_r(X(s))ds,$$

and let $K_r^{a,b}$ have the Skellam($\Lambda_r^{a,b}, \Lambda_r^{a,b}$) distribution. Then

$$P_\nu(X(t) = Z(t)) = \sum_{k \in null(S)} \mathbb{E}_{\nu,0}\left[\prod_{r=1}^{R} P\left(K_r^{0,t} = k_r \mid \Lambda_r^{0,t}\right)\right]. \tag{3.7}$$

Note that X is the process (2.2) that is of interest to us. Returning to our setup, if we assume that

$$\int_{t-h}^{t} \lambda_r(X_{11}(s))\, ds \approx \int_{t-h}^{t} \lambda_r(X_{12}(s))\, ds,$$

which should be valid for small h, then Theorem 3.2 leads to an approximation of $P_\nu(X_{11}(t) = X_{12}(t))$. In particular, we may sample $\tilde{n}$ paths and for the ith such path define

$$\Lambda_{r,i}^{t-h,t} = \int_{t-h}^{t} \lambda_r(X_i(s))\, ds,\ 1 \le i \le \tilde{n}.$$

Then $P_\nu(X_{11}(t) = X_{12}(t)) \approx \hat{P}_\nu(X_{11}(t) = X_{12}(t))$, where

$$\hat{P}_\nu(X_{11}(t) = X_{12}(t)) \stackrel{\text{def}}{=} \sum_{k \in \tilde{N}} \frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} \prod_{r=1}^{R} P\left(K_r^{t-h,t} = k_r \mid \Lambda_{r,i}^{t-h,t}\right), \tag{3.8}$$

and $\tilde{N}$ is a finite subset of null(S). Since the probabilities inside the product are symmetric about $k_r = 0$, a reasonable choice of $\tilde{N}$ would be a rectangle with the same number of dimensions as null(S) and centered at 0. For our implementation, we let $\tilde{N}$ be equal to linear combinations of the basis of the nullspace, where the coefficients in the linear combinations ranged over a finite set centered at zero (we used $\{-4, \ldots, 4\}$). For all the models we tested, this method was sufficiently fast and accurate. Algorithm 3.1 summarizes how we compute $\hat{P}_\nu(X_{11}(t) = X_{12}(t))$.

3.4. Approximation to the optimization problem. By using the joint probability approximation (3.8), we can approximate the function f in the optimization problem (3.6). In particular, let

$$\hat{f}(m,h) \stackrel{\text{def}}{=} \left(\frac{1}{m}\int_0^{t-h} \bar{\lambda}_0(X(s))\, ds + \int_{t-h}^{t} \bar{\lambda}_0(X(s))\, ds\right) \left(1 + (m-1)\hat{P}_\nu(X_{11}(t) = X_{12}(t)) - m \sum_{x \in \mathbb{Z}^d_{\ge 0}} p_t^\nu(x)^2\right), \tag{3.9}$$

where $\bar{\lambda}_0(X(s)) = \frac{1}{\tilde{n}} \sum_{i=1}^{\tilde{n}} \sum_{r=1}^{R} \lambda_r(X_i(s))$, and the $\{X_i\}_{i=1}^{\tilde{n}}$ are independent paths of X. Then we may substitute f with $\hat{f}$ and our new optimization problem is the

Algorithm 3.1 Algorithm for computing $\hat{P}_\nu(X_{11}(t) = X_{12}(t))$

Require: $\tilde{n}$ i.i.d. samples of X, denoted $\{X_i\}_{i=1}^{\tilde{n}}$ ▷ $\tilde{n} = 500$ was more than sufficient.
Require: the stoichiometry matrix S, and a finite $\tilde{N} \subset \text{null}(S)$
for all r in $1, \ldots, R$ and i in $1, \ldots, \tilde{n}$ **do**
  $\Lambda_{r,i}^{t-h,t} \leftarrow \int_{t-h}^{t} \lambda_r(X_i(s))\, ds$
end for

$\hat{P} \leftarrow 0$
for all k in $\tilde{N}$ and i in $1, \ldots, \tilde{n}$ **do**
  $\hat{P} \leftarrow \hat{P} + \prod_{r=1}^{R} P\left(K_r^{t-h,t} = k_r \,\middle|\, \Lambda_{r,i}^{t-h,t}\right)$ ▷ $K_r^{t-h,t} \sim \text{Skellam}(\Lambda_{r,i}^{t-h,t}, \Lambda_{r,i}^{t-h,t})$
end for

$\hat{P}_\nu(X_{11}(t) = X_{12}(t)) \leftarrow \hat{P}/\tilde{n}$

following:

$$\min_{m,h} \hat{f}(m,h) \qquad m \in \mathbb{R}_{\geq 1}, 0 \leq h \leq t. \tag{3.10}$$

Note that above we have allowed m to be real–valued, as opposed to integer valued. This allows us to use continuous optimization algorithms, which generally converge more rapidly. According to Figure SM1, which shows $\hat{f}(m,h)$ for many values of m and h, $\hat{f}$ does not change too quickly with m, so allowing m to range over the reals instead of the integers should not change the optimal values of m and h appreciably.

It is important to know when the optimization problem (3.10) has a finite solution. In the proposition below, we show that a solution necessarily exists when $\hat{P}_\nu(X_{11}(t) = X_{12}(t))$ is larger than the approximation used for $\sum_x p_t^\nu(x)^2$. Since we approximate the sum with zero, we may conclude that a finite solution always exists in our setup.

PROPOSITION 3.3. *Let $\widehat{p^2}$ be our approximation to $\sum_x p_t^\nu(x)^2$. If $\hat{P}_\nu(X_{11}(t) = X_{12}(t)) > \widehat{p^2}$ for all $h \in [0,t]$, then (3.10) has a finite solution.*

Proof. Since the integrals are nonnegative, h is in a compact domain, $\hat{f}$ depends continuously on h and m, and $\lim_{m\to\infty} \hat{f}(m,h) = \infty$, a finite solution exists. ∎

Algorithm 3.2 outlines the full conditional Monte Carlo algorithm, which brings together all of the individual pieces of the algorithm that we previously discussed.

4. Numerical results. In this section, we present numerical results demonstrating the improvement in accuracy, quantified via the mean integrated squared error (3.1), that comes from using our conditional Monte Carlo estimator instead of the classical Monte Carlo estimator. In particular, when near–optimal values of m and h are utilized, the accuracy often improves by an order of magnitude for a fixed computational budget. Moreover, we show that the function $\hat{f}$ of (3.10) is indeed a very good approximation for f of (3.6) for the examples we considered, allowing us to conclude that the values of m and h our method produces are near–optimal.

The following steps were carried out on each of our test examples. First, we fixed

Algorithm 3.2 Conditional Monte Carlo algorithm

Require: $\tilde{n}$ i.i.d. samples of X, denoted $\{X_i\}_{i=1}^{\tilde{n}}$ ▷ $\tilde{n} = 500$ was more than sufficient.

$m, h \leftarrow \arg\min_{\substack{m\in\mathbb{R}_{\geq 1}\\ 0\leq h\leq t}} \hat{f}(m,h)$

▷ Use $\{X_i\}_{i=1}^{\tilde{n}}$, (3.9), and Algorithm 3.1 to evaluate $\hat{f}$.

for all i in $1,\dots,n$ **do**

  Sample $X_i(t-h)$. ▷ The $X_i(t-h)$ are i.i.d.

  for all j in $1,\dots,m$ **do**

    Sample $X_{ij}(t)$ conditioned on $X_{ij}(t-h) = X_i(t-h)$.

    ▷ See section 1 for details about X_{ij}.

  end for

end for

$\hat{p}_t^{\nu}(x;n,m,h) \leftarrow \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m}\sum_{j=1}^{m}\mathbb{1}(X_{ij}(t)=x)$

an integer n_1 and computed the classical Monte Carlo estimator

$$p_t^{\mathrm{MC}}(x;n_1) = \frac{1}{n_1}\sum_{i=1}^{n_1}\mathbb{1}(X_i(t)=x),\ x\in\mathbb{Z}^d_{\geq 0}.$$

For all models, we used $n_1 = 10^4$. We also recorded the number of random variates used in generating $p_t^{\mathrm{CMC}}(\,\cdot\,;n_1)$, which served as the budget b in the computational cost constraint (3.5).

After obtaining $p_t^{\mathrm{MC}}(\,\cdot\,;n_1)$, we computed the conditional Monte Carlo estimator

$$p_t^{\mathrm{CMC}}(x;n_2,m,h) = \frac{1}{n_2}\sum_{i=1}^{n_2}\frac{1}{m}\sum_{j=1}^{m}\mathbb{1}(X_{ij}(t)=x),\ x\in\mathbb{Z}^d_{\geq 0},$$

for various pairs of m and h, and n_2 was allowed to increase until the conditional estimator used essentially the same number of random variates as the classical Monte Carlo estimator. All random variates generated for the conditional estimators were independent of those utilized for the classical estimator.

Next, for both classical and conditional Monte Carlo, we computed the integrated squared error

$$\mathrm{ISE} = \sum_{\tilde{S}}\left(\hat{p}(x) - p_t^{\nu}(x)\right)^2, \tag{4.1}$$

where $\tilde{S}$ was a large fixed subset of the state space, and $\hat{p}(x)$ was either the classical or conditional Monte Carlo estimate. The ISE is itself a random variable, and so we approximated the mean integrated square error (MISE) by averaging 100 independent samples of the ISE.

The exact values of $p_t^{\nu}(x)$ were unknown. Thus the values were estimated with conditional Monte Carlo with a large value of n_1 (we used $n_1 = 10^9$), and with m and h chosen so that they approximately minimize the MISE.

Finally, we denote by $\mathrm{MISE}_{\mathrm{MC}}$ our estimate of the classical Monte Carlo MISE, and, for a given m and h, we denote by $\mathrm{MISE}_{\mathrm{CMC}}(m,h)$ the conditional version. For each model, and for each choice of m and h, an "empirical error improvement" was

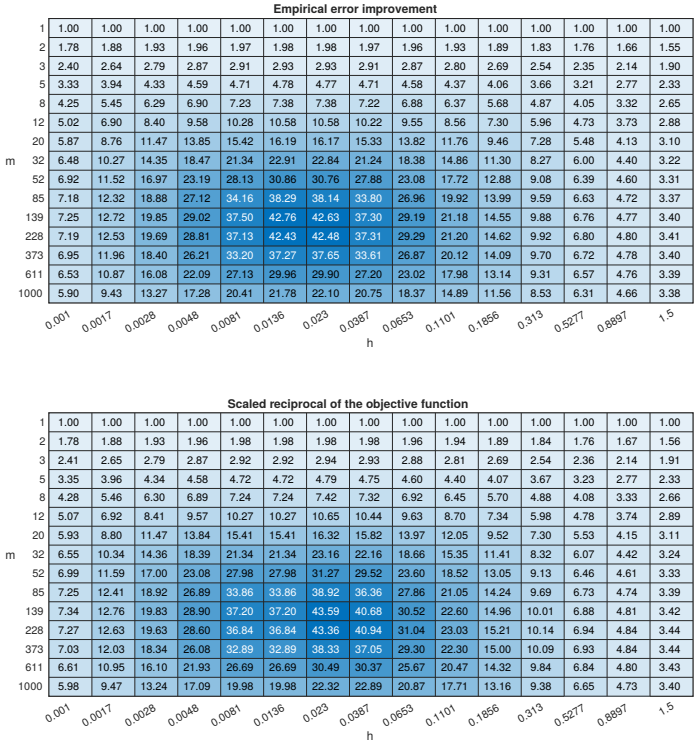


FIG. 2. *Lotka-Volterra model. The first heatmap shows $MISE_{MC}/MISE_{CMC}(m,h)$ for different values of m and h. The method we used to obtain the ratio is described in section 4. The second heatmap shows that value of $\hat{f}(1,0)/\hat{f}(m,h)$. The definition of $\hat{f}$ is given by (3.9).*

computed as the following ratio

$$\frac{\text{MISE}_{\text{MC}}}{\text{MISE}_{\text{CMC}}(m,h)},$$

where a number greater than one implies that conditional Monte Carlo has a lower MISE than classical Monte Carlo when given the same computational budget. These values, one for each pair of m and h, can then be plotted. In the top half of Figures 2 and 3 (and Figures SM2 to SM5), we display these values with a heatmap. Of particular interest is the order of magnitude improvement in computational efficiency we see with the conditional Monte Carlo estimator as compared to classical Monte Carlo *when well–chosen values of h and m are utilized*. In particular, for the Lotka-Volterra model we see a 40-fold improvement, for the dimerization model we see a 20-fold improvement, for the toggle model we see a 20-fold improvement, and for the fast/slow model we see a 20-fold improvement. For the birth and birth–death models we see more modest improvements in computational efficiency, but this can be explained by the simplicity of these models which makes classical Monte Carlo sufficient for the task at hand. In particular, one promising aspect of the present work comes into focus with these numerical results: the more complicated the model, and the larger and more diffuse the distribution of the model (which is where other methods, including those that approximately solve the chemical master equation directly, struggle), the better the performance of the conditional Monte Carlo estimator.

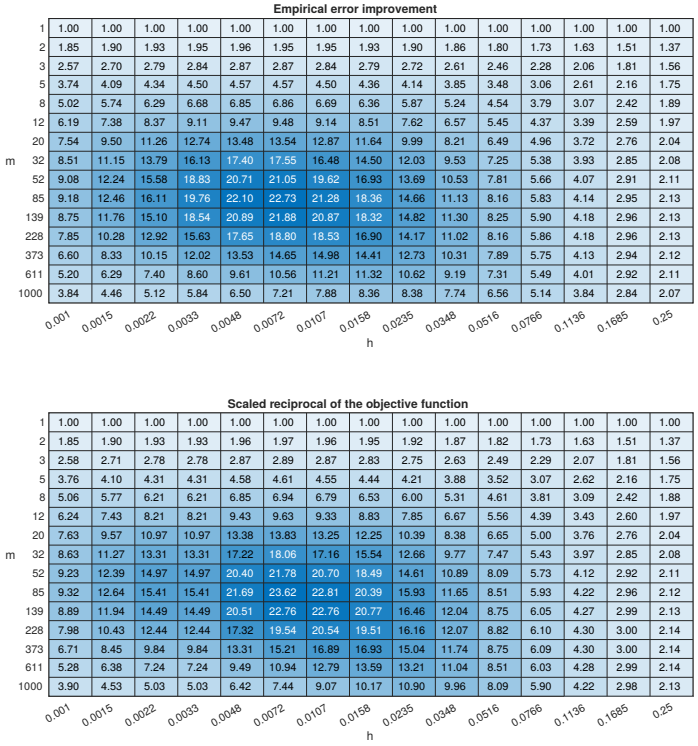


FIG. 3. *Dimerization model. The first heatmap shows $MISE_{MC}/MISE_{CMC}(m,h)$ for different values of m and h. The method we used to obtain the ratio is described in section 4. The second heatmap shows that value of $\hat{f}(1,0)/\hat{f}(m,h)$. The definition of $\hat{f}$ is given by (3.9).*

Of course, we do not a priori know the optimal values of the parameters m and h, and must find them via our optimization problem (3.10). In each of the bottom portions of Figures 2 and 3 (and Figures SM2 to SM5), we provide the values of $\hat{f}(m,h)$ for the different pairs of m and h. We report the inverse so that the heatmap will agree qualitatively with the top portion of the figures (higher values are desirable). We also normalized the values by multiplying them by $\hat{f}(1,0)$, which does not affect the results of the optimization problem in any way. To generate each value $1/\hat{f}(m,h)$ we first sampled $\tilde{n} = 500$ paths, which then allowed us to compute $\bar{\lambda}_0$ and $\hat{P}_\nu(X_{11}(t) = X_{12}(t))$ as detailed in the previous section. We could then use these values to compute $\hat{f}(m,h)$ via (3.9).

Note that the empirical error improvement and $\hat{f}$ do not need to have the same value for a pair of m and h. The important thing is that the maximizer of the empirical error improvement is similar to the minimizer of $\hat{f}$. The heatmaps do indeed suggest that the true and approximate optimization problems have similar solutions. What is also clear from these numerical results is that even if m and h slightly deviate from their optimal values, we still get a substantial improvement.

We stress that such heatmaps do not need to be made by anyone who uses the conditional Monte Carlo algorithm. They are only used here to demonstrate that the optimization problem (3.10) can be safely used to find the near–optimal values of m and h, which can then be used to construct the desired estimator (1.4) via Algorithm 3.2.

5. A central limit theorem. In this section, we will show how to obtain an approximate one-sided confidence interval for the integrated squared error (4.1) without running more simulations. Specifically, for a fixed (presumably large) finite subset of the state space $\tilde{S}$, a fixed $\alpha \in (0,1)$, and large n, we want to find a sequence of positive constants $\{C_n\}$ and a constant $u > 0$ such that

$$\lim_{n\to\infty} P\Big(C_n \underbrace{\sum_{x\in\tilde{S}} \left(\hat{p}_t^{\nu}(x;n,m,h) - p_t^{\nu}(x)\right)^2}_{\text{integrated squared error}} \leq u\Big) = 1-\alpha, \tag{5.1}$$

where C_n is allowed to depend on m and h. The following central limit theorem will lead us to values for $\{C_n\}$ and u.

THEOREM 5.1. *Fix $m \in \mathbb{Z}_{\geq 1}$ and $h \in [0,t]$. Let $\mathcal{S} \subset \mathbb{Z}_{\geq 0}^d$ be the state space of the continuous time Markov chain, and let $\tilde{\mathcal{S}}$ be a finite subset of $\mathcal{S}$. Choose an enumeration of $\tilde{\mathcal{S}}$ and denote it $\{x_i\}_{i=1}^{|\tilde{\mathcal{S}}|}$. Let $p_t^{\nu}, \hat{p}_t^{\nu} \in \mathbb{R}^{|\tilde{\mathcal{S}}|}$ with their ith elements equal to $p_t^{\nu}(x_i)$ and $\hat{p}_t^{\nu}(x_i;n,m,h)$, respectively. Let*

$$\Sigma \stackrel{def}{=} m\, diag(p_t^{\nu}) + m(m-1)A - m^2 p_t^{\nu}(p_t^{\nu})^T, \tag{5.2}$$

where $diag(p_t^{\nu})$ is the diagonal matrix with p_t^{ν} along its diagonal, and A is a $|\tilde{\mathcal{S}}| \times |\tilde{\mathcal{S}}|$ matrix where $A_{ij} = P_{\nu}(X_{11}(t) = x_i, X_{12}(t) = x_j)$. Then

$$nm^2 \sum_{x\in\tilde{\mathcal{S}}} \left(\hat{p}_t^{\nu}(x;n,m,h) - p_t^{\nu}(x)\right)^2 \stackrel{d}{\to} \sum_{\ell=1}^{|\tilde{S}|} \lambda_\ell Z_\ell^2, \text{ as } n\to\infty, \tag{5.3}$$

where the $\{\lambda_\ell\}_{\ell=1}^{|\tilde{S}|}$ are the eigenvalues of Σ and $Z_\ell \stackrel{i.i.d.}{\sim} N(0,1)$.

Σ is usually an enormous matrix, so we do not want to store it, much less compute its eigenvalues. The Satterthwaite approximation [43] says that

$$\sum_\ell \lambda_\ell Z_\ell^2 \stackrel{d}{\approx} \frac{\sum_\ell \lambda_\ell^2}{\sum_\ell \lambda_\ell} \chi^2\left(\frac{\left(\sum_\ell \lambda_\ell\right)^2}{\sum_\ell \lambda_\ell^2}\right) = \frac{\mathrm{tr}\left(\Sigma^2\right)}{\mathrm{tr}\left(\Sigma\right)} \chi^2\left(\frac{\mathrm{tr}\left(\Sigma\right)^2}{\mathrm{tr}\left(\Sigma^2\right)}\right), \tag{5.4}$$

where $\chi^2(v)$ denotes a χ^2 random variable with v degrees of freedom. The advantage of this approximation is that we can estimate $\mathrm{tr}\left(\Sigma\right)$ and $\mathrm{tr}\left(\Sigma^2\right)$ without storing Σ explicitly or computing its eigenvalues.

THEOREM 5.2. *Fix $n, m \in \mathbb{Z}_{\geq 1}$ and $h \in [0,t]$. Let $\tilde{\mathcal{S}}$, $\{x_k\}_{k=1}^{|\tilde{\mathcal{S}}|}$, and $\hat{p}_t^{\nu}$ be defined as in Theorem 5.1. For $1 \leq i \leq n$, let $M_i \in \mathbb{Z}_{\geq 0}^{|\tilde{\mathcal{S}}|}$, and set its kth element to $M_i(x_k) \stackrel{def}{=} \sum_{j=1}^m \mathbb{1}(X_{ij} = x_k)$ (the $\{X_{ij}\}$ are defined in section 1). Let $\hat{\Sigma}_n$ be the usual sample covariance matrix of $\{M_i\}_{i=1}^n$. Specifically,*

$$\hat{\Sigma}_n \stackrel{def}{=} \frac{1}{n-1} \sum_{i=1}^n \left(M_i - \overline{M}\right)\left(M_i - \overline{M}\right)^T,$$

where $\overline{M} = n^{-1}\sum_{i=1}^n M_i$. Then

$$tr\left(\hat{\Sigma}_n\right) = \frac{1}{n-1}\sum_{i=1}^n M_i^T M_i - \frac{nm^2}{n-1}(\hat{p}_t^{\nu})^T \hat{p}_t^{\nu}, \tag{5.5}$$

and

$$tr\left(\hat{\Sigma}_n^2\right) = \frac{1}{(n-1)^2}\sum_{i=1}^{n}\left[M_i^T M_i - 2\overline{M}^T M_i + m^2(\hat{p}_t^\nu)^T\hat{p}_t^\nu\right]^2 + \frac{2}{(n-1)^2}\sum_{1\le i<j\le n}\left[M_i^T M_j - \overline{M}^T M_i - \overline{M}^T M_j + m^2(\hat{p}_t^\nu)^T\hat{p}_t^\nu\right]^2. \tag{5.6}$$

Furthermore

$$tr\left(\hat{\Sigma}_n\right) \xrightarrow{a.s.} tr(\Sigma) \text{ and } tr\left(\hat{\Sigma}_n^2\right) \xrightarrow{a.s.} tr\left(\Sigma^2\right) \text{ as } n\to\infty.$$

Since the optimal h is generally small and the optimal value of m tends to be only moderately large (on the order of 10 to 100 for the models we tested), the indicator in the summand of $M_i(x)$ is zero for many values of x. In other words, M_i is sparse. Consequently, storing $\{M_i\}_{i=1}^n$ does not require too much memory, and the terms $M_i^T M_j$ and $\overline{M}^T M_i$ are cheap to compute. Algorithm 5.1 summarizes how we compute the traces. Using the sparsity of the M_i is important, because otherwise the vectors are too large to store and the operations are slow.

Algorithm 5.1 Algorithm for computing $\hat{p}_t^\nu$, $\mathrm{tr}\left(\hat{\Sigma}_n\right)$, and $\mathrm{tr}\left(\hat{\Sigma}_n^2\right)$

Require: $n, m \in \mathbb{Z}_{\ge 1}$ and $h \in [0, t]$

for i in $\{1, \dots, n\}$ **do**
  Sample $X_i(t-h)$.
  Given $X_i(t-h)$, sample $\{X_{ij}(t)\}_{j=1}^m$.
  for x in $\tilde{S}$ **do**
    $M_i(x) \leftarrow \sum_{j=1}^m \mathbb{1}(X_{ij}(t) = x)$ ▷ Store M_i as a sparse vector.
  end for
end for

$\hat{p}_t^\nu \leftarrow \frac{1}{nm}\sum_{i=1}^n M_i$
Compute $\mathrm{tr}\left(\hat{\Sigma}_n\right)$ according to (5.5).
Compute $\mathrm{tr}\left(\hat{\Sigma}_n^2\right)$ according to (5.6).

COROLLARY 5.3. *Fix $n, m \in \mathbb{Z}_{\ge 1}$ and $h \in [0, t]$. Also fix an $\alpha \in (0, 1)$, and let $\chi_\alpha^2(v)$ be the $1-\alpha$ quantile of the χ^2 distribution with v degrees of freedom. An approximate $1-\alpha$ confidence interval for $\sum_{x\in\tilde{\mathcal{S}}}\left(\hat{p}_t^\nu(x; n, m, h) - p_t^\nu(x)\right)^2$ is $[0, U_n/(nm^2)]$, where*

$$U_n \overset{def}{=} \frac{tr\left(\hat{\Sigma}_n^2\right)}{tr\left(\hat{\Sigma}_n\right)}\chi_\alpha^2\left(\frac{tr\left(\hat{\Sigma}_n\right)^2}{tr\left(\hat{\Sigma}_n^2\right)}\right). \tag{5.7}$$

Figures 4a and 4b (and also Figures SM6 to SM9), compare the empirical distribution of

$$nm^2\sum_{x\in\tilde{\mathcal{S}}}\left(\hat{p}_t^\nu(x; n, m, h) - p_t^\nu(x)\right)^2 \tag{5.8}$$

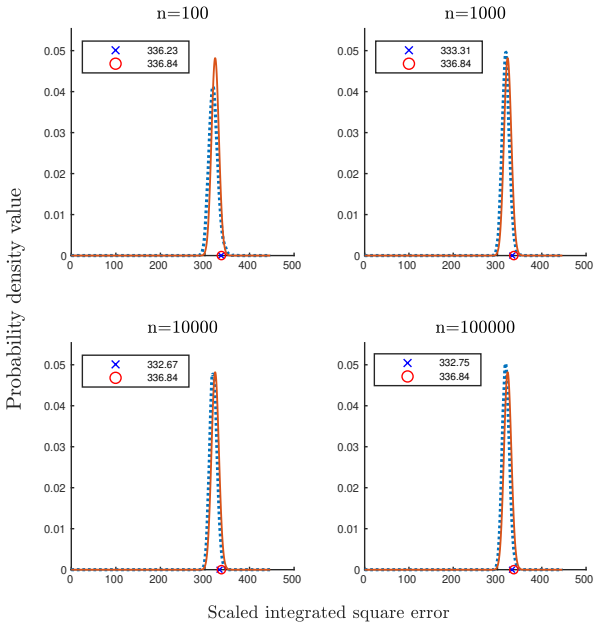
to our approximate asymptotic distribution (5.4), in which we replace the true traces with the sample traces from Algorithm 5.1. The figures also compare the sample 95% quantile to the same quantile based on Corollary 5.3. Some of the empirical distributions have more of a distinct "peak" than the approximate distribution. However, we simply want a confidence interval for the integrated squared error, and for all of the models the empirical and approximate 95% quantiles are very close.

6. Directions for future research. We demonstrated how to implement a version of conditional Monte Carlo in the context of continuous time Markov chain models for reaction networks. There are many possible directions for future research; we list two.

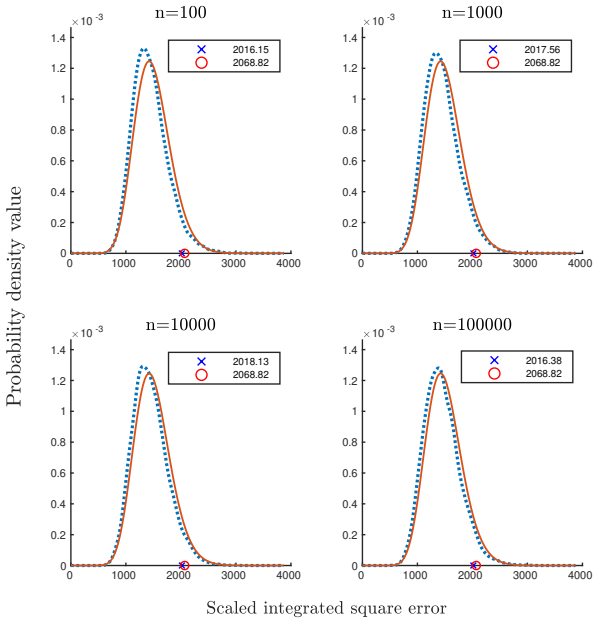
1. The method could be extended so that it provides estimates of the distribution at multiple fixed time-points. The method we developed here, and in particular the optimization problem we utilize to find the values of m and h, is specifically tailored to the single time-point case.
2. In the method developed here the conditional expectation in (1.3),

$$\mathbb{E}_{X_i(t-h),t-h}\left[\mathbb{1}(X(t)=x)\right],$$

is approximated by Monte Carlo with m conditionally independent realizations. However, it could instead be approximated by solving the chemical master equation directly, perhaps via a version of the finite state projection algorithm [36]. Because the solver need only integrate the system of ODEs over the time interval $[t-h,t]$, the probability mass should not become too diffuse, thereby solving one of the major difficulties related to these solvers. We implemented this approach. We did observe some increase in efficiency over the conditional Monte Carlo algorithm of Algorithm 3.2, with a typical gain in efficiency of a factor of three. However, these gains were only realized when an optimal value of h was chosen, and we needed to test many different h values in order to find the optimal value. In practice, we would need a faster method for finding the optimal parameters, similar to the optimization problem detailed in this paper.



(a) Lotka-Volterra model.



(b) Dimerization model.

FIG. 4. *The dashed blue density is the empirical density of the integrated squared error (5.8), whereas the solid red density is the Satterwaithe approximation to the asymptotic density (5.4). The blue cross and red circle are the 95% quantiles of their respective densities. To generate the blue curve, first we sampled* 10^4 *values of* $nm^2 \sum_{x\in\tilde{\mathcal{S}}} \left(\hat{p}_t^{\nu}(x;n,m,h) - p_t^{\nu}(x)\right)^2$ *(which we call the "scaled integrated squared error") for different values of* n*. Given those samples, we used MATLAB's ksdensity function to generate the blue curve. The traces of* Σ *and* Σ^2 *were estimated with an independent set of* 10^5 *simulations and Algorithm 5.1.*

Appendix A. Proofs.

A.1. Theorem regarding the expected number of reactions.

THEOREM A.1. *Suppose that the process X is non-explosive and fix $h \in [0,t]$ and $m \in \mathbb{Z}_{\geq 1}$. Then the expected number of reactions required to sample $\{X_{1j}\}_{j=1}^m$ is*

$$\mathbb{E}_{\nu,0}\left[\int_0^{t-h} \lambda_0(X(s))\,ds\right] + m\,\mathbb{E}_{\nu,0}\left[\int_{t-h}^{t} \lambda_0(X(s))\,ds\right].$$

Proof. The number of reactions required to sample $\{X(s)\}_{s\in[a,b]}$ is

$$\sum_{r=1}^{R}\left[Y_r\left(\int_0^b \lambda_r\left(X(s)\right)\,ds\right) - Y_r\left(\int_0^a \lambda_r\left(X(s)\right)\,ds\right)\right],$$

where the Y_r are independent unit-rate Poisson processes [30]. For each r,

$$Y_r\left(\int_0^t \lambda_r\left(X(s)\right)\,ds\right) - \int_0^t \lambda_r\left(X(s)\right)\,ds$$

is a martingale [6, Theorem 1.22], so the result follows. ∎

A.2. Proof of Theorem 3.1. For simplicity, denote $X_{ij}(t)$ as X_{ij}. We start with the left-hand side of the desired equality. The monotone convergence theorem implies that we can move the expectation inside the sum, by which we mean

$$\begin{aligned}\mathbb{E}_{\nu,0}\left[\sum_x \left(\hat{p}_t^{\,\nu}(x;n,m,h) - p_t^\nu(x)\right)^2\right] &= \sum_x \mathbb{E}_{\nu,0}\left[\left(\hat{p}_t^{\,\nu}(x;n,m,h) - p_t^\nu(x)\right)^2\right]\\ &= \sum_x \mathrm{Var}[\hat{p}_t^{\,\nu}(x;n,m,h)].\end{aligned}$$

The last line follows from the fact that the estimator $\hat{p}_t^{\,\nu}$ is unbiased. From the definition of $\hat{p}_t^{\,\nu}$, and also basic properties of variance, the above is equal to

$$\begin{aligned}&= \sum_x \mathrm{Var}\left[\frac{1}{nm}\sum_{i=1}^n\sum_{j=1}^m \mathbb{1}(X_{ij}=x)\right]\\
&= \frac{1}{nm^2}\sum_x\left[\sum_{j=1}^m \mathrm{Var}[\mathbb{1}(X_{1j}=x)] + 2\sum_{1\leq i<j\leq m}\mathrm{Cov}\big(\mathbb{1}(X_{1i}=x),\mathbb{1}(X_{1j}=x)\big)\right]\\
&= \frac{1}{nm^2}\sum_x\left[m\mathrm{Var}[\mathbb{1}(X_{11}=x)] + m(m-1)\mathrm{Cov}(\mathbb{1}(X_{11}=x),\mathbb{1}(X_{12}=x))\right]\\
&= \frac{1}{nm}\sum_x\left[p_t^\nu(x)(1-p_t^\nu(x)) + (m-1)\Big(\mathbb{E}_{\nu,0}\left[\mathbb{1}(X_{11}=x)\mathbb{1}(X_{12}=x)\right] - p_t^\nu(x)^2\Big)\right]\\
&= \frac{1}{nm}\sum_x\left[p_t^\nu(x) + (m-1)P_\nu(X_{11}=x,X_{12}=x) - mp_t^\nu(x)^2\right]\\
&= \frac{1}{n}\left[\frac{1}{m} + \left(1-\frac{1}{m}\right)P_\nu(X_{11}=X_{12}) - \sum_x p_t^\nu(x)^2\right].\end{aligned}$$

We can also take $p_t^\nu(x)$ to be a marginal distribution. In that case, interpret sums over x as sums over the lower-dimensional marginal variables. Also, view $X_{11}=X_{12}$ as being true if their coordinates corresponding to the marginal variables are equal.

A.3. Proof of Theorem 3.2. Let $\Lambda^{0,t} \in \mathbb{R}^R_{\geq 0}$ be the vector whose rth element is $\Lambda_r^{0,t}$, and let $Y^X, Y^Z \in \mathbb{Z}^R_{\geq 0}$ be the vectors whose rth elements are $Y_r^X(\Lambda_r^{0,t})$ and $Y_r^Z(\Lambda_r^{0,t})$, respectively. Then

$$\begin{aligned} P_\nu(X(t) = Z(t)) &= P_\nu\left(SY^X = SY^Z\right) \\ &= P_\nu\left(S(Y^X - Y^Z) = 0\right) \\ &= \sum_{k \in \text{null}(S)} P_\nu(Y^X - Y^Z = k) \\ &= \sum_{k \in \text{null}(S)} \mathbb{E}_{\nu,0}\left[P(Y^X - Y^Z = k \,\middle|\, \Lambda^{0,t})\right]. \end{aligned}$$

The elements of Y^X and Y^Z are independent when conditioned on $\Lambda^{0,t}$. Therefore we can expand the conditional probability into a product of probabilities, by which we mean

$$P\left(Y^X - Y^Z = k \,|\, \Lambda^{0,t}\right) = \prod_{r=1}^R P\left(Y_r^X - Y_r^Z = k_r \,\middle|\, \Lambda_r^{0,t}\right).$$

When conditioned on $\Lambda_r^{0,t}$, $Y_r^X - Y_r^Z$ is the difference of two independent Poissons with the same intensity $\Lambda_r^{0,t}$. Therefore the difference follows a Skellam distribution. To summarize,

$$K_r^{0,t} \stackrel{\text{def}}{=} Y_r^X - Y_r^Z \sim \text{Skellam}(\Lambda_r^{0,t}, \Lambda_r^{0,t}), \text{ when conditioned on } \Lambda^{0,t}.$$

Continuing from above,

$$P_\nu(X_{11}(t) = X_{12}(t)) = \sum_{k \in \text{null}(S)} \mathbb{E}_{\nu,0}\left[\prod_{r=1}^R P\left(K_r^{0,t} = k_r \,\middle|\, \Lambda_r^{0,t}\right)\right],$$

where the expectation is taken over $\Lambda^{0,t}$.

If we are estimating a marginal distribution, then we need to modify the sum slightly. Let S' be the same as S, except the rows corresponding to the marginalized-out variables are removed. Then replace null(S) with null(S').

A.4. Proof of Theorem 5.1. Let $\{X_i(t-h)\}_{i=1}^n$ be i.i.d. realizations of $X(t-h)$. Define $X_{ij}(t)$ to be the state of the CTMC conditioned on $X_{ij}(t-h) = X_i(t-h)$, where $1 \leq j \leq m$. For simplicity, later we will denote $X_{ij}(t)$ as just X_{ij}.

Let $M_i \in \mathbb{Z}^{|\tilde{\mathcal{S}}|}_{\geq 0}$, where the kth element of M_i is defined as $\sum_{j=1}^m \mathbb{1}(X_{ij} = x_k)$. Let $\Sigma \in \mathbb{R}^{|\tilde{\mathcal{S}}| \times |\tilde{\mathcal{S}}|}$ be the covariance matrix of M_1. The M_i are i.i.d., so if Σ is finite, then the usual multivariate central limit theorem implies that

$$\frac{1}{\sqrt{n}} \sum_{i=1}^n (M_i - m p_t^\nu) \xrightarrow{d} N(0, \Sigma), \text{ as } n \to \infty.$$

Let $M_i(x)$ denote the element if M_i corresponding to x. Then by definition, for all x

$$nm\hat{p}_t^\nu(x; n, m, h) = \sum_{i=1}^n M_i(x).$$

Therefore

$$\sqrt{n}m\left(\hat{p}_t^{\nu} - p_t^{\nu}\right) \xrightarrow{d} N(0,\Sigma), \text{ as } n \to \infty.$$

The dot product is continuous, so the continuous mapping theorem implies that

$$nm^2 \sum_{x\in\tilde{\mathcal{S}}} \left(\hat{p}_t^{\nu}(x;n,m,h) - p_t^{\nu}(x)\right)^2 \xrightarrow{d} N(0,\Sigma)^T N(0,\Sigma), \text{ as } n \to \infty.$$

[8, Theorem 2.1] implies that the right side has the same distribution as $\sum_{\ell=1}^{|\tilde{S}|} \lambda_\ell Z_\ell^2$. Let Σ_{xx} be the element of Σ on the diagonal corresponding to state x. Then by definition

$$\begin{aligned}\Sigma_{xx} &= \text{Var}\left[\sum_{j=1}^{m} \mathbb{1}(X_{1j} = x)\right] \\ &= \sum_{j=1}^{m} \text{Var}\left[\mathbb{1}(X_{1j} = x)\right] + 2\sum_{1\le j<k\le m} \text{Cov}\left(\mathbb{1}(X_{1j} = x), \mathbb{1}(X_{1k} = x)\right).\end{aligned}$$

$\text{Var}\left[\mathbb{1}(X_{1j} = x)\right] = p_t^{\nu}(x)(1 - p_t^{\nu}(x))$, and the covariance simplifies when we rewrite it in terms of expectations. We get

$$\Sigma_{xx} = mp_t^{\nu}(x) + m(m-1)P_\nu(X_{11}(t) = x, X_{12}(t) = x) - m^2 p_t^{\nu}(x)^2 < \infty.$$

Let x_1 and x_2 be distinct states, and let Σ_{x_1,x_2} be the element whose row and column correspond to the states x_1 and x_2, respectively. By definition

$$\begin{aligned}\Sigma_{x_1,x_2} &= \text{Cov}\left[\sum_{j=1}^{m} \mathbb{1}(X_{1j} = x_1), \sum_{j=1}^{m} \mathbb{1}(X_{1j} = x_2)\right] \\ &= \sum_{j=1}^{m}\sum_{k=1}^{m} \text{Cov}\left[\mathbb{1}(X_{1j} = x_1), \mathbb{1}(X_{1k} = x_2)\right].\end{aligned}$$

Rearrange the terms in the sum to get

$$\sum_{j=1}^{m} \text{Cov}\left[\mathbb{1}(X_{1j} = x_1), \mathbb{1}(X_{1j} = x_2)\right] + \sum_{j=1}^{m}\sum_{\substack{k=1\\k\neq j}}^{m} \text{Cov}\left[\mathbb{1}(X_{1j} = x_1), \mathbb{1}(X_{1k} = x_2)\right],$$

which is equivalent to

$$\sum_{j=1}^{m} \Big(\mathbb{E}_{\nu,0}\left[\mathbb{1}(X_{1j} = x_1)\mathbb{1}(X_{1j} = x_2)\right] - p(x_1)p(x_2)\Big) + \sum_{j=1}^{m}\sum_{\substack{k=1\\k\neq j}}^{m} \Big(\mathbb{E}_{\nu,0}\left[\mathbb{1}(X_{1j} = x_1)\mathbb{1}(X_{1k} = x_2)\right] - p(x_1)p(x_2)\Big).$$

Since $x_1 \neq x_2$, $\mathbb{1}(X_{1j} = x_1)\mathbb{1}(X_{1j} = x_2) = 0$. Also, the second expectation can be rewritten as a probability. The above expression simplifies to

$$m(m-1)P_\nu\left(X_{11}(t) = x_1, X_{12}(t) = x_2\right) - m^2 p_t^{\nu}(x_1)p_t^{\nu}(x_2) < \infty.$$

Equation (5.2) simply expresses the above results with matrix-vector notation.

If we are estimating a marginal distribution, then take $\mathcal{S}$ to be the lower dimensional space corresponding to the marginal variables. Also interpret $X(t)$ as the state vector containing only the marginal variables.

A.5. Proof of Theorem 5.2. If we write out the definition of $\hat{\Sigma}_n$ and use the fact that the trace is linear, we can see that

$$\operatorname{tr}\left(\hat{\Sigma}_n\right) = \frac{1}{n-1}\sum_{i=1}^{n}\operatorname{tr}\left(\left(M_i-\bar{M}\right)\left(M_i-\bar{M}\right)^T\right).$$

We use the cyclic property of the trace to rewrite the right side as

$$\frac{1}{n-1}\sum_{i=1}^{n}\left(M_i-\bar{M}\right)^T\left(M_i-\bar{M}\right).$$

Expanding the summands leads to

$$\frac{1}{n-1}\sum_{i=1}^{n}\left(M_i^TM_i - 2\bar{M}^TM_i + \bar{M}^T\bar{M}\right).$$

From the definition of $\bar{M}$, the above expression is equal to

$$-\frac{n}{n-1}\bar{M}^T\bar{M} + \frac{1}{n-1}\sum_{i=1}^{n}M_i^TM_i.$$

By definition, $m\hat{p}_t = \bar{M}$, therefore

$$\operatorname{tr}\left(\hat{\Sigma}_n\right) = -\frac{nm^2}{n-1}(\hat{p}_t^{\nu})^T\hat{p}_t^{\nu} + \frac{1}{n-1}\sum_{i=1}^{n}M_i^TM_i.$$

Next consider $\operatorname{tr}\left(\hat{\Sigma}_n^2\right)$. We will proceed in a similar way. By definition

$$\begin{aligned}\hat{\Sigma}_n^2 &= \frac{1}{(n-1)^2}\left[\sum_{i=1}^{n}(M_i-\bar{M})(M_i-\bar{M})^T\right]^2\\ &= \frac{1}{(n-1)^2}\sum_{i=1}^{n}\sum_{j=1}^{n}(M_i-\bar{M})(M_i-\bar{M})^T(M_j-\bar{M})(M_j-\bar{M})^T.\end{aligned}$$

The trace is linear, so

$$\begin{aligned}\operatorname{tr}\left(\hat{\Sigma}_n^2\right) &= \frac{1}{(n-1)^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\operatorname{tr}\left((M_i-\bar{M})(M_i-\bar{M})^T(M_j-\bar{M})(M_j-\bar{M})^T\right)\\ &= \frac{1}{(n-1)^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left[(M_i-\bar{M})^T(M_j-\bar{M})\right]^2.\end{aligned}$$

The last line follows from the cyclic property of the trace. When we expand the summands, the right side becomes

$$\frac{1}{(n-1)^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left[M_i^TM_j - \bar{M}^TM_i - \bar{M}^TM_j + m^2(\hat{p}_t^{\nu})^T\hat{p}_t^{\nu}\right]^2.$$

As for the claim about almost sure convergence of the traces, note that $\hat{\Sigma}_n \overset{\text{a.s.}}{\to} \Sigma$. Since matrix multiplication and the trace are continuous, the continuous mapping theorem implies the result.

A.6. Proof of Corollary 5.3. Define

$$U = \frac{\text{tr}\left(\Sigma^2\right)}{\text{tr}\left(\Sigma\right)}\chi^2_\alpha\left(\frac{\text{tr}\left(\Sigma\right)^2}{\text{tr}\left(\Sigma^2\right)}\right).$$

Since $\hat{\Sigma}_n \overset{\text{a.s.}}{\to} \Sigma$ as $n \to \infty$, the continuous mapping theorem and Lemma A.2 taken together imply that $U_n \to U$ almost surely as $n \to \infty$. Also Theorem 5.1 says that

$$nm^2 \sum_{x\in\tilde{\mathcal{S}}} \left(\hat{p}^\nu_t(x;n,m,h) - p^\nu_t(x)\right)^2 \overset{d}{\to} \sum_{\ell=1}^{|\tilde{S}|} \lambda_\ell Z^2_\ell, \text{ as } n \to \infty.$$

Therefore by Slutsky's theorem

$$\frac{nm^2 \sum_{x\in\tilde{\mathcal{S}}} \left(\hat{p}^\nu_t(x;n,m,h) - p^\nu_t(x)\right)^2}{U_n} \overset{d}{\to} \frac{\sum_{\ell=1}^{|\tilde{S}|} \lambda_\ell Z^2_\ell}{U}, \text{ as } n \to \infty,$$

which we can rewrite as

$$\lim_{n\to\infty} P_\nu\left(nm^2 \sum_{x\in\tilde{\mathcal{S}}} \left(\hat{p}^\nu_t(x;n,m,h) - p^\nu_t(x)\right)^2 \le U_n\right) = P\left(\sum_{\ell=1}^{|\tilde{S}|} \lambda_\ell Z^2_\ell \le U\right).$$

Applying the Satterthwaite approximation [43] to the right-hand side gives

$$\begin{aligned}\lim_{n\to\infty} P_\nu &\left(nm^2 \sum_{x\in\tilde{\mathcal{S}}} \left(\hat{p}^\nu_t(x;n,m,h) - p^\nu_t(x)\right)^2 \le U_n\right) \\ &\approx P\left(\frac{\text{tr}\left(\Sigma^2\right)}{\text{tr}\left(\Sigma\right)}\chi^2\left(\frac{\text{tr}\left(\Sigma\right)^2}{\text{tr}\left(\Sigma^2\right)}\right) \le U\right) \\ &= 1-\alpha.\end{aligned}$$

The result still holds for marginal distributions. We just need to remove the coordinates of $\tilde{\mathcal{S}}$ corresponding to the variables that are marginalized out.

LEMMA A.2. *Let X_θ be a family of random variables parameterized by $\theta \in \mathbb{R}$ with strictly increasing cumulative distribution functions F_θ. Suppose that for each θ, the function F_θ is continuous. Assume also that $F_\theta(x)$ is continuous in θ for each $x \in \mathbb{R}$. Then the $1-\alpha$ quantiles of F_θ are also continuous in θ for all $\alpha \in (0,1)$.*

Proof. Let $\alpha \in (0,1)$, and let $\{\theta_n\}_{n=1}^\infty$ be a sequence that converges to θ. Define q_n and q to be the $1-\alpha$ quantiles corresponding the θ_n and θ, respectively. We want to show that q_n converges to q.

Let $\varepsilon > 0$. Since $\alpha \in (0,1)$, we know that q is finite. Therefore, we can choose $\underline{q}$ and $\overline{q}$ such that

$$\underline{q} < q < \overline{q} \quad \text{and} \quad \overline{q} - \underline{q} < \varepsilon.$$

We want to show that $|q_n - q| < \varepsilon$ for all sufficiently large n, so it will suffice to prove that $\underline{q} < q_n < \overline{q}$ for all n large enough.

By assumption, $F_\theta(\underline{q})$ is continuous in θ, so

$$\lim_{n\to\infty} F_{\theta_n}(\underline{q}) = F_\theta(\underline{q}) < F_\theta(q) = 1 - \alpha = F_{\theta_n}(q_n).$$

The inequality is strict, because q is a quantile and F_θ is strictly increasing and $\underline{q} < q$. Since F_{θ_n} is non–decreasing, $q_n > \underline{q}$ for all sufficiently large n. We can use essentially the same argument to conclude that $q_n < \overline{q}$ for all n large enough. ∎

Acknowledgments. We are grateful for financial support from the Army Research Office through grant W911NF-18-1-0324.

REFERENCES

[1] D. F. Anderson, *A modified next reaction method for simulating chemical systems with time dependent propensities and delays*, The Journal of chemical physics, 127 (2007), p. 214107.
[2] D. F. Anderson, *Incorporating postleap checks in tau-leaping*, The Journal of chemical physics, 128 (2008), p. 054103.
[3] D. F. Anderson, D. Cappelletti, M. Koyama, and T. G. Kurtz, *Non-explosivity of stochastically modeled reaction networks that are complex balanced*, Bull. Math. Biol., 80 (2018), pp. 2561–2579.
[4] D. F. Anderson and T. G. Kurtz, *Error analysis of tau-leap simulation methods*, Annals of Applied Probability, 72 (2010), pp. 1947–1970.
[5] D. F. Anderson and T. G. Kurtz, *Continuous time Markov chain models for chemical reaction networks*, chapter in Design and Analysis of Biomolecular Circuits: Engineering Approaches to Systems and Synthetic Biology, H. Koeppl et al. (Editors), (2011).
[6] D. F. Anderson and T. G. Kurtz, *Stochastic analysis of biochemical systems*, Springer, 2015.
[7] D. F. Anderson, D. Schnoerr, and C. Yuan, *Time-dependent product-form poisson distributions for reaction networks with higher order complexes*. submitted, arxiv.org/abs/1904.11583, 2019.
[8] G. E. Box, *Some theorems on quadratic forms applied in the study of analysis of variance problems, i. effect of inequality of variance in the one-way classification*, The annals of mathematical statistics, (1954), pp. 290–302.
[9] H. Busch, W. Sandmann, and V. Wolf, *A numerical aggregation algorithm for the enzyme-catalyzed substrate conversion*, in International Conference on Computational Methods in Systems Biology, Springer, 2006, pp. 298–311.
[10] Y. Cao, D. T. Gillespie, and L. R. Petzold, *Efficient step size selection for the tau-leaping simulation method*, The Journal of chemical physics, 124 (2006), p. 044109.
[11] Y. Cao, D. T. Gillespie, and L. R. Petzold, *Adaptive explicit-implicit tau-leaping method with automatic tau selection*, The Journal of chemical physics, 126 (2007), p. 224101.
[12] Y. Cao, A. Terebus, and J. Liang, *Accurate chemical master equation solution using multi-finite buffers*, Multiscale Modeling & Simulation, 14 (2016), pp. 923–963.
[13] F. Didier, T. A. Henzinger, M. Mateescu, and V. Wolf, *Fast adaptive uniformization of the chemical master equation*, in 2009 International Workshop on High Performance Computational Systems Biology, IEEE, 2009, pp. 118–127.
[14] K. Ehlert and L. Loewe, *Lazy updating of hubs can enable more realistic models by speeding up stochastic simulations*, The Journal of chemical physics, 141 (2014), p. 11B617_1.
[15] S. Engblom, *Spectral approximation of solutions to the chemical master equation*, Journal of computational and applied mathematics, 229 (2009), pp. 208–221.
[16] M. Feinberg, *Lectures on chemical reaction networks*. Delivered at the Mathematics Research Center, Univ. Wisc.-Madison. Available for download at http://crnt.engineering.osu.edu/LecturesOnReactionNetworks, 1979.
[17] M. A. Gibson and J. Bruck, *Efficient exact stochastic simulation of chemical systems with many species and many channels*, The journal of physical chemistry A, 104 (2000), pp. 1876–1889.
[18] D. T. Gillespie, *A general method for numerically simulating the stochastic time evolution of coupled chemical reactions*, Journal of computational physics, 22 (1976), pp. 403–434.
[19] D. T. Gillespie, *Approximate accelerated stochastic simulation of chemically reacting systems*, The Journal of Chemical Physics, 115 (2001), pp. 1716–1733.

[20] D. T. Gillespie, A. Hellander, and L. R. Petzold, *Perspective: Stochastic algorithms for chemical kinetics*, The Journal of chemical physics, 138 (2013), p. 05B201_1.
[21] P. Glasserman, *Monte Carlo Methods in Financial Engineering*, Springer-Verlag New York, 2003.
[22] E. L. Haseltine and J. B. Rawlings, *Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics*, The Journal of chemical physics, 117 (2002), pp. 6959–6969.
[23] M. Hegland, C. Burden, L. Santoso, S. MacNamara, and H. Booth, *A solver for the stochastic master equation applied to gene regulatory networks*, Journal of computational and applied mathematics, 205 (2007), pp. 708–724.
[24] T. A. Henzinger, M. Mateescu, and V. Wolf, *Sliding window abstraction for infinite markov chains*, in International Conference on Computer Aided Verification, Springer, 2009, pp. 337–352.
[25] T. Jahnke and W. Huisinga, *Solving the chemical master equation for monomolecular reaction systems analytically*, Journal of mathematical biology, 54 (2007), pp. 1–26.
[26] T. Jahnke and T. Udrescu, *Solving chemical master equations by adaptive wavelet compression*, Journal of Computational Physics, 229 (2010), pp. 5724–5741.
[27] G. Karlebach and R. Shamir, *Modelling and analysis of gene regulatory networks*, Nature Reviews Molecular Cell Biology, 9 (2008), p. 770.
[28] V. Kazeev, M. Khammash, M. Nip, and C. Schwab, *Direct solution of the chemical master equation using quantized tensor trains*, PLoS computational biology, 10 (2014), p. e1003359.
[29] I. Kryven, S. Röblitz, and C. Schütte, *Solution of the chemical master equation by radial basis functions approximation with interface tracking*, BMC systems biology, 9 (2015), p. 67.
[30] T. G. Kurtz, *Representations of markov processes as multiparameter time changes*, The Annals of Probability, (1980), pp. 682–715.
[31] S. MacNamara, A. M. Bersani, K. Burrage, and R. B. Sidje, *Stochastic chemical kinetics and the total quasi-steady-state assumption: application to the stochastic simulation algorithm and chemical master equation*, The Journal of chemical physics, 129 (2008), p. 09B605.
[32] S. MacNamara, K. Burrage, and R. B. Sidje, *Multiscale modeling of chemical kinetics via the master equation*, Multiscale Modeling & Simulation, 6 (2008), pp. 1146–1168.
[33] S. Mauch and M. Stalzer, *Efficient formulations for exact stochastic simulation of chemical systems*, IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB), 8 (2011), pp. 27–35.
[34] H. H. McAdams and A. Arkin, *Stochastic mechanisms in gene expression*, Proceedings of the National Academy of Sciences, 94 (1997), pp. 814–819, https://doi.org/10.1073/pnas.94.3.814, http://www.pnas.org/content/94/3/814, https://arxiv.org/abs/http://www.pnas.org/content/94/3/814.full.pdf.
[35] J. M. McCollum, G. D. Peterson, C. D. Cox, M. L. Simpson, and N. F. Samatova, *The sorting direct method for stochastic simulation of biochemical systems with varying reaction execution behavior*, Computational biology and chemistry, 30 (2006), pp. 39–49.
[36] B. Munsky and M. Khammash, *The finite state projection algorithm for the solution of the chemical master equation*, The Journal of chemical physics, 124 (2006), p. 044104.
[37] J. R. Norris, *Markov Chains*, Cambridge University Press, 1997.
[38] A. B. Owen, *Monte carlo theory, methods and examples.* http://statweb.stanford.edu/~owen/mc/, 2013.
[39] S. Peleš, B. Munsky, and M. Khammash, *Reduction and solution of the chemical master equation using time scale separation and finite state projection*, The Journal of chemical physics, 125 (2006), p. 204104.
[40] R. Ramaswamy, N. González-Segredo, and I. F. Sbalzarini, *A new class of highly efficient exact stochastic simulation algorithms for chemical reaction networks*, The Journal of chemical physics, 130 (2009), p. 244104.
[41] C. V. Rao, D. M. Wolf, and A. P. Arkin, *Control, exploitation and tolerance of intracellular noise*, Nature, 420 (2002), p. 231.
[42] M. Rathinam, L. R. Petzold, Y. Cao, and D. T. Gillespie, *Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method*, The Journal of Chemical Physics, 119 (2003), pp. 12784–12794.
[43] F. E. Satterthwaite, *Synthesis of variance*, Psychometrika, 6 (1941), pp. 309–316.
[44] R. B. Sidje and H. D. Vo, *Solving the chemical master equation by a fast adaptive finite state projection based on the stochastic simulation algorithm*, Mathematical biosciences,

269 (2015), pp. 10–16.
[45] A. Slepoy, A. P. Thompson, and S. J. Plimpton, *A constant-time kinetic monte carlo algorithm for simulation of large biochemical reaction networks*, The journal of chemical physics, 128 (2008), p. 05B618.
[46] M. Thattai and A. van Oudenaarden, *Intrinsic noise in gene regulatory networks*, Proceedings of the National Academy of Sciences, 98 (2001), pp. 8614–8619, https://doi.org/10.1073/pnas.151588598, http://www.pnas.org/content/98/15/8614, https://arxiv.org/abs/http://www.pnas.org/content/98/15/8614.full.pdf.
[47] H. D. Vo and R. B. Sidje, *Improved krylov-fsp method for solving the chemical master equation*, in Proceedings of the World Congress on Engineering and Computer Science, vol. 2, 2016.
[48] H. D. Vo and R. B. Sidje, *An adaptive solution to the chemical master equation using tensors*, The Journal of chemical physics, 147 (2017), p. 044102.
[49] D. J. Wilkinson, *Stochastic modelling for systems biology*, Chapman and Hall/CRC, 2006.
[50] V. Wolf, R. Goel, M. Mateescu, and T. A. Henzinger, *Solving the chemical master equation using sliding windows*, BMC systems biology, 4 (2010), p. 42.