WEAK ERROR ANALYSIS OF NUMERICAL METHODS FOR STOCHASTIC MODELS OF POPULATION PROCESSES[*]

DAVID F. ANDERSON[†] AND MASANORI KOYAMA[†]

Abstract. The simplest, and most common, stochastic model for population processes, including those from biochemistry and cell biology, are continuous time Markov chains. Simulation of such models is often relatively straightforward, as there are easily implementable methods for the generation of exact sample paths. However, when using ensemble averages to approximate expected values, the computational complexity can become prohibitive as the number of computations per path scales linearly with the number of jumps of the process. When such methods become computationally intractable, approximate methods, which introduce a bias, can become advantageous. In this paper, we provide a general framework for understanding the weak error, or bias, induced by different numerical approximation techniques in the current setting. The analysis takes into account both the natural scalings within a given system and the step size of the numerical method. Examples are provided to demonstrate the main analytical results as well as the reduction in computational complexity achieved by the approximate methods.

1. Introduction. This paper provides a general framework for analyzing the weak error of numerical approximation techniques for the continuous time Markov chain models typically found in the study of population processes, including chemistry and cell biology. The main novelty of this work lies in how the analysis takes account of both the natural multiple scalings of a given system and the step size of the numerical method, and it is best viewed as an extension of the papers [3, 22, 23].

For $k \in \{1, \ldots, R\}$, let $\zeta_k \in \mathbb{R}^d$ denote the possible transition directions for a continuous time Markov chain, and let $\lambda'_k : \mathbb{R}^d \to \mathbb{R}$ denote the respective intensity, or propensity functions.[1] The random time change representation for the model of interest is then

$$(1.1) \qquad X(t) = X(0) + \sum_{k=1}^{R} Y_k \left(\int_0^t \lambda'_k(X(s)) ds \right) \zeta_k,$$

where the Y_k are independent, unit-rate Poisson processes. See, for example, [27], [14, Chapter 6], or the recent survey [5]. The infinitesimal generator for the model (1.1) is the operator $\mathcal{A}$ satisfying

$$(\mathcal{A}f)(x) = \sum_k \lambda'_k(x)(f(x + \zeta_k) - f(x)),$$

[†]Department of Mathematics, University of Wisconsin, Madison, WI 53706 (anderson@math.wisc.edu, koyama@math.wisc.edu). The second author's research was also supported by NSF-DMS-0805793 and NSF-DMS-1106424.

[1]In the language of probability, the functions are nearly universally termed *intensity* functions, whereas in the language of chemistry and cell biology these functions are nearly universally termed *propensity* functions. We choose the language of probability theory throughout this paper.

where $f : \mathbb{R}^d \to \mathbb{R}$ is chosen from a sufficiently large class of functions.

The problem of simulating (1.1) (and particularly of approximating expected values) seems deceivingly easy since we can simulate the continuous time Markov chains exactly.[2] Letting f be some function of the state of the system giving us some quantity of interest, we may estimate $\mathbb{E}f(X(T))$ via an ensemble average,

$$(1.2) \qquad \widehat{\mu}_n = \frac{1}{n} \sum_{i=1}^{n} f(X_{[i]}(T)),$$

where $X_{[i]}$ is the ith independent copy of (1.1). The law of large numbers then ensures that

$$(1.3) \qquad \lim_{n \to \infty} \widehat{\mu}_n = \mathbb{E}f(X(T)),$$

with a probability of one. However, it is the computational work needed to achieve an accuracy with a given tolerance, and not simply the fact that such a limit holds, that is of most interest to us.

1.1. Scalings and computational cost. The main potential problem in trying to naively apply the limit (1.3) to a given system stems from the fact that there is an expected computational cost to the generation of each independent realization, which we denote by $\overline{N}$ for now and explicitly quantify in (1.7) below. Assuming we wish to approximate $\mathbb{E}f(X(T))$ to an accuracy of $\epsilon > 0$, in terms of confidence intervals, we must generate $O(\epsilon^{-2})$ paths yielding a total computational complexity of order $O(\overline{N}\epsilon^{-2})$. This computational complexity can be substantial when $\overline{N}$ is large and/or ϵ is small.

In many models of interest, including many from cell and population biology, we do, in fact, have that $\overline{N} \gg 1$. It is therefore natural to consider how approximation schemes perform. Before considering such schemes, however, it is important (from an analytical point of view) to modify (1.1) by incorporating into the model a scaling parameter, N, that can eventually be used to quantify $\overline{N}$. The value N is usually taken to be the order of magnitude of $\max_i |X_i|$. We then scale the process by setting

$$X_i^N = N^{-\alpha_i} X_i,$$

where α_i is chosen so that X_i^N is $O(1)$. Defining $\zeta_k^N = N^{-\alpha_i} \zeta_{ki}$, the general form of the scaled model is then

$$(1.4) \qquad X^N(t) = X^N(0) + \sum_{k=1}^{R} Y_k \left(N^\gamma \int_0^t N^{c_k} \lambda_k(X^N(s)) ds \right) \zeta_k^N,$$

where γ and c_k are scalars such that

$$(1.5) \qquad |\zeta_k^N| = O(N^{-c_k}),$$

with $|\zeta_k^N| \approx N^{-c_k}$ for at least one k, and both X^N and $\lambda_k(X^N(\cdot))$ are $O(1)$. We explicitly note that we are allowing for the possibility that $|\zeta_k^N| \ll N^{-c_k}$ for some of the k. Also, note that the models (1.1) and (1.4) are equivalent in that one is simply

---

[2]This assumes that the pseudorandom numbers generated by modern computers are "random enough" to be considered truly random. We take this viewpoint throughout.

a scaled version of the other. For concreteness, the scaling thus described will be carried out explicitly for the stochastic models arising in biochemistry in section 2.

The infinitesimal generator $\mathcal{A}^N$ for the model (1.4) is

$$(1.6) \qquad (\mathcal{A}^N f)(x) = N^\gamma \sum_k N^{c_k} \lambda_k(x)(f(x + \zeta_k^N) - f(x)),$$

where $f : \mathbb{R}^d \to \mathbb{R}$ is chosen from a sufficiently large class of functions. Note that it is now natural to take

$$(1.7) \qquad \overline{N} = N^\gamma \sum_k N^{c_k}$$

as the order of magnitude for the number of steps required to generate a single path up to a time of $T > 0$.

The parameter γ of (1.4) should be thought of as representing the natural time scale of the problem with $\gamma > 0$ implying a relevant time scale smaller than one. In this case of $\gamma > 0$, the explicit numerical schemes considered in this paper are usually *not* a good choice and other methods, such as averaging techniques, are usually required in conjunction with the methods described here [5, 7, 9, 13, 25]. This fact is demonstrated by our main analytical results, which provide error bounds for the different schemes that grow exponentially in N^γ. Therefore, our main results are most useful when $\gamma \le 0$.

The model (1.4) is henceforth our main model of interest. We make the following running assumption, which, in light of the fact that both X^N and $\lambda_k(X^N(\cdot))$ are $O(1)$, is a light one.

Running assumption. The intensity functions λ_k for the scaled process X^N satisfying (1.4), together with all of their derivatives, are uniformly bounded.

The above running assumption can almost certainly be weakened to a local Lipschitz condition, in which case analytical methods similar to those found in [24] and/or [28] can be applied. Proving our main results in such generality, while possible and certainly worth doing in future work, will be significantly messier, and we feel the main points of the analysis will be lost.

We return to our problem of interest and let f be a function of the state giving a quantity of interest and consider how to approximate $\mathbb{E}f(X^N(T))$. As already discussed, the computational cost of approximating $\mathbb{E}f(X^N(T))$ to an accuracy of ϵ, in the sense of confidence intervals, using the estimator (1.2) is $O(\overline{N}\epsilon^{-2})$. Suppose now that Z^N is an approximation of X^N constructed with a time-discretization step of size $h > 0$.[3] Letting $Z_{[i]}^N$ denote independent copies of Z^N, we construct the estimator

$$(1.8) \qquad \widehat{\mu}_n = \frac{1}{n} \sum_{i=1}^n f(Z_{[i]}^N(T)).$$

Suppose that it can be shown that the approximation scheme has a weak error, or bias, of order one. That is,

$$\mathbb{E}f(X^N(T)) - \mathbb{E}f(Z^N(T)) = O(h)$$

---

[3] As the approximate process explicitly depends upon the choice of h, we could denote it as Z_h^N. However, for ease of exposition we choose to drop the h dependence from the notation.

for a suitably large class of functions f. Then, noting that

$$\mathbb{E}f(X^N(T)) - \widehat{\mu}_n = \left[\mathbb{E}f(X^N(T)) - \mathbb{E}f(Z^N(T))\right] + \left[\mathbb{E}f(Z^N(T)) - \widehat{\mu}_n\right],$$

we see that we must choose $h = O(\epsilon)$ to make the first term on the right, the bias, $O(\epsilon)$, and $n = O(\epsilon^{-2})$ to make the second term, the statistical error, have a variance of $O(\epsilon^2)$, and a standard deviation of $O(\epsilon)$. This gives a total computational complexity of $O(\epsilon^{-3})$. This will greatly lower the computational complexity of the problem, as compared with using exact sample paths, if $\epsilon^{-1} \ll \overline{N}$.

If, instead, the method for generating Z^N is second order accurate in a weak sense, that is, if

$$\mathbb{E}f(X^N(T)) - \mathbb{E}f(Z^N(T)) = O(h^2),$$

then we could choose $h = O(\epsilon^{1/2})$ to yield a bias of $O(\epsilon)$. This leads to a total computational complexity of $O(\epsilon^{-2.5})$, which for small ϵ represents a substantial improvement over using an order one method.

The above discussion points out that the key quantity to understand for a given approximation method, and the focus of this paper, is the bias, or weak error, it induces for a given function f:

$$(1.9) \qquad B_f(Z^N, x, t) \stackrel{\text{def}}{=} \mathbb{E}_x f(X^N(t)) - \mathbb{E}_x f(Z^N(t)).$$

Note that $B_f(Z^N, x, h)$ represents the local, one-step error of the method as the fixed time step is of size $h > 0$. Analyzing the bias induced by different numerical schemes is by now classical in the study of stochastic processes, with nearly all the focus falling on how the bias scales with the size of the time step, h [26]. However, it is not sufficient in the current setting to simply understand how the bias (1.9) scales with the time discretization alone. Care must also be taken to quantify how the leading order constants depend upon the natural scalings of a given system and given method, here quantified by the parameter $N > 0$. For example, if

$$\mathbb{E}f(X^N(T)) - \mathbb{E}f(Z^N(T)) = O(c_1^N h + c_2^N h^2),$$

then we wish to understand how c_1^N, c_2^N depend upon N since for a given choice of h we may have that $c_1^N h < c_2^N h^2$. In this case, the method will behave as if it is an order two method until h is reduced to the point when $c_1^N h > c_2^N h^2$, in which case it will behave like an order one method.

1.2. Notation and terminology. In this short subsection, we collect some necessary extra notation and terminology used throughout the paper. We first note that for $f : \mathbb{R}^d \to \mathbb{R}$, and any $t \geq 0$, Dynkin's formula for the process (1.4) is

$$(1.10) \qquad \mathbb{E}_x f(X^N(t)) = f(x) + \mathbb{E}_x \int_0^t \mathcal{A}^N f(X(s))ds,$$

which holds as long as the expectations exist. Similar expressions will hold for the approximate methods under consideration. Dynkin's formula will be our main analytical tool, as it will allow us to quantify the bias (1.9) for the different methods, and we therefore focus on developing compact notation for the generators of our processes.

We define the operator ∇_k^N for the kth possible transition, which we will typically call a "reaction" in keeping with the motivating application of section 2, via

$$(1.11) \qquad \nabla_k^N f(x) \stackrel{\text{def}}{=} N^{c_k}(f(x + \zeta_k^N) - f(x)).$$

Note that if f is globally Lipschitz, then $\nabla_k^N f(x)$ is uniformly bounded over k and x since $|\zeta_k^N| = O(N^{-c_k})$. We may now write (1.6) as

$$(\mathcal{A}^N f)(x) = \sum_k N^\gamma \lambda_k(x) \nabla_k^N f(x).$$

Defining the vector valued operators

$$(1.12) \qquad \lambda \overset{\text{def}}{=} [\lambda_1, \ldots, \lambda_R], \qquad \nabla^N \overset{\text{def}}{=} [\nabla_1^N, \ldots, \nabla_R^N],$$

where we recall that R is the number of reactions, we obtain

$$(\mathcal{A}^N f)(x) = (N^\gamma \lambda \cdot \nabla^N) f(x).$$

For $i \in \{1, \ldots, d\}$ and $k \in \{1, \ldots, R\}$, we let m_k satisfy

$$|\zeta_k^N| = N^{-m_k}.$$

Note that, by construction, we have $c_k \leq m_k$ for all k. Finally, we denote the jth directional derivative of f into the direction $[v_1, v_2, \ldots, v_j]$ by $f'[v_1, \ldots, v_j]$ and make the usual definition

$$(1.13) \qquad \|f\|_j \overset{\text{def}}{=} \sup_x \{f'[v_1, \ldots, v_j](x), \|v\| = 1\}.$$

1.3. Summary of main results. The following list is a summary of our main results. Technical details and assumptions have been omitted from the statements below for the sake of clarity.

1. In Theorem 4.1, we prove that for any explicit numerical scheme with a step size of $h > 0$,

$$B_f(Z^N, x, T) = O\big(Th^{-1} \sup_z |B_f(Z^N, z, h)|\big),$$

   where $B_f(Z^N, x, t)$ is the bias defined in (1.9). Thus, if the numerical scheme has a local, one-step error of $O(h^{p+1})$, then the global error is $O(h^p)$. This result is standard and should be compared to similar results in [6, 22]. It is included here since it is necessary to show that the scalings do not alter the usual result.

2. In Theorem 5.4, we prove that if Z_E^N is generated via Euler's method, also known as explicit τ-leaping in the setting of biochemistry, then

$$B_f(Z_E^N, x, h) = O(ch^2),$$

   where c is independent of N. Thus, after applying Theorem 4.1, Euler's method is proven to be an order one method, in that the leading order of the global error satisfies

$$B_f(Z_E^N, x, T) = O(ch)$$

   and decreases linearly with the step size. This fact is formally stated in Theorem 6.2.

3. In Theorem 5.5, we prove that if Z_M^N is generated via an approximate midpoint method, then

$$B_f(Z_M^N, x, h) = O(c_1^N h^2 + c_2^N h^3),$$

where c_1^N, c_2^N depend upon the natural scalings of the system, quantified here by $N > 0$. The term that dominates this error then depends upon the specific scalings of a system, encapsulated in the constants c_1^N and c_2^N, and the size of the time discretization h. Theorem 4.1 then implies

$$B_f(Z_M^N, x, T) = O(c_1^N h + c_2^N h^2),$$

and the midpoint method will sometimes behave like a first order method, and other times it will behave like a second order method. This fact is formally stated in Theorem 6.3. A transition point, in terms of N and h, for this change in behavior is also provided.

4. In Theorem 5.6, we prove that if Z_{trap}^N is generated via the weak trapezoidal method, which was originally formulated in the diffusive setting [6] and is extended to the discrete setting in section 3, then

$$B_f(Z_{trap}^N, x, h) = O(ch^3),$$

where c is independent of N. Thus, after applying Theorem 4.1, the weak trapezoidal method is proven to be a second order method, in that the leading order of the global error satisfies

$$B_f(Z_{trap}^N, x, T) = O(ch^2)$$

and decreases quadratically with the step size. This fact is formally stated in Theorem 6.5.

1.4. Context. We attempt to put the present work in the correct historical context. In [3], Anderson, Ganguly, and Kurtz provided the first error analysis of different approximation techniques that incorporated the natural scalings of the system (1.1) into the analysis. Specifically, they considered models satisfying the "classical scaling," which using our present terminology corresponds with $\gamma = 0$, $c_k \equiv 1$, and $\alpha_i \equiv 1$. They further coupled the time discretization to the scaling, thereby ensuring h was always in a useful regime, and derived results for both the weak and the strong errors of Euler's method and the midpoint method. They proved that, in this specific setting, Euler's method is an order one method in both a weak and a strong (in the L^1 norm) sense. They proved that the strong error of the midpoint method falls between orders one and two (see [3] for precise statements), and that the leading order term of the weak error of the midpoint method scales quadratically with the step size.

In [22], it was shown by Hu, Li, and Min that the $O(h^2)$ weak convergence rate of the midpoint method given in [3] depended intimately on the coupling between the time discretization and the scaling parameter of the system. It was this particular observation that in large part motivates the present work, as we wish to provide a general analytical rate of convergence for the different relevant methods in the most general possible scaling regime and in which the time-discretization parameter is independent of the natural scalings.

In [6], the weak trapezoidal algorithm was introduced in the context of SDEs driven by Brownian motions. There, it was shown to be an easy to implement method that is second order accurate in a weak sense. Further, it has the nice property that no costly derivatives need be computed during the course of the simulation. In [23], it was pointed out that since the motivation for the original weak trapezoidal algorithm comes from viewing SDEs as driven by space-time Wiener processes, the same algorithm can work in the present jump setting by viewing the driving forces

as space-time Poisson processes. This observation was also made independently in an earlier version of the present paper.

It is worth pointing out that there are at least two other trapezoidal-type algorithms in the literature pertaining to models of stochastic chemical kinetics. These are the implicit and explicit trapezoidal methods of [11]. These methods were explicitly developed to give better stability than the usual methods, and so they do not exhibit better convergence than Euler's method.

1.5. Paper outline. The remainder of the paper is organized as follows. In section 2, we show how the basic models considered in this paper, (1.1) and (1.4), arise naturally in biochemistry, which is the main area of motivation for this work. This section can safely be skipped by anyone not interested in that application. In section 3, we discuss numerical methods for the models under consideration, including both exact and approximate schemes. In section 4, we prove Theorem 4.1, as stated loosely above, and relevant corollaries. In section 5, we prove Theorems 5.4, 5.5, and 5.6, each stated loosely above, providing the local, one-step errors induced by the approximate schemes considered here. In section 6, we provide bounds on the semigroup operator of the exact process X^N, yielding the final piece to the global analysis of the weak error of the different methods. We also briefly discuss stability concerns in section 6. In section 7, we provide relevant examples.

2. Motivating systems: Biochemical reaction networks. In this section, we build the relevant models (1.1) and (1.4) used in the study of stochastically modeled biochemical reaction networks. We feel it is worthwhile to include this section, as this is the area of main motivation for the present work. However, it can safely be skipped by those wishing to simply see the mathematical analysis and not the areas of application.

2.1. The unscaled model. A chemical reaction network is a dynamical system involving multiple reactions and chemical species. The simplest stochastic models of such networks treat the system as a continuous time Markov chain with the state, $X \in \mathbb{Z}_{\geq 0}^d$, giving the number of molecules of each species and with reactions modeled as possible transitions of the chain.

An example of a chemical reaction is

$$2S_1 + S_2 \rightarrow S_3,$$
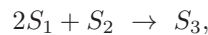
where we would interpret the above as saying two molecules of type S_1 combine with a molecule of type S_2 to produce a molecule of type S_3. The S_i are called chemical *species*. Letting

$$\nu_1 = \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix}, \quad \nu_1' = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad \text{and} \quad \zeta_1 = \nu_1' - \nu_1 = \begin{pmatrix} -2 \\ -1 \\ 1 \end{pmatrix},$$

we see that every instance of the reaction changes the state of the system by the addition of ζ_1. Here the subscript "1" is used to denote the first (and in this case only) reaction of the system.

In the general setting we denote the number of species by d, and for $i \in \{1, \ldots, d\}$ we denote the ith species by S_i. We then consider a finite set of R reactions, where the model for the kth reaction is determined by

(i) a vector of inputs ν_k specifying the number of molecules of each chemical species that are consumed in the reaction,

(ii) a vector of outputs ν_k' specifying the number of molecules of each chemical species that are created in the reaction, and

(iii) a function of the state λ_k' that gives the transition intensity, rate, or propensity at which the reaction occurs.

Specifically, if we denote the state of the system at time t by $X(t) \in \mathbb{Z}^d$, and if the kth reaction occurs at time t, we update the state by the addition of the *reaction vector*

$$\xi_k \stackrel{\text{def}}{=} \nu_k' - \nu_k$$

and the new state becomes $X(t) = X(t-) + \xi_k$. For the standard Markov chain model, the number of times that the kth reaction occurs by time t can be represented by the counting process

$$R_k(t) = Y_k\left(\int_0^t \lambda_k'(X(s))ds\right),$$

where the Y_k are independent, unit-rate Poisson processes [27], [14, Chapter 6]. The state of the system then satisfies

$$X(t) = X(0) + \sum_k Y_k\left(\int_0^t \lambda_k'(X(s))ds\right)\xi_k,$$

which was (1.1) in the introduction. The above formulation is termed a random time change representation and is equivalent to the chemical master equation representation found in much of the biology and chemistry literature, where the master equation is Kolmogorov's forward equation in the terminology of probability.

A common choice of intensity function for chemical reaction systems is that of mass action kinetics. Under mass action kinetics, the intensity function for the kth reaction is

(2.1) $$\lambda_k'(x) = \kappa_k' \prod_{i=1}^d \frac{x_i!}{(x_i - \nu_{ki})!},$$

where ν_{ki} is the ith component of ν_k.

Example 1. To solidify notation, we consider the network

$$S_1 \underset{\kappa_2}{\overset{\kappa_1}{\rightleftarrows}} S_2, \qquad 2S_2 \overset{\kappa_k}{\rightarrow} S_3,$$
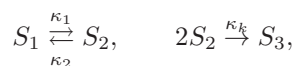
where we have placed the rate constants κ_k above or below their respective reactions. For this example, (1.1) is

$$X(t) = X(0) + Y_1\left(\int_0^t \kappa_1 X_1(s)ds\right)\begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix} + Y_2\left(\int_0^t \kappa_2 X_2(s)ds\right)\begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$$

$$+ Y_3\left(\int_0^t \kappa_3 X_2(s)(X_2(s) - 1)ds\right)\begin{bmatrix} 0 \\ -2 \\ 1 \end{bmatrix}.$$

Defining $\zeta_1 = [-1, 1, 0]^T$, $\zeta_2 = [1, -1, 0]^T$, and $\zeta_3 = [0, -2, 1]^T$, the generator $\mathcal{A}$ satisfies

$$(\mathcal{A}f)(x) = \kappa_1 x_1(f(x+\zeta_1) - f(x)) + \kappa_2 x_2(f(x+\zeta_2) - f(x)) + \kappa_3 x_2(x_2 - 1)(f(x+\zeta_3) - f(x)).$$

2.2. Scaled biochemical models. The scaling described below has been used previously in at least [4, 5, 7, 25]. We emphasize that the scaling is an analytical tool used to understand the behavior of the different processes, and that the actual simulations using the different methods make no use of, nor have need for, an understanding of N, α, or the β_k.

Let $N \gg 1$ be a natural parameter of the system, perhaps the abundance of the species with the highest number of molecules. Assume that the system satisfies (1.1) with λ'_k determined via mass action kinetics (2.1), and with $\zeta_k \in \mathbb{Z}^d$ representing the reaction vectors described in section 2.1. For each species i, define the *normalized abundance* (or, simply, the abundance) by

$$X_i^N(t) = N^{-\alpha_i} X_i(t),$$

where $\alpha_i \geq 0$ should be selected so that X_i^N is $O(1)$. Here X_i^N may be the species number ($\alpha_i = 0$), the species concentration, or something else.

Since the rate constants may also vary over several orders of magnitude, we write $\kappa'_k = \kappa_k N^{\beta_k}$, where the β_k are selected so that $\kappa_k = O(1)$ (recall that κ'_k is the original system parameter). Note that while the α_i are nonnegative if N is chosen to be the abundance of the species with the highest number of molecules, β_k can be positive, negative, or zero.

Under the mass action kinetics assumption, we always have that $\lambda'_k(X(s)) = N^{\beta_k + \nu_k \cdot \alpha} \lambda_k(X^N(s))$, where λ_k denotes the deterministic mass action kinetics with rate constants κ_k [5, 7, 25]. Our model has therefore become

$$(2.2) \quad X^N(t) = X^N(0) + \sum_k Y_k \left(\int_0^t N^{\beta_k + \nu_k \cdot \alpha} \lambda_k(X^N(s)) ds \right) \zeta_k^N, \quad i \in \{1, \dots, d\},$$

where $\zeta_{ki}^N \stackrel{\text{def}}{=} N^{-\alpha_i} \zeta_{ki}$. To quantify the natural time scale of the system, define $\gamma \in \mathbb{R}$ via

$$\gamma = \max_{\{i,k \,:\, \zeta_{ki}^N \neq 0\}} \{\beta_k + \nu_k \cdot \alpha - \alpha_i\},$$

where we recall that ν_k is the source vector for the kth reaction. Letting

$$c_k = \beta_k + \nu_k \cdot \alpha - \gamma,$$

for each k, the model (2.2) is seen to be exactly (1.4).

Remark 1. If $\beta_k + \nu_k \cdot \alpha = \alpha_i = 1$ for all i, k in (2.2), in which case $\gamma = 0$, then we have what is typically called the *classical scaling*. It was specifically this scaling that was used in the analyses of the Euler and midpoint methods found in [3, 22, 23]. In this case it is natural to consider X^N as a vector whose ith component gives the concentration, in moles per unit volume, of the ith species.

Example 2. As an instructive example, consider the system

$$S_1 \underset{100}{\overset{100}{\rightleftarrows}} S_2$$
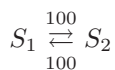
with $X_1(0) = X_2(0) = 10{,}000$. In this case, it is natural to take $N = 10{,}000$ and $\alpha_1 = \alpha_2 = 1$. As the rate constants are $100 = \sqrt{10{,}000}$, we take $\beta_1 = \beta_2 = 1/2$ and find that $\gamma = 1/2$. The equation governing the normalized process X_1^N is

$$X_1^N(t) = X_1^N(0) - Y_1 \left(N^{1/2} N \int_0^t X_1^N(s) ds \right) \frac{1}{N} + Y_2 \left(N^{1/2} N \int_0^t (2 - X_1^N(s)) ds \right) \frac{1}{N},$$

where we have used that $X_1^N + X_2^N \equiv 2$.

3. Numerical methods.

3.1. Exact methods. As already discussed in the introduction, because we are considering continuous time Markov chains, there are a number of numerical methods available for the generation of exact sample paths for the model (1.1) or the equivalent model (1.4). All are examples of discrete event simulation [21]. In the language of biochemistry these methods include the stochastic simulation algorithm, best known as Gillespie's algorithm in this setting [17, 18], the first reaction method [17], and the next reaction method [1, 16]. All such algorithms perform the same two basic steps multiple times until a sample path is produced over a desired time interval: conditioned on the current state of the system, both (i) the amount of time that passes until the next reaction takes place, Δt, is computed and (ii) the specific reaction that has taken place is found. Note that Δt is an exponential random variable with a parameter of $\sum_k \lambda_k(X(t))$. Therefore, if

$$(3.1) \qquad \sum_k \lambda_k(X(t)) \approx \overline{N} \gg 1 \qquad \text{so that} \qquad \mathbb{E}\Delta t = \frac{1}{\sum_k \lambda_k(X(t))} \approx \frac{1}{\overline{N}} \ll 1,$$

then the runtime needed to produce a single exact sample path may be prohibitive when coupled with Monte Carlo techniques, and approximate methods may be desirable.

3.2. Approximate methods. There will be times when we will wish to discuss an arbitrary approximation to X or X^N, and other times we will wish to consider specific approximations. When we consider an arbitrary approximation we will simply denote the approximation by Z or Z^N. When we distinguish the Euler, midpoint, and weak trapezoidal approximations, the main approximations under consideration here, we will denote by Z_E, Z_M, and Z_{trap} the respective approximations to X and by Z_E^N, Z_M^N, and Z_{trap}^N the respective approximations to X^N. Throughout, our time-discretization parameter will be denoted by $h > 0$.[4]

3.2.1. Euler's method. The Euler approximation, Z_E, to the model (1.1) is the solution to

$$(3.2) \qquad Z_E(t) = Z_E(0) + \sum_k Y_k \left(\int_0^t \lambda_k'(Z_E \circ \eta(s))ds \right) \zeta_k,$$

where $\eta(s) \overset{\text{def}}{=} \lfloor s/h \rfloor h$, and all other notation is as before. Note that $Z_E(\eta(s)) = Z_E(t_n)$ if $t_n \le s < t_{n+1}$. The basic algorithm for the simulation of (3.2) up to a time of $T > 0$ is the following. For $x \ge 0$ we will write Poisson(x) for a Poisson random variable with a parameter of x.

ALGORITHM 1 (Euler's method). *Fix $h > 0$. Set $Z_E(0) = x_0$, $t_0 = 0$, and $n = 0$, and repeat the following until $t_{n+1} = T$:*
 (i) *Set $t_{n+1} = t_n + h$. If $t_{n+1} \ge T$, set $t_{n+1} = T$ and $h = T - t_n$.*
 (ii) *For $k \in \{1, \dots, R\}$, let $\Lambda_k = Poisson(\lambda_k'(Z_E(t_n))h)$ be independent of each other and all previous random variables.*
 (iii) *Set $Z_E(t_{n+1}) = Z_E(t_n) + \sum_k \Lambda_k \zeta_k$.*
 (iv) *Set $n \leftarrow n + 1$.*

---

[4]Historically, the time-discretization parameter for the methods described in this paper have been τ, thus giving these methods the general name "τ-leaping methods." We choose to break from this tradition and denote our time step by h so as not to confuse τ with a stopping time.

The above algorithm is termed *explicit tau-leaping* in the biology and biochemistry literature [19]. Several improvements and modifications have been made to the basic algorithm described above over the years in the context of biochemical processes. Many of the improvements are concerned with how to choose the step size adaptively [10, 20] and/or how to ensure that population values do not go negative during the course of a simulation [2, 8, 12], which is a relevant issue, as population processes have a natural nonnegativity constraint. For the simulations carried out in section 7, we choose to simply keep a fixed step size and set any species that goes negative in the course of a jump to zero.

Defining the operator

$$(3.3) \qquad (\mathcal{B}_z f)(x) \stackrel{\text{def}}{=} \sum_k \lambda'_k(z)(f(x + \zeta_k) - f(x)),$$

we see that for $t > 0$

$$(3.4) \qquad \mathbb{E}f(Z_E(t)) = \mathbb{E}f(Z_E \circ \eta(t)) + \mathbb{E} \int_{\eta(t)}^t (\mathcal{B}_{Z_E \circ \eta(t)} f)(Z_E(s)) ds,$$

as long as the expectations exist. The scaled version of (3.2), which is an approximation to X^N satisfying (1.4), is

$$(3.5) \qquad Z_E^N(t) = Z_E^N(0) + \sum_k Y_k \left(N^\gamma \int_0^t N^{c_k} \lambda_k(Z_E^N \circ \eta(s)) ds \right) \zeta_k^N,$$

where all notation is as before. Define the operator $\mathcal{B}_z^N$ by

$$(3.6) \qquad \mathcal{B}_z^N f(x) \stackrel{\text{def}}{=} (N^\gamma \lambda(z) \cdot \nabla^N) f(x).$$

If Z_E^N satisfies (3.5), then for all $t > 0$

$$\mathbb{E}f(Z_E^N(t)) = \mathbb{E}f(Z_E^N(\eta(t))) + \mathbb{E} \int_{\eta(t)}^t (\mathcal{B}_{Z_E^N(\eta(t))}^N f)(Z_E^N(s)) ds,$$

as long as the expectations exist.

3.2.2. Approximate midpoint method. A midpoint-type method was first described in [19][5] and analyzed in [3, 22]. Define the function

$$\rho(z) \stackrel{\text{def}}{=} z + \frac{1}{2}h \sum_k \lambda'_k(z)\zeta_k,$$

which computes an approximate midpoint for the system (1.1) assuming the state of the system is z and the time step is h. Then define Z_M to be the process that satisfies

$$(3.7) \qquad Z_M(t) = Z_M(0) + \sum_k Y_k \left(\int_0^t \lambda'_k \circ \rho(Z_M \circ \eta(s)) ds \right) \zeta_k.$$

The basic algorithm for the simulation of (3.7) up to a time of $T > 0$ is the following. Note that only step (ii) changes from Euler's method.

ALGORITHM 2 (midpoint method). *Fix $h > 0$. Set $Z_M(0) = x_0$, $t_0 = 0$, and $n = 0$, and repeat the following until $t_{n+1} = T$:*

---

[5]The midpoint method detailed in [19] is actually a slight variant of the method described here. In [19] the approximate midpoint, called $\rho(z)$ above, is rounded to the nearest integer value.

(i) *Set $t_{n+1} = t_n + h$. If $t_{n+1} \geq T$, set $t_{n+1} = T$ and $h = T - t_n$.*
(ii) *For $k \in \{1, \ldots, R\}$, let $\Lambda_k = Poisson(\lambda_k' \circ \rho(Z_M(t_n))h)$ be independent of each other and all previous random variables.*
(iii) *Set $Z_M(t_{n+1}) = Z_M(t_n) + \sum_k \Lambda_k \zeta_k$.*
(iv) *Set $n \leftarrow n + 1$.*

For $\mathcal{B}_z$ defined via (3.3), any $t > 0$, and Z_M satisfying (3.7), we have

$$\mathbb{E}f(Z_M(t)) = \mathbb{E}f(Z_M \circ \eta(t)) + \mathbb{E} \int_{\eta(t)}^{t} (\mathcal{B}_{\rho \circ Z_M \circ \eta(t)} f)(Z_M(s)) ds,$$

as long as the expectations exist. The scaled version of (3.7), which is an approximation to X^N satisfying (1.4), is

$$(3.8) \qquad Z_M^N(t) = Z_M^N(0) + \sum_k Y_k \left(N^\gamma \int_0^t N^{c_k} \lambda_k \circ \rho(Z_M^N \circ \eta(s)) ds \right) \zeta_k^N,$$

where now

$$\rho(z) = z + \frac{1}{2} h N^\gamma \sum_k N^{c_k} \lambda_k(z) \zeta_k^N.$$

While we should write ρ^N in the above, we repress the "N" in this case for ease of notation. For $\mathcal{B}_z^N$ defined via (3.6) and Z_M^N satisfying (3.8), we have

$$\mathbb{E}f(Z_M^N(t)) = \mathbb{E}f(Z_M^N(\eta(t))) + \mathbb{E} \int_{\eta(t)}^{t} (\mathcal{B}_{\rho(Z_M^N \circ \eta(t))}^N f)(Z_M^N(s)) ds$$

for all $t > 0$, as long as the expectations exist.

3.2.3. The weak trapezoidal method. We will now describe a trapezoidal-type algorithm for approximating the solutions of (1.1) and/or (1.4). The method was originally introduced in the work of Anderson and Mattingly in the diffusive setting, where it is best understood by using a pathwise representation that incorporates space-time white noise processes; see [6]. It has independently been extended to the current jump setting in [23], where it was studied in the classical scaling ($\gamma = 0$, $\alpha_i \equiv 1$, $c_k \equiv 1$, with the step size coupled to the system size similarly to the analysis in [3]).

In the algorithm below, which simulates a path up to a time $T > 0$, it is notationally convenient to define $[x]^+ = x \vee 0 = \max\{x, 0\}$.

ALGORITHM 3 (weak trapezoidal method). *Fix $h > 0$. Set $Z(0) = x_0$, $t_0 = 0$, and $n = 0$. Fixing a $\theta \in (0, 1)$, we define*

$$(3.9) \qquad \xi_1 \stackrel{def}{=} \frac{1}{2} \frac{1}{\theta(1 - \theta)} \qquad and \qquad \xi_2 \stackrel{def}{=} \frac{1}{2} \frac{(1 - \theta)^2 + \theta^2}{\theta(1 - \theta)}.$$

We repeat the following steps until $t_{n+1} = T$, in which we first compute a θ-midpoint y^* and then the new value $Z_{trap}(t_{n+1})$:
(i) *Set $t_{n+1} = t_n + h$. If $t_{n+1} \geq T$, set $t_{n+1} = T$ and $h = T - t_n$.*
(ii) *For $k \in \{1, \ldots, R\}$, let $\Lambda_{k,1} = Poisson(\lambda_k'(Z_{trap}(t_n))\theta h)$ be independent of each other and all previous random variables.*
(iii) *Set $y^* = Z_{trap}(t_n) + \sum_k \Lambda_{k,1} \zeta_k$.*

(iv) *For $k \in \{1, \ldots, R\}$, let $\Lambda_{k,2} = Poisson([\xi_1 \lambda_k'(y^*) - \xi_2 \lambda_k'(t_n)]^+ (1 - \theta)h)$ be independent of each other and all previous random variables.*

(v) *Set $Z_{trap}(t_{n+1}) = y^* + \sum_k \Lambda_{k,2} \zeta_k$.*

(vi) *Set $n \leftarrow n + 1$.*

Remark 2. Notice that on the $(n + 1)$st step, y^* is the Euler approximation to $X(nh + \theta h)$ starting from $Z_{trap}(t_n)$ at time nh.

Remark 3. Notice that for all $\theta \in (0, 1)$ one has $\xi_1 > \xi_2$ and $\xi_1 - \xi_2 = 1$.

We define the operator $\mathcal{B}_{z_1, z_2}$ by

$$(\mathcal{B}_{z_1, z_2} f)(x) \overset{\text{def}}{=} \sum_k [\xi_1 \lambda_k'(z_1) - \xi_2 \lambda_k'(z_2)]^+ (f(x + \zeta_k) - f(x)).$$

Then, for $\eta(t) \leq t \leq \eta(t) + \theta h$, the process Z_{trap} satisfies

$$\mathbb{E}f(Z_{trap}(t)) = \mathbb{E}f(Z_{trap}(\eta(t))) + \mathbb{E} \int_{\eta(t)}^t (\mathcal{B}_{Z_{trap}(\eta(t))} f)(Z_{trap}(s)) ds,$$

where we recall that $\mathcal{B}_z$ is defined via (3.3), and for $\eta(t) + \theta h \leq t \leq \eta(t) + h$, the process Z_{trap} satisfies

$$\mathbb{E}f(Z_{trap}(t)) = \mathbb{E}f(Z_{trap}(\eta(t) + \theta h)) + \mathbb{E} \int_{\eta(t) + \theta h}^t (\mathcal{B}_{Z_{trap}(\eta(t) + \theta h), Z_{trap}(\eta(t))} f)(Z_{trap}(s)) ds.$$

Finally, define the operator $\mathcal{B}_{z_1, z_2}^N$ by

$$(\mathcal{B}_{z_1, z_2}^N f)(x) \overset{\text{def}}{=} (N^\gamma [\xi_1 \lambda(z_1) - \xi_2 \lambda(z_2)]^+ \cdot \nabla^N) f(x),$$

where for some $\theta \in (0, 1)$, ξ_1 and ξ_2 satisfy (3.9), and for $v \in \mathbb{R}^d$ the ith component of v^+ is $[v_i]^+ = \max\{v_i, 0\}$. Then, if Z_{trap}^N represents the approximation to (1.4) via the weak trapezoidal method, for $\eta(t) \leq t < \eta(t) + \theta h$

$$\mathbb{E}f(Z_{trap}^N(t)) = \mathbb{E}f(Z_{trap}^N(\eta(t))) + \mathbb{E} \int_{\eta(t)}^t (\mathcal{B}_{Z_{trap}^N(\eta(t))}^N f)(Z_{trap}^N(s)) ds,$$

whereas for $\eta(t) + \theta h \leq t < \eta(t) + h$

$$\mathbb{E}f(Z_{trap}^N(t)) = \mathbb{E}f(Z_{trap}^N(\eta(t) + \theta h)) + \mathbb{E} \int_{\eta(t) + \theta h}^t (\mathcal{B}_{Z_{trap}^N(\eta(t) + \theta h), Z_{trap}^N(\eta(t))}^N f)(Z_{trap}^N(s)) ds.$$

4. Global error from local error. Throughout the section, we will denote the vector valued process whose ith component satisfies (1.4) by X^N and denote an arbitrary approximate process via Z^N. Also, we define the following semigroup operators acting on $f \in C_0(\mathbb{R}^d, \mathbb{R})$:

(4.1)
$$\mathcal{P}_t f(x) \overset{\text{def}}{=} \mathbb{E}_x f(X^N(t)),$$
$$P_t f(x) \overset{\text{def}}{=} \mathbb{E}_x f(Z^N(t)),$$

where for ease of notation we choose not to incorporate the notation N into either $\mathcal{P}_t$ or P_t. Returning to the notation introduced in section 1, we note that

$$B_f(Z^N, x, t) = \mathbb{E}_x f(X^N(t)) - \mathbb{E}_x f(Z^N(t)) = (\mathcal{P}_t - P_t)f(x).$$

We may therefore interpret the difference between the above two operators, for $t \in [0, T]$, as the weak error, or bias, of the approximate process Z^N on the interval $[0, T]$. As $h > 0$ is our time step, we note that $B_f(Z^N, x, h) = (\mathcal{P}_h - P_h)f(x)$ is the one-step local error.

DEFINITION 4.1. *Let n be an arbitrary nonnegative integer, and let $\mathcal{M}$ be an m-dimensional vector of $C(\mathbb{R}^d, \mathbb{R})$ valued operators on $C(\mathbb{R}^d, \mathbb{R})$, with its ℓth coordinate denoted by $\mathcal{M}_\ell$. Then we define*

$$\|f\|_n^{\mathcal{M}} = \sup\left\{\left\|\left(\prod_{i=1}^p \mathcal{M}_{\ell_i}\right)f\right\|_\infty, 1 \le \ell_i \le m, \ p \le n\right\}.$$

For example, if $j, k, \ell \in \{1, \ldots, R\}$ then

$$|(\nabla_j^N \nabla_k^N \nabla_\ell^N f)(x)| \le \|f\|_3^{\nabla^N},$$

where we recall that ∇^N is defined in (1.12). Note that, for any $\mathcal{M}$,

(4.2) $$\|f\|_0^{\mathcal{M}} = \|f\|_0 = \|f\|_\infty.$$

Also note that, by definition, for $n \ge 0$

$$\|f\|_n^{\mathcal{M}} \le \|f\|_{n+1}^{\mathcal{M}}.$$

DEFINITION 4.2. *Suppose $\mathcal{M} : C(\mathbb{R}^d, \mathbb{R}) \to C(\mathbb{R}^d, \mathbb{R}^R)$ and $Q : C(\mathbb{R}^d, \mathbb{R}) \to C(\mathbb{R}^d, \mathbb{R})$ are operators. Then define*

$$\|Q\|_{j \to \ell}^{\mathcal{M}} \stackrel{def}{=} \sup_{f \in C^j, f \ne 0} \frac{\|Qf\|_\ell^{\mathcal{M}}}{\|f\|_j^{\mathcal{M}}}.$$

Note that as stated in the introduction, the purpose of this paper is to derive bounds for the global weak error of the different approximate processes, which, due to (4.2), consists of deriving bounds for $\|(P_h^n - \mathcal{P}_{nh})\|_{m \to 0}^{\mathcal{M}}$, for an appropriately defined $\mathcal{M}$ and a reasonable choice of $m \ge 0$. Theorem 4.1 quantifies how the global error $\|(P_h^n - \mathcal{P}_{nh})\|_{m \to 0}^{\mathcal{M}}$ can be bounded using the one-step local error $\|P_h - \mathcal{P}_h\|_{m \to 0}^{\mathcal{M}}$. In section 5, we will derive the requisite bounds for the local error for each of the three methods.

THEOREM 4.1. *Let $\mathcal{M}$ be a $C(\mathbb{R}^d, \mathbb{R}^R)$ valued operator on $C(\mathbb{R}^d, \mathbb{R})$. Then for any $n, m \ge 0$, and $h > 0$*

$$\|(P_h^n - \mathcal{P}_{nh})\|_{m \to 0}^{\mathcal{M}} = O\left(n \ \|P_h - \mathcal{P}_h\|_{m \to 0}^{\mathcal{M}} \max_{\ell \in \{1, \ldots, n\}} \{\|\mathcal{P}_{\ell h}\|_{m \to m}^{\mathcal{M}}\}\right).$$

Proof. Let $f \in C_0(\mathbb{R}^d, \mathbb{R})$. Note that, since $\|g\|_0 = \|g\|_0^{\mathcal{M}}$ for any g,

$$\|P_h^{j-1}\|_{0 \to 0}^{\mathcal{M}}\|P_h - \mathcal{P}_h\|_{m \to 0}^{\mathcal{M}} = \|P_h^{j-1}\|_{0 \to 0}\|P_h - \mathcal{P}_h\|_{m \to 0}^{\mathcal{M}}.$$

With this in mind,

$$\|(P_h^n - \mathcal{P}_{nh})f\|_0 = \left\|\sum_{j=1}^n (P_h^j \mathcal{P}_{h(n-j)} - P_h^{j-1} \mathcal{P}_{h(n-j+1)})f\right\|_0$$

$$\le \sum_{j=1}^n \|P_h^{j-1}(P_h - \mathcal{P}_h)\mathcal{P}_{h(n-j)}f\|_0$$

$$\le \sum_{j=1}^n \|P_h^{j-1}\|_{0 \to 0}\|P_h - \mathcal{P}_h\|_{m \to 0}^{\mathcal{M}}\|\mathcal{P}_{h(n-j)}\|_{m \to m}^{\mathcal{M}}\|f\|_m^{\mathcal{M}}.$$

Since P_h is a contraction, i.e., $\|P_h\|_{0\to 0} \le 1$, the result is shown. $\square$

The following result, where ∇^N replaces $\mathcal{M}$ in Theorem 4.1, is now immediate.

COROLLARY 4.2. *Under the same assumptions of Theorem* 4.1 *and with* $f \in C_0^m(\mathbb{R}^d, \mathbb{R})$,

$$\|(P_h^n - \mathcal{P}_{nh})f\|_0^{\nabla^N} = O\big(n\|P_h - \mathcal{P}_h\|_{m\to 0}^{\nabla^N} \max_{\ell \in \{1,\ldots,n\}}\{\|\mathcal{P}_{\ell h}f\|_m^{\nabla^N}\}\big).$$

The following generalization, which allows for variable step sizes, is straightforward.

COROLLARY 4.3. *For* $f \in C_0^m(\mathbb{R}^d, \mathbb{R})$

$$\|\mathbb{E}_x f(Z_{t_n}) - \mathbb{E}_x f(X_{t_n})\|_\infty = O\big(n \max_{i=1,\ldots,n}\{\|P_{h_i} - \mathcal{P}_{h_i}\|_{m\to 0}^{\nabla^N}\} \max_{\ell \in \{1,\ldots,n\}}\{\|\mathcal{P}_{t_\ell}f\|_m^{\nabla^N}\}\big).$$

Thus, once we compute the local one-step error $\|P_h - \mathcal{P}_h\|_{m\to 0}^{\nabla^N}$ for an approximate process, we have a bound on the global weak error that depends only on the semigroup $\mathcal{P}_t$ of the original process. We will delay discussion of $\|\mathcal{P}_t f\|_m^{\nabla^N}$ for now, as this term is independent of the approximate process. Instead, in the next section we provide bounds for $\|P_h - \mathcal{P}_h\|_{m\to 0}^{\nabla^N}$ for each of the three methods described in section 3.

5. Local errors. Section 5.1 will present some necessary propositions and lemmas. Sections 5.2, 5.3, and 5.4 will present the local analyses of the Euler, midpoint, and weak trapezoidal methods, respectively.

5.1. Analytical tools.
PROPOSITION 5.1. *Let* $f \in C_0^1(\mathbb{R}^d, \mathbb{R}^R)$. *For any* $k \in \{1,\ldots,R\}$

$$\nabla_k^N f \in O(N^{c_k - m_k}\|f\|_1) \subset O(1).$$

In particular, $N^{-c_k}\nabla_k^N f$ *is bounded.*

Proof. The result follows from the fact that for any $w \in \mathbb{R}^d$

$$|f(x+w) - f(x)| \le |w|\|f\|_1.   \square$$

Define, for any multisubset I of $\{1,\ldots,R\}$,

$$\nabla_I^N f \stackrel{\text{def}}{=} \left\{\left(\prod_{i=1}^{|I|} \nabla_{\ell_i}^N\right)f\right\}$$

so that

$$\|f\|_n^{\nabla^N} = \sup_{|I|\le n} \|\nabla_I^N f\|_\infty.$$

PROPOSITION 5.2. *Let* $f \in C_0^j(\mathbb{R}^d, \mathbb{R}^R)$. *Then*

$$\|f\|_j^{\nabla^N} = O(\|f\|_j).$$

Proof. The case $j = 1$ follows from Proposition 5.1. Now consider $\nabla_I^N f(x)$ for a multiset I of $\{1,\ldots,R\}$, with $|I| = j \ge 2$. If $m_k > 0$ for all $k \in I$, the statement is clear. If, on the other hand, $m_k = 0$ for some $k \in I$, then for this specific k we have $c_k \le 0$ and

$$\|\nabla_I^N f\|_\infty \le 2N^{c_k}\|\nabla_{I\backslash k}^N f\|_\infty = O(\|f\|_{j-1}) = O(\|f\|_j),$$

where the second to last equality follows by an inductive hypothesis.   ☐

We make some definitions associated with ∇^N. Let $g : \mathbb{R}^d \to \mathbb{R}^R$. For $i, j \in \{1, \ldots, R\}$

$$[D^N g(x)]_{ij} \stackrel{\text{def}}{=} \nabla_j^N g_i(x),$$

(5.1) $$[(\nabla^N)^2]_{ij} \stackrel{\text{def}}{=} \nabla_i^N \nabla_j^N,$$

$$\text{diag}(N^c) \stackrel{\text{def}}{=} \text{diag}(N^{c_1}, \ldots, N^{c_R}).$$

Also, we define $\mathbf{1}_R$ to be the R-dimensional vector whose entries are all 1.

LEMMA 5.1 (product rule). *Let $g, q : \mathbb{R}^d \to \mathbb{R}^R$ be vector valued functions. Then*

$$\nabla_k^N (g \cdot q)(x) = (\nabla_k^N g \cdot q)(x) + (g \cdot \nabla_k^N q)(x) + N^{-c_k}(\nabla_k^N g \cdot \nabla_k^N q)(x).$$

Also,

$$\nabla^N(g \cdot q)(x) = [D^N g]^T q(x) + [D^N q]^T g(x) + \text{diag}(N^c)^{-1}([D^N g]^T \times [D^N q]^T)(x)\mathbf{1}_R f.$$

Proof. Note that, for any k,

$$\begin{aligned}
\nabla_k^N(g \cdot q)(x) &= N^{c_k}(g(x + \zeta_k^N)q(x + \zeta_k^N) - g(x)q(x)) \\
&= N^{c_k}(g(x + \zeta_k^N) - g(x))q(x) + N^{c_k}(q(x + \zeta_k^N) - q(x))g(x) \\
&\quad + N^{-c_k}N^{c_k}(q(x + \zeta_k^N) - q(x))N^{c_k}(g(x + \zeta_k^N) - g(x)) \\
&= (\nabla_k^N g) \cdot q)(x) + (\nabla_k^N q \cdot g)(x) + N^{-c_k}(\nabla_k^N g \cdot \nabla_k^N q)(x),
\end{aligned}$$

verifying the first statement. To verify the second, one simply notes that the above calculation holds for every coordinate, and the result follows after simple bookkeeping.   ☐

COROLLARY 5.3. *Let $\lambda : \mathbb{R}^d \to \mathbb{R}^R$ be a vector valued function, and let $f : \mathbb{R}^d \to \mathbb{R}$. Then*

$$\nabla_k^N(\lambda \cdot \nabla^N f)(x) = (\nabla_k^N \lambda \cdot \nabla^N)f + \lambda \cdot \nabla^N \nabla_k^N f + N^{-c_k}\nabla_k^N \lambda \cdot \nabla^N \nabla_k^N f.$$

Also,

(5.2) $$\nabla^N(\lambda \cdot \nabla^N f) = [D^N \lambda]^T \nabla^N f + [(\nabla^N)^2 f]\lambda + \text{diag}(N^c)^{-1}[D^N \lambda \times (\nabla^N)^2]\mathbf{1}_R f.$$

Proof. Simply put $g = \lambda$ and $q = \nabla^N f$, and note that ∇^2 is symmetric.   ☐

5.2. Euler's method. Throughout subsection 5.2, we let Z_E^N be the Euler approximation to X^N, and we let

$$P_{E,h}f(x) \stackrel{\text{def}}{=} \mathbb{E}_x f(Z_E^N(h)),$$

where h is the step size taken in the algorithm. Below, we will assume $h < N^{-\gamma}$, which is a natural stability condition and is discussed further in section 6.2.

THEOREM 5.4. *Suppose that the step size h satisfies $h < N^{-\gamma}$. Then*

$$\|P_{E,h} - \mathcal{P}_h\|_{2 \to 0}^{\nabla^N} = O(N^{2\gamma} h^2).$$

Proof. For Euler's method with initial condition x_0

(5.3) $$P_{E,h}f(x_0) = f(x_0) + h\mathcal{B}_{x_0}^N f(x_0) + \frac{h^2}{2}(\mathcal{B}_{x_0}^N)^2 f(x_0) + O(N^{3\gamma}\|f\|_3^{\nabla^N} h^3),$$

where, noting that $\nabla^N \lambda(x_0) = 0$ and using the product rule in Lemma 5.1, we have

$$\mathcal{B}_{x_0}^N f = N^\gamma \lambda(x_0) \cdot \nabla^N f,$$
$$(\mathcal{B}_{x_0}^N)^2 f = N^\gamma \lambda(x_0) \cdot \nabla^N (N^\gamma \lambda(x_0) \cdot \nabla^N f)$$
$$\text{(5.4)} \qquad\qquad = N^{2\gamma} \lambda(x_0)^T [(\nabla^N)^2 f] \lambda(x_0).$$

On the other hand, for the exact process (1.4),

$$\text{(5.5)} \qquad \mathcal{P}_h f(x_0) = f(x_0) + h\mathcal{A}^N f(x_0) + \frac{h^2}{2}(\mathcal{A}^N)^2 f(x_0) + O(N^{3\gamma} \|f\|_3^{\nabla^N} h^3),$$

where, again,

$$\mathcal{A}^N f = N^\gamma \lambda \cdot \nabla^N f.$$

Noting that

(5.6)
$$(\mathcal{A}^N)^2 f(x) = N^{2\gamma} (\lambda \cdot \nabla^N (\lambda \cdot \nabla^N f(x)))$$
$$= N^{2\gamma} \lambda^T ([D^N \lambda]^T \nabla^N f(x) + [(\nabla^N)^2 f] \lambda(x)) + N^{2\gamma} \lambda^T (\text{diag}(N^{-c})[D^N \lambda \times (\nabla)^2] 1_R f)$$

and defining

$$a(x) \stackrel{\text{def}}{=} N^{2\gamma} \lambda^T [D^N \lambda]^T \nabla^N f(x),$$
$$b(x) \stackrel{\text{def}}{=} N^{2\gamma} \lambda^T [(\nabla^N)^2 f] \lambda(x),$$
$$c(x) \stackrel{\text{def}}{=} N^{2\gamma} \lambda^T [\text{diag}(N^{-c})[D^N \lambda \times (\nabla^N)^2] 1_R f(x)],$$

we can write

$$\mathcal{P}_h f(x_0) = f(x_0) + h\mathcal{A}^N f(x_0) + \frac{h^2}{2}(a(x_0) + b(x_0) + c(x_0)) + O(N^{3\gamma} \|f\|_3^{\nabla^N} h^3).$$

Note that $\mathcal{B}_{x_0}^N f(x_0) = \mathcal{A}^N f(x_0)$ and $b(x_0) = (\mathcal{B}_{x_0}^N)^2 f(x_0)$. We may then compare (5.3) and (5.5):

$$(P_{E,h} - \mathcal{P}_h) f(x_0) = \frac{h^2}{2}((\mathcal{B}_{x_0}^N)^2 f(x_0) - (a(x_0) + b(x_0) + c(x_0))) + O(N^{3\gamma} \|f\|_3^{\nabla^N} h^3)$$
$$= \frac{h^2}{2}(-a(x_0) - c(x_0)) + O(N^{3\gamma} \|f\|_3^{\nabla^N} h^3).$$

The term $a(x) + c(x) = O(N^{2\gamma} \|f\|_2^{\nabla^N})$ is clearly nonzero in general, giving the desired result. $\square$

5.3. Approximate midpoint method. Throughout subsection 5.3, we let Z_M^N be the midpoint method approximation to X^N, and we let

$$P_{M,h} f(x) \stackrel{\text{def}}{=} \mathbb{E}_x f(Z_M^N(h)),$$

where h is the step size taken in the algorithm. As before, we will assume $h < N^{-\gamma}$, which is a natural stability condition and is discussed further in section 6.2.

THEOREM 5.5. *Suppose that the step size h satisfies $h < N^{-\gamma}$. Then*

$$\|(P_{M,h} - \mathcal{P}_h)\|_{3\to0}^{\nabla^N} = O(N^{3\gamma} h^3 + N^{2\gamma - \min\{m_k\}} h^2).$$

Remark 4. Theorem 5.5 predicts that the midpoint method behaves locally like a third order method and globally like a second order method if h is in a regime satisfying $N^\gamma h \gg N^{-\min\{m_k\}}$, or equivalently if $h \gg N^{-\gamma-\min\{m_k\}}$. This agrees with the result found in [3] pertaining to the midpoint method, which had $\gamma = 0$, $m_k \equiv 1$, and the running assumption that $h \gg 1/N$. This behavior is demonstrated via numerical example in section 7.

Proof of Theorem 5.5. Let ζ^N denote the matrix with kth column ζ_k^N, i.e.,

$$[\zeta^N] = [\zeta_1^N, \zeta_2^N, \ldots, \zeta_R^N].$$

Recall that ρ is defined via

$$\rho(z) = z + \frac{h}{2} N^\gamma \sum_k \lambda_k(z) N^{c_k} \zeta_k^N.$$

After some algebra, we have

$$\mathcal{B}^N_{\rho(x_0)} f(x) = N^\gamma \left(\lambda \left(x_0 + \frac{h}{2} N^\gamma \sum_k \lambda_k(x_0) N^{c_k} \zeta_k^N \right) \right) \cdot \nabla^N f(x)$$

$$= N^\gamma \lambda(x_0) \cdot \nabla^N f(x) + w(x_0) + O(N^{2\gamma} \|f\|_1^{\nabla^N} h^2),$$

where

$$w(x) \stackrel{\text{def}}{=} N^{2\gamma} \frac{h}{2} [D\lambda(x_0)][\zeta^N] \text{diag}(N^c) \lambda(x_0) \cdot \nabla^N f(x).$$

Next, using the product rule (5.2), we see that

$$(\mathcal{B}^N_{\rho(x_0)})^2 f(x)$$

$$= N^\gamma \lambda \left(x_0 + \frac{h}{2} [\zeta^N] \text{diag}(N^c) \lambda(x_0) \right)$$

$$\cdot \nabla^N \left(N^\gamma \lambda \left(x_0 + \frac{h}{2} [\zeta^N] \text{diag}(N^c) \lambda(x_0) \right) \cdot \nabla^N f \right)(x)$$

$$= N^{2\gamma} \lambda \left(x_0 + \frac{h}{2} [\zeta^N] \text{diag}(N^c) \lambda(x_0) \right)^T$$

$$\cdot [(\nabla^N)^2 f] \lambda \left(x_0 + \frac{h}{2} [\zeta^N] \text{diag}(N^c) \lambda(x_0) \right) \cdot \nabla^N f(x)$$

$$= g(x_0) + O(N^{2\gamma} \|f\|_2^{\nabla^N} h),$$

where

$$g(x_0) \stackrel{\text{def}}{=} N^{2\gamma} \lambda(x_0)^T [(\nabla^N)^2 f(x)] \lambda(x_0).$$

Therefore, since $N^\gamma \lambda(x_0) \cdot \nabla^N f(x_0) = \mathcal{A}^N f(x_0)$, it follows that

$$P_{M,h} f(x_0) = f(x_0) + h \mathcal{B}^N_{\rho(x_0)} f(x_0) + \frac{h^2}{2} (\mathcal{B}^N_{\rho(x_0)})^2 f(x_0) + O(N^{3\gamma} \|f\|_3^{\nabla^N} h^3)$$

$$= f(x_0) + h \left(\mathcal{A}^N f(x_0) + w(x_0) + O(N^{2\gamma} \|f\|_2^{\nabla^N} h^2) \right)$$

$$+ \frac{h^2}{2} \left(g(x_0) + O(N^{2\gamma} \|f\|_2^{\nabla^N} h) \right) + O(N^{3\gamma} \|f\|_3^{\nabla^N} h^3).$$

Recall that

$$(\mathcal{A}^N)^2 f(x) = a(x) + b(x) + c(x),$$

where

$$a(x) = N^{2\gamma} \lambda^T [D^N \lambda]^T \nabla^N f(x),$$
$$b(x) = N^{2\gamma} \lambda^T [(\nabla^N)^2 f] \lambda(x),$$
(5.7) $$c(x) = N^{2\gamma} \lambda^T [\mathrm{diag}(N^{-c})[D^N \lambda \times (\nabla^N)^2] 1_R f(x)],$$

and

$$\mathcal{P}_h f(x_0) = f(x_0) + h \mathcal{A}^N f(x_0) + \frac{h^2}{2}(a(x_0) + b(x_0) + c(x_0)) + O(N^{3\gamma} \|f\|_3^{\nabla^N} h^3).$$

Noting that $b(x_0) = g(x_0)$, we see that

$$(P_{M,h} - \mathcal{P}_h) f(x_0)$$
$$= h\ w(x_0) + \frac{h^2}{2}(g(x_0) - (a(x_0) + b(x_0) + c(x_0))) + O(N^{3\gamma} \|f\|_3^{\nabla^N} h^3)$$
(5.8) $$= \left(h\ w(x_0) - \frac{h^2}{2} a(x_0) \right) - \frac{h^2}{2} c(x_0) + O(N^{3\gamma} \|f\|_3^{\nabla^N} h^3).$$

We will now gain control over the terms $(h\ w(x_0) - \frac{h^2}{2} a(x_0))$ and $\frac{h^2}{2} c(x_0)$ separately. Handling $\frac{h^2}{2} c(x_0)$ first, we have that $\nabla^N \lambda_k \in O(N^{c_k - m_k})$, and so

$$c(x_0) = O(N^{2\gamma - \min\{m_k\}} \|f\|_2^{\nabla^N}).$$

Next, we will show that

$$h\ w(x_0) - \frac{h^2}{2} a(x_0) = O(N^{2\gamma - \min\{m_k\}} \|f\|_1^{\nabla^N} h^2).$$

We have

$$h\ w(x_0) - \frac{h^2}{2} a(x_0)$$
(5.9) $$= \frac{h^2}{2} N^{2\gamma} [D\lambda(x_0)][\zeta^N] \mathrm{diag}(N^c) \lambda(x_0) \cdot \nabla^N f(x_0) - \frac{h^2}{2} N^{2\gamma} \lambda^T [D^N \lambda]^T \nabla^N f(x)$$
$$= \frac{h^2}{2} N^{2\gamma} \left([D\lambda(x_0)][\zeta^N] \mathrm{diag}(N^c) - [D^N \lambda(x_0)] \right) \lambda(x_0) \cdot \nabla^N f(x_0).$$

By Proposition 5.2, $\nabla^N f(x)$ is bounded by $\|f\|_1^{\nabla^N}$. Therefore, we need only show that the difference between the two square matrices

(5.10) $$[D^N \lambda(x_0)] \qquad \text{and} \qquad [D\lambda(x_0)][\zeta^N] \mathrm{diag}(N^c)$$

is $O(N^{-\min\{m_k\}})$. Recalling the definitions in (5.1), the (i, j)th entry of the left-hand side of (5.10) is

$$N^{c_j} (\lambda_i(x_0 + \zeta_j^N) - \lambda_i(x_0)),$$

whereas that of the right-hand side of (5.10) is

$$N^{c_j} \nabla \lambda_i \cdot \zeta_j^N.$$

Also, note that, for $\lambda \in C_c^2(\mathbb{R}^d, \mathbb{R})$,

$$((\lambda(x+v) - \lambda(x)) - \nabla\lambda(x) \cdot v) \in O(|v|^2 \|\lambda\|_2),$$

where

$$\|\lambda\|_2 = \sup\{\|\lambda\|_\infty, \|\partial_{x_i}\lambda\|_\infty, \|\partial_{x_j}\partial_{x_\ell}\lambda\|_\infty, i, j, k \le d\}.$$

Since $\|\lambda_k\|_2$ is bounded for any k, the difference between the (i,j)th entries of the two expressions in (5.10) is

$$O(N^{c_j} N^{-2m_j}).$$

Also, recall that $c_j - m_j \le 0$. Thus the above is also

$$O(N^{-\min\{m_k\}}).$$

Therefore (5.9) is of order

$$O(N^{2\gamma - \min\{m_k\}} h^2 \|f\|_1^{\nabla^N}),$$

as desired. Combining the above with (5.8) gives us

$$
\begin{aligned}
\|(\mathcal{P}_h - P_{M,h})f\|_0 &= O(N^{2\gamma - \min\{m_k\}} \|f\|_1^{\nabla^N} h^2 \\
&\quad + N^{2\gamma - \min\{m_k\}} \|f\|_2^{\nabla^N} h^2 + N^{3\gamma} \|f\|_3^{\nabla^N} h^3) \\
&= O(\|f\|_3^{\nabla^N} [N^{3\gamma} h^3 + N^{2\gamma - \min\{m_k\}} h^2]),
\end{aligned}
$$

(5.11)

implying

$$\|P_{M,h} - \mathcal{P}_h\|_{3\to 0}^{\nabla^N} = O(N^{3\gamma} h^3 + N^{2\gamma - \min\{m_k\}} h^2),$$

as desired.   □

5.4. Weak trapezoidal method. Throughout subsection 5.4, we let Z_{trap}^N be the weak trapezoidal approximation to X^N, and we let

$$P_{trap,h}f(x) \overset{\text{def}}{=} \mathbb{E}_x f(Z_{trap}^N(h)),$$

where h is the size of the time discretization. We will again consider only the case $h < N^{-\gamma}$, which is a natural stability condition and is discussed further in section 6.2.

We make the standing assumption that for all x in our state space of interest, and $k, j \in \{1, \dots, R\}$, we have that

(5.12) $$\xi_1 \lambda_k(x + \zeta_j^N) - \xi_2 \lambda_k(x) \ge 0,$$

where $\xi_1 > \xi_2$ are defined in (3.9) for some $\theta \in (0, 1)$. Noting that ζ_j^N will often be small, and that $\xi_1 - \xi_2 = 1$, for most processes, including those arising from biochemistry, the requirement (5.12) holds as long as the process is not directly at

the boundary of the positive orthant. Weakening (5.12) is almost certainly doable, for example, by gaining control over the probability that a process leaves a region in which the condition holds. This is an avenue for future work.

THEOREM 5.6. *Suppose that the step size h satisfies $h < N^{-\gamma}$. Then*

$$\|(P_{trap,h} - \mathcal{P}_h)\|_{3\to0}^{\nabla^N} = O(N^{3\gamma}h^3).$$

Proof. Consider one step of the method with a step size of size h and with initial value x_0. Note that the first step of the algorithm produces a value y^* that is distributionally equivalent to one produced by a Markov process with generator B_1^N given by

$$B_1^N f(x) = N^\gamma \lambda(x_0) \cdot \nabla^N f(x).$$

Next, given both x_0 and y^*, the second step produces a value which is distributionally equivalent to one produced by a Markov process with generator

(5.13) $$B_2^N f(x) = N^\gamma [\xi_1 \lambda(y^*) - \xi_2 \lambda(x_0)]^+ \cdot \nabla^N f(x).$$

Recall that, for the exact process,

$$\mathcal{P}_h f(x_0) = f(x_0) + h\mathcal{A}^N f(x_0) + \frac{h^2}{2}(\mathcal{A}^N)^2 f(x_0) + O(N^{3\gamma}\|f\|_3^{\nabla^N}h^3).$$

For the approximate process we have that

(5.14) $$\begin{aligned} P_{trap,h} f(x_0) &= \mathbb{E}_{x_0}[\mathbb{E}_{x_0}[f(Z_{trap}^N(h))|y^*]] \\ &= \mathbb{E}_{x_0} f(y^*) + (1-\theta)h\mathbb{E}_{x_0}[B_2^N f(y^*)] \\ &\quad + \frac{(1-\theta)^2 h^2}{2}\mathbb{E}_{x_0}[(B_2^N)^2 f(y^*)] + O(N^{3\gamma}\|f\|_3^{\nabla^N}h^3). \end{aligned}$$

We will expand each piece of (5.14) in turn. Noting that $B_1^N f(x_0) = \mathcal{A}^N f(x_0)$, the first term is

$$\begin{aligned} \mathbb{E}_{x_0} f(y^*) &= f(x_0) + \mathbb{E}_{x_0}\left[\int_0^{\theta h} B_1^N f(Z_s)ds\right] \\ &= f(x_0) + \theta h\mathcal{A}^N f(x_0) + \frac{\theta^2 h^2}{2}(B_1^N)^2 f(x_0) + O(N^{3\gamma}\|f\|_3^{\nabla^N}h^3). \end{aligned}$$

We turn our attention to the second term, $(1-\theta)h\mathbb{E}_{x_0}[B_2^N f(y^*)]$, and begin by making the following definition:

$$g(y^*) \stackrel{\text{def}}{=} B_2^N f(y^*) = N^\gamma [\xi_1 \lambda(y^*) - \xi_2 \lambda(x_0)]^+ \cdot \nabla^N f(y^*)$$

so that $g(x) = N^\gamma([\xi_1 \lambda(x) - \xi_2 \lambda(x_0)]^+ \cdot \nabla^N)f(x)$. Because $\xi_1 - \xi_2 = 1$, we have

$$g(x_0) = N^\gamma \lambda(x_0) \cdot \nabla^N f(x_0) = \mathcal{A}^N f(x_0).$$

By our standing assumption (5.12) we have that

$$g(x_0+\zeta_k) - g(x_0) = N^\gamma(\xi_1 \lambda(x_0+\zeta_k) - \xi_2 \lambda(x_0)) \cdot \nabla^N f(x_0+\zeta_k) - N^\gamma \lambda(x_0) \cdot \nabla^N f(x_0).$$

After some algebra,

$$B_1^N g(x_0) = N^\gamma (\lambda(x_0) \cdot \nabla^N g)(x_0) = N^\gamma \sum_k N^{c_k} \lambda_k(x_0)[g(x_0 + \zeta_k) - g(x_0)]$$
$$= \xi_1 N^\gamma \lambda(x_0) \cdot \nabla^N (N^\gamma \lambda \cdot f)(x_0) - \xi_2 N^\gamma \lambda(x_0) \cdot \nabla^N (\lambda(x_0) \cdot f)(x_0)$$
$$= \xi_1 (B_1^N \mathcal{A}^N f(x_0)) - \xi_2 ((B_1^N)^2 f)(x_0).$$

Thus,

$$\mathbb{E}_{x_0}[B_2^N f(y^*)] = \mathbb{E}_{x_0}[g(y^*)] = g(x_0) + \theta h B_1^N g(x_0) + O(N^{3\gamma} \|f\|_3^{\nabla^N} h^2)$$
$$= \mathcal{A}^N f(x_0) + \theta h \left[\xi_1 (B_1^N \mathcal{A}^N f)(x_0) - \xi_2 (B_1^N)^2 f(x_0) \right] + O(N^{3\gamma} \|f\|_2^{\nabla^N} h^2)$$
$$= \mathcal{A}^N f(x_0) + \theta h \left[\xi_1 (\mathcal{A}^N)^2 f(x_0) - \xi_2 (B_1^N)^2 f(x_0) \right] + O(N^{3\gamma} \|f\|_3^{\nabla^N} h^2),$$

where the last line follows since $B_1^N f(x_0) = \mathcal{A}^N f(x_0)$ for any f.

Finally, we turn the the last term in (5.14). Define

$$q(y^*) \stackrel{\text{def}}{=} (B_2^N)^2 f(y^*)$$
$$= [\xi_1 \lambda(y^*) - \xi_2 \lambda(x_0)]^+ \cdot \nabla^N ([\xi_1 \lambda(y^*) - \xi_2 \lambda(x_0)]^+ \nabla^N f)(y^*)$$

so that

$$q(x) = [\xi_1 \lambda - \xi_2 \lambda(x_0)]^+ \cdot \nabla^N ([\xi_1 \lambda - \xi_2 \lambda(x_0)]^+ \nabla^N f)(x).$$

By our standing assumption (5.12) we have that

$$\mathbb{E}_{x_0}[(B_2^N)^2 f(y^*)] = \mathbb{E}_{x_0}[q(y^*)]$$
$$(5.15) \qquad = q(x_0) + O(N^{3\gamma} \|f\|_3^{\nabla^N} h)$$
$$= (B_1^N)^2 f(x_0) + O(N^{3\gamma} \|f\|_3^{\nabla^N} h).$$

Noting that

$$(1 - \theta)\theta \xi_1 = \frac{1}{2} \qquad \text{and} \qquad (1 - \theta)\theta \xi_2 = \frac{(1 - \theta)^2 + \theta^2}{2},$$

we may conclude the following from the above calculations:

$$\mathbb{E}_{x_0}[f(Z_{trap,h}^N)] = \mathbb{E}_{x_0} f(y^*) + (1 - \theta) h \mathbb{E}_{x_0}[B_2^N f(y^*)] + \frac{(1 - \theta)^2 h^2}{2} \mathbb{E}_{x_0}[(B_2^N)^2 f(y^*)]$$
$$+ O(N^{3\gamma} \|f\|_3^{\nabla^N} h^3)$$
$$= f(x_0) + \theta h \mathcal{A}^N f(x_0) + \frac{\theta^2 h^2}{2} (B_1^N)^2 f(x_0)$$
$$+ (1 - \theta) h \mathcal{A}^N f(x_0) + \frac{h^2}{2} (\mathcal{A}^N)^2 f(x_0) - \frac{h^2}{2} [(1 - \theta)^2 + \theta^2] (B_1^N)^2 f(x_0)$$
$$+ \frac{(1 - \theta)^2 h^2}{2} (B_1^N)^2 f(x_0) + O(N^{3\gamma} \|f\|_3^{\nabla^N} h^3)$$
$$= f(x_0) + \mathcal{A}^N f(x_0) + \frac{h^2}{2} (\mathcal{A}^N)^2 f(x_0) + O(N^{3\gamma} \|f\|_3^{\nabla^N} h^3).$$

Thus

$$\|(P_{trap,h} - \mathcal{P}_h) f\|_0 \in O(N^{3\gamma} \|f\|_3^{\nabla^N} h^3),$$

and the proof is complete. $\square$

6. Global bounds and stability. In section 6.1, we bound $\|\mathcal{P}_t f\|_n^{\nabla^N}$, which was the remaining piece to handle in Theorem 4.1 to give us global bounds on the weak error induced by the different methods. In section 6.2, we briefly discuss some issues related to stability of the different methods.

6.1. Bounds on $\|\mathcal{P}_t f\|_n^{\nabla^N}$. In this section we bound $\|\mathcal{P}_t f\|_n^{\nabla^N}$, where n is a nonnegative integer and $\mathcal{P}_t$ is the semigroup operator (4.1) of the scaled process (1.4). We point out, however, that for any process X^N for which $\mathcal{P}_t$ is well behaved, in that $\|\mathcal{P}_t\|_{n\to 0}^{\nabla^N}$ is bounded uniformly in N, the following results are not needed, and, in fact, would most likely be a *least optimal* bound, as the bound grows exponentially in $N^\gamma t$. Note that any system satisfying the classical scaling has $\gamma = 0$. We also point out that the arguments used below are quite similar to those used in [22] by Hu, Li, and Min, which were extensions of those used in [3] by Anderson, Ganguly, and Kurtz.

For $t \geq 0$ and any $x \in \mathbb{R}^d$ we define

$$v(t,x) \stackrel{\text{def}}{=} \mathcal{P}_t f(x) = \mathbb{E}_x f(X_t^N).$$

THEOREM 6.1. *If $\|f\|_n^{\nabla^N} < \infty$, then*

$$\|v(t,\cdot)\|_n^{\nabla^N} = \|\mathcal{P}_t f\|_n^{\nabla^N} \leq \|f\|_n^{\nabla^N} e^{N^\gamma C_n t},$$

where

(6.1) $$C_n = 2\left(\|\lambda\|_1^{\nabla^N} n\ R + R(n-1)\|\lambda\|_n^{\nabla^N}\right).$$

We delay the proof of Theorem 6.1 until the following lemma is shown, the proof of which is similar to that found in [22], which itself was an extension of the proof of Lemma 4.3 in [3].

LEMMA 6.1. *Given a multiset I of $\{1,\ldots,R\}$, there exists a function $q_I(x)$ that is a linear function of terms of the form $\nabla_J^N v(t,x)$ with $|J| < |I|$, so that*

$$\partial_t \nabla_I^N v(t,x) = N^\gamma (\lambda \cdot \nabla^N)\nabla_I^N v(t,x) + N^\gamma \sum_{i=1}^{|I|} (\beta_i \cdot \nabla^N)\nabla_{I\setminus \ell_i} v(t, x + \zeta_{\ell_i}) + N^\gamma q_I(x),$$

where $\beta_i = \nabla_{\ell_i}^N \lambda$. Further, q_I consists of at most $R(|I|-1)$ terms of the form $\nabla_J^N v(t,x)$, each of whose coefficients are bounded above by $\|\lambda\|_{|I|}^{\nabla^N}$.

Proof. This goes by induction. For $|I| = 0$, the statement follows because

(6.2) $$\partial_t v(t,x) = N^\gamma (\lambda \cdot \nabla^N)v(t,x).$$

Note that in this case there are no β_i or q terms. It is instructive to perform the $|I| = 1$ case. We have

$$\begin{aligned}
\partial_t \nabla_k^N v(t,x) &= \nabla_k^N \partial_t v(t,x) \\
&= \nabla_k^N (N^\gamma \lambda \cdot \nabla^N v(t,x)) \\
&= N^\gamma(\nabla_k^N \lambda \cdot \nabla^N)v(t,x) + N^\gamma \lambda \cdot \nabla_k^N \nabla^N v(t,x) + N^\gamma(N^{-c_k}\nabla_k^N \lambda \cdot \nabla_k^N \nabla^N v(t,x)).
\end{aligned}$$

Note that for any $g: \mathbb{R}^d \to \mathbb{R}$

(6.3) $$(\nabla_k^N \lambda \cdot \nabla^N)g(x) + (N^{-c_k}\nabla_k^N \lambda \cdot \nabla^N)\nabla_k^N g(x) = (\nabla_k^N \lambda \cdot \nabla^N)g(x + \zeta_k).$$

Therefore, with $g(x) = v(t, x)$ in the above, we have that

$$\partial_t \nabla_k^N v(t, x) = N^\gamma (\lambda(x) \cdot \nabla^N) \nabla_k^N v(t, x) + N^\gamma (\nabla_k^N \lambda(x) \cdot \nabla^N) v(t, x + \zeta_k).$$

Now assume that it holds for a set of size $\leq |I|$. Then using the inductive hypothesis, Lemma 5.1, and (6.3) yields

$$
\begin{aligned}
\partial_t \, & \nabla_k^N \nabla_I^N v(t, x) \\
&= \nabla_k^N \partial_t \nabla_I^N v(t, x) \\
&= N^\gamma \nabla_k^N \left[(\lambda \cdot \nabla^N) \nabla_I^N v(t, x) + \sum_{i=1}^{|I|} (\beta_i \cdot \nabla^N) \nabla_{I \backslash \ell_i} v(t, x + \zeta_{\ell_i}) + q_I(x) \right] \\
&= N^\gamma \left[(\lambda \cdot \nabla^N) \nabla_{I \cup k}^N v(t, x) + (\nabla_k^N \lambda \cdot \nabla^N) \nabla_I^N v(t, x + \zeta_k) \right] \\
&\quad + N^\gamma \sum_{i=1}^{|I|} \left[(\beta_i \cdot \nabla^N) \nabla_k^N \nabla_{I \backslash \ell_i}^N v(t, x + \zeta_{\ell_i}) + (\nabla_k^N \beta_i \cdot \nabla^N) \nabla_{I \backslash \ell_i}^N v(t, x + \zeta_{\ell_i} + \zeta_k) \right] \\
&\quad + N^\gamma \nabla_k^N q_I(x) \\
&= N^\gamma (\lambda \cdot \nabla^N) \nabla_{I \cup k}^N v(t, x) \\
&\quad + N^\gamma \left[(\nabla_k^N \lambda \cdot \nabla^N) \nabla_{I \cup k \backslash k}^N v(t, x + \zeta_k) + \sum_{i=1}^{|I|} (\beta_i \cdot \nabla^N) \nabla_{I \cup k \backslash \ell_i}^N v(t, x + \zeta_{\ell_i}) \right] \\
&\quad + N^\gamma \left[\nabla_k^N q_I(x) + (\nabla_k^N \beta_i \cdot \nabla^N) \nabla_{I \backslash \ell_i}^N v(t, x + \zeta_{\ell_i} + \zeta_k) \right],
\end{aligned}
$$

showing the result. □

Proof of Theorem 6.1. Let $n \geq 0$. Define

$$U_n(t) \stackrel{\text{def}}{=} \max_{x, |I| \leq n} |\nabla_I^N v(t, x)| = \|v\|_n^{\nabla^N}.$$

Each $\nabla_I^N v(t, x)$ is a continuously differentiable function with respect to t. Therefore, the maximum above is achieved at some (I^*, x^*) for all $t \in [0, t_1]$, where $t_1 > 0$. Fixing this choice of (I^*, x^*), we have

$$U_n(t) = \nabla_{I^*}^N v(t, x^*)$$

for all $t < t_1$.

Note that

$$
\begin{aligned}
[(\lambda \cdot \nabla^N) \nabla_{I^*}^N v(t, x^*)] \nabla_{I^*}^N v(t, x^*) &= \sum_k \lambda_k(x) (\nabla_k^N \nabla_{I^*}^N v(t, x^*)) \nabla_{I^*}^N v(t, x^*) \\
&= \sum_k N^{c_k} \lambda_k(x) (\nabla_{I^*}^N v(t, x^* + \zeta_k) - \nabla_{I^*}^N v(t, x^*)) \nabla_{I^*}^N v(t, x^*) \\
&\leq 0,
\end{aligned}
$$

(6.4)

where the final inequality holds by the specific choices of I^* and x^*. Also note that for any $\ell_i \in I^*$ and any choice of x

$$(6.5) \qquad |\nabla^N \nabla_{I^* \backslash \ell_i}^N v(t, x)| \leq \sum_{k=1}^R |\nabla_k \nabla_{I^* \backslash \ell_i}^N v(t, x)| \leq R |\nabla_{I^*}^N v(t, x^*)|.$$

From Lemma 6.1 and (6.4) and (6.5), we have that

(6.6)
$$\frac{1}{2}\partial_t(\nabla_{I^*}^N v(t,x^*))^2 = (\partial_t \nabla_{I^*}^N v(t,x^*))\nabla_{I^*}^N v(t,x^*)$$

$$= N^\gamma \left[(\lambda \cdot \nabla^N)\nabla_{I^*}^N v(t,x^*) + \sum_{i=1}^{|I^*|} (\beta_i \cdot \nabla^N)\nabla_{I^*\setminus \ell_i} v(t,x^* + \zeta_{\ell_i}) + q_{I^*}(x^*) \right] \nabla_{I^*}^N v(t,x^*)$$

$$\leq N^\gamma \left[\|\lambda\|_1^{\nabla^N} |I^*| \ R \ |\nabla_{I^*}^N v(t,x^*)|^2 + R(|I^*|-1)\|\lambda\|_{|I^*|}^{\nabla^N} |\nabla_{I^*}^N v(t,x^*)|^2 \right],$$

where we have used the fact that each $\beta_i = \nabla_{\ell_i}\lambda$ for $\ell_i \in I^*$. Setting

(6.7)
$$C_n = 2\left(\|\lambda\|_1^{\nabla^N} n \ R + R(n-1)\|\lambda\|_n^{\nabla^N} \right),$$

we see by an application of Gronwall's inequality that the conclusion of the theorem holds for all $t < t_1$. That is, for $t < t_1$

$$U_n(t) \leq \|f\|_n^{\nabla^N} e^{N^\gamma C_n t}.$$

To continue, repeat the above argument on the interval $[t_1, t_2)$, with I^*, x^* again chosen to maximize U_n on that interval, and note that

$$U_n(t_1) \leq \|f\|_n^{\nabla^N} e^{N^\gamma C_n t_1},$$

so that we may conclude that for $t_1 \leq t < t_2$

$$U_n(t) \leq \|f\|_n^{\nabla^N} e^{N^\gamma C_n t_1} e^{N^\gamma C_n(t-t_1)} = \|f\|_n^{\nabla^N} e^{N^\gamma C_n t}.$$

Continuing on, we see that $t_i \to \infty$ as $i \to \infty$ by the boundedness of the time derivatives of $v(t,x)$, thereby concluding the proof. $\quad\square$

Remark 5. In the theorem above, $C_n \in \|\lambda\|_n^{\nabla^N}$.

Combining all of the previous results, we have the following theorems.

THEOREM 6.2 (global bound for the Euler method). *Suppose that the step size h satisfies $h < N^{-\gamma}$, and $T = nh$. Then*

$$\|(P_{E,h}^n - \mathcal{P}_{nh})\|_{2\to 0}^{\nabla^N} = O(N^{2\gamma}h e^{C_2 N^\gamma T}),$$

where $C_2 \in O(\|\lambda\|_2^{\nabla^N})$ is defined in (6.1).

THEOREM 6.3 (global bound for the midpoint method). *Suppose that the step size h satisfies $h < N^{-\gamma}$, and $T = nh$. Then*

$$\|(P_{M,h}^n - \mathcal{P}_{nh})\|_{3\to 0}^{\nabla^N} = O([N^{3\gamma}h^2 + N^{2\gamma-\min\{m_k\}}h]e^{C_3 N^\gamma T}),$$

where $C_3 \in O(\|\lambda\|_3^{\nabla^N})$ is defined in (6.1).

The following immediate corollary to the theorem above recovers the result in [3].

COROLLARY 6.4. *Under the additional condition $h > N^{-\gamma-\min\{m_k\}}$ in Theorem 6.3, the leading order of the error of the midpoint method is $O(h^2)$.*

THEOREM 6.5 (global bound for the weak trapezoidal method). *Suppose that the step size h satisfies $h < N^{-\gamma}$, and $T = nh$. Then*

$$\|P_{trap,h}^n - \mathcal{P}_{nh}\|_{3\to 0}^{\nabla^N} = O(h^2 N^{3\gamma} e^{N^\gamma C_3 T}),$$

where $C_3 \in O(\|\lambda\|_3^{\nabla^N})$ is defined in (6.1).

Thus, we see that the weak trapezoidal method detailed in Algorithm 3 is the only method that boasts a global error of second order in the step size h in an "honest sense." That is, it is a second order method regardless of the relation of h with respect to N. This is in contrast to the midpoint method which has second order accuracy only when the order of h is larger than $N^{-\gamma - \min\{m_k\}}$.

6.2. Stability concerns. The main results and proofs of our paper have incorporated stability concerns into the analysis. This is seen in the statements of the theorems by the running condition that $h < N^{-\gamma}$, where we recall that N^γ should be interpreted as the time scale of the system. Without this condition, the methods are unstable. It is an interesting question, and the subject of future work, of how to determine the stability properties of other methods in this setting.

As an instructive example, again consider the system

$$S_1 \underset{100}{\overset{100}{\rightleftarrows}} S_2,$$
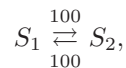
with $X_1(0) = X_2(0) = 10,000$. In this case, it is natural to take $N = 10,000$. As the rate constants are $100 = \sqrt{10,000}$, we take $\beta_1 = \beta_2 = 1/2$ and find that $\gamma = 1/2$. The equation governing the normalized process X_1^N is

$$X_1^N(t) = X_1^N(0) - Y_1\left(N^{1/2}N\int_0^t X_1^N(s)ds\right)\frac{1}{N} + Y_2\left(N^{1/2}N\int_0^t (2 - X_1^N(s))ds\right)\frac{1}{N},$$

where we have used that $X_1^N + X_2^N \equiv 2$. It is now clear that if the condition $h < N^{-\gamma}$ is violated, a path generated by any of the explicit methods discussed in this paper will behave quite poorly.

7. Examples. We provide two test systems. The first is a simple linear system with three species that we will use to demonstrate our main analytical results. The second is a gene-protien-mRNA model that we will use to demonstrate the capabilities of the different methods on an actual test problem. We note that in all simulations of the weak trapezoidal algorithm, we chose $\theta = 1/2$.

Example 1. Consider the following first order reaction network:

$$A \underset{\kappa_2}{\overset{\kappa_1}{\rightleftarrows}} B \underset{\kappa_4}{\overset{\kappa_3}{\rightleftarrows}} C,$$
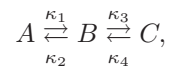
with $\kappa_1 = 0.03$, $\kappa_2 = 1$, $\kappa_3 = 0.1$, and $\kappa_4 = 1$. We start from the initial state

$$X(0) = (X_A(0), X_B(0), X_C(0)) = (13000, 100, 20),$$

where we make the obvious associations $X_1 = X_A$, $X_2 = X_B$, and $X_3 = X_C$. We approximate $X(2)$ using the three methods considered in this paper: Euler, midpoint, and weak trapezoidal with a choice of $\theta = 1/2$. For first order systems, we may find the first moments and the covariances of $X(t)$ as solutions of linear ODEs using a moment generating function approach [15].

In Figure 7.1, we show a log-log plot of $|\mathbb{E}[X_3^2(2)] - \mathbb{E}[Z_3^2(2)]|$ against h for the three approximation methods. Each data point was found from either 10^6, 2.9×10^6, 3.9×10^6, 4.9×10^6, 8×10^6, or 10^7 independent simulations, with the number of simulations depending upon the size of h and the method being used. The slope for
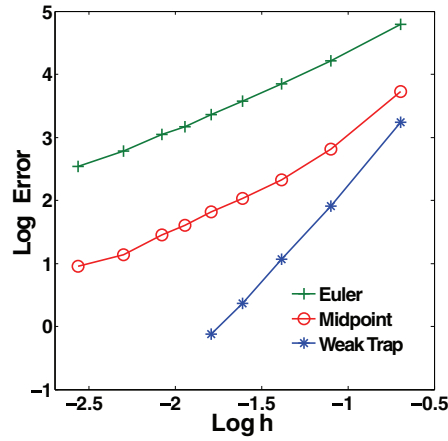
FIG. 7.1. *The log-log plot of $|\mathbb{E}[X_3^2(2)] - \mathbb{E}[Z_3^2(2)]|$ against h for the three approximation methods. The slope for Euler's method is 1.21, whereas the slope for the weak trapezoidal solution with $\theta = 1/2$ is 3.06, which is better than expected. The curve governing the solution from the midpoint method appears not to be linear, a behavior predicted by Theorem 5.5.*

FIG. 7.2. *The log-log plot of $|\mathbb{E}[X_3^2(2)] - \mathbb{E}[Z_3^2(2)]|$ against h. The slope generated via midpoint tau-leaping shifts from 2.03 in (a) to 1.12 in (b).*

Euler's method is 1.21, whereas the slope for the weak trapezoidal solution is 3.06, which is better than expected. The curve governing the solution from the midpoint method appears not to be linear, a behavior predicted by Theorem 5.5.
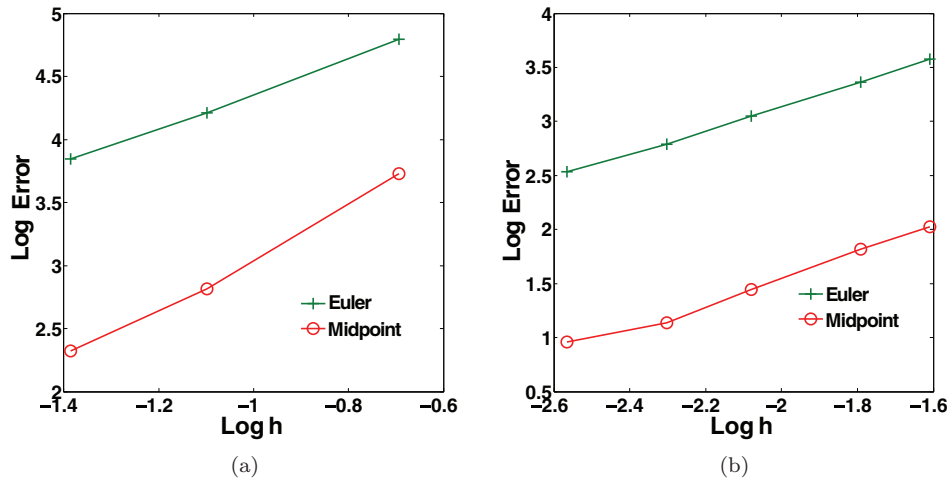
In Figure 7.2 we again consider the log-log plots of $|\mathbb{E}[X_3^2(2)] - \mathbb{E}[Z_3^2(2)]|$ against h, but now only for Euler's method and the midpoint method so that we may see the change in behavior in the midpoint method predicted in Theorem 5.5. In (a), we see that for larger h the slope generated via the midpoint method is 2.03, whereas in (b) the slope is 1.12 when h is smaller. For reference, in (a) the slope generated by Euler's method is 1.366, whereas in (b) it is 1.09.
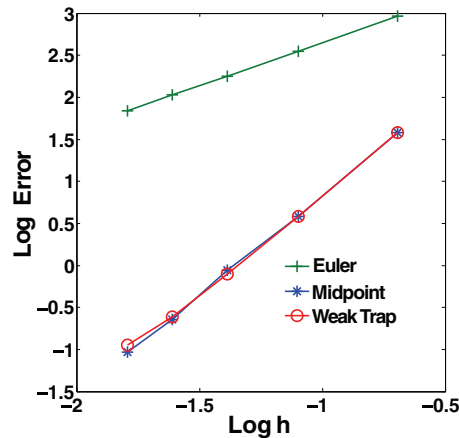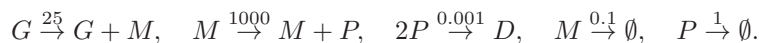
FIG. 7.3. *Log-log plot of* $|\mathbb{E}[X_3(2)] - \mathbb{E}[Z_3(2)]|$ *against h for the three approximation methods. The approximate slopes are* 1.02 *for Euler's method,* 2.372 *for the midpoint method, and* 2.3 *for the trapezoidal method.*

While the simulations make no use of the scalings inherent in the system, it is instructive for us to quantify them in this example so that we are able to understand the behavior of the midpoint method. We have $N \approx 10^4$, $\alpha_1 = 1$, $\alpha_2 = 1/2$, $\alpha_3 = 1/4$, and $m_k = 1/4$. Also, $\gamma \approx 0$. Therefore, Theorem 5.5 predicts that the midpoint method will behave as an order two method if $h \gg N^{-1/4} \approx 1/10$, or if $\log(h) \gg -2.3$, which roughly agrees with what is shown in Figures 7.1 and 7.2. Note that Theorem 5.5 will never provide a sharp estimate as to when the behavior will change, as it is a local result and the scalings in the system will change during the course of a simulation.

The fact that the trapezoidal method gave an order three convergence rate above does not hold in general. This was demonstrated in the proof of Theorem 5.6, but it is helpful to also show this via example. In Figure 7.3 we present a log-log plot of $|\mathbb{E}X_2(2) - \mathbb{E}Z_2(2)|$ for the different algorithms on this same example. The approximate slopes are 1.02 for Euler's method, 2.372 for the midpoint method, and 2.3 for the trapezoidal method. We point out that all of the plots above represent results pertaining to the *nonnormalized processes*, as the simulation methods themselves make no use of the scalings.

Example 2. Consider a model of gene transcription and translation:

$$G \overset{25}{\to} G + M, \quad M \overset{1000}{\to} M + P, \quad 2P \overset{0.001}{\to} D, \quad M \overset{0.1}{\to} \emptyset, \quad P \overset{1}{\to} \emptyset.$$

Here a single gene is being translated into mRNA, which is then being transcribed into proteins, and finally the proteins produce stable dimers. The final two reactions represent degradation of mRNA and proteins, respectively. Suppose we start with one gene and no other molecules, and that we want to estimate the expected number of dimers at time $T = 1$ to an accuracy of ± 1.0 with 95% confidence. Therefore, we want the variance of our estimator to be smaller than $(1/1.96)^2 = .2603$.

While $\epsilon = 1$ for the unscaled version of this problem, the simulation of just a few paths of the system will show that there will be somewhere in the magnitude of 3,500 dimers at time $T = 1$. Therefore, for the scaled system, we are asking for an accuracy

TABLE 7.1

Performance of an exact algorithm with crude Monte Carlo estimator (1.2). *Approximation is the* 95% *confidence interval.*

Approximation	# paths	CPU time	# updates
3714.2 ± 1.0	4,740,000	149,000 CPU S	8.27×10^{10}

TABLE 7.2

Performance of Euler's method with crude Monte Carlo. Approximation is the 95% *confidence interval.*

Step size	Approximation	# paths	CPU time	# updates
$h = 3^{-7}$	$3,712.3 \pm 1.0$	4,750,000	13,374.6 CPU s	6.2×10^{10}
$h = 3^{-6}$	$3,707.5 \pm 1.0$	4,750,000	6,207.9 CPU s	2.1×10^{10}
$h = 3^{-5}$	$3,693.4 \pm 1.0$	4,700,000	2,803.9 CPU s	6.9×10^{9}
$h = 3^{-4}$	$3,654.6 \pm 1.0$	4,650,000	1,219 CPU s	2.6×10^{9}

of $\widetilde{\epsilon} = 1/3500 \approx 0.0002857$. Also, a few paths (100 is sufficient) show that the order of magnitude of the variance of the normalized number of dimers is approximately 0.11. Thus, the approximate number of exact sample paths we will need to generate can be found by solving

$$\frac{1}{n}\mathsf{Var}(\text{normalized \# dimers}) = (\widetilde{\epsilon}/1.96)^2 \implies n = 5.18 \times 10^6.$$

Therefore, we will need approximately five million independent sample paths generated via an exact algorithm. Implementing the modified next reaction method [1] on our machine (using MATLAB), each path takes approximately 0.03 CPU s to generate. Therefore, the approximate amount of time to solve this particular problem will be 155,000 CPU s, which is about 43 hours. The outcome of such a simulation is detailed in Table 7.1, where "# updates" refers to the total number, over all paths, of updates to the system performed, and random variables generated, and is used as a quantification for the computational complexity of the different methods under consideration. In terms of software and hardware, the authors used MATLAB for all computations, which were performed on an Apple machine with a 2.2 GHz Intel i7 processor.

Next, we solved the problem using Euler's method, the approximate midpoint method, and the weak trapezoidal method with $\theta = 1/2$. We note that, for each of the three approximations, we used the most naive implementation possible by simply setting the value of any component that goes negative in the course of a step to zero, and by using a fixed step size, $h > 0$. Thus, improvements can be gained on the stated results by using a more sophisticated implementation [2, 8]. However, we did produce our approximate paths in batches of 50,000, which greatly reduces the cost of generating the Poisson random variables with the built-in MATLAB Poisson random number generator.

In Table 7.2 we provide data on the performance of Euler's method with various step sizes, combined with a crude Monte Carlo estimator (1.8). Note that the bias in Euler's method is apparent even for very small h. In Table 7.3 we provide data on the performance of the midpoint method with various step sizes, combined with a crude Monte Carlo estimator (1.8). Note that the solution has a much higher variance when $h = 1/3$, thereby necessitating significantly more paths to get a desired tolerance. This demonstrates the stability concerns discussed in section 6.2. This problem does not

TABLE 7.3

Performance of the midpoint method with crude Monte Carlo. Approximation is the 95% confidence interval.

Step size	Approximation	# paths	CPU time	# updates
$h = 3^{-4}$	$3{,}713.6 \pm 1.0$	4,650,000	1,269.1 CPU s	2.3×10^9
$h = 3^{-3}$	$3{,}713.9 \pm 1.0$	4,500,000	497.5 CPU s	7.6×10^8
$h = 3^{-2}$	$3{,}722.4 \pm 1.0$	4,050,000	177.6 CPU s	2.2×10^8
$h = 3^{-1}$	$3{,}986.1 \pm 1.0$	18,500,000	376.0 CPU s	3.3×10^8

TABLE 7.4

Performance of the weak trapezoidal method with $\theta = 1/2$, with crude Monte Carlo. Approximation is the 95% confidence interval.

Step size	Approximation	# paths	CPU time	# updates
$h = 3^{-4}$	$3{,}714.4 \pm 1.0$	4,750,000	2,120.5 CPU s	4.6×10^9
$h = 3^{-3}$	$3{,}714.6 \pm 1.0$	4,750,000	898.2 CPU s	1.6×10^9
$h = 3^{-2}$	$3{,}725.6 \pm 1.0$	4,800,000	349.8 CPU s	5.2×10^8
$h = 3^{-1}$	$3{,}673.3 \pm 1.0$	8,850,000	238.2 CPU s	3.2×10^8

TABLE 7.5

Approximation of $P\{X_{Dimer}(1) \geq 6{,}000\}$ using different methods and different step sizes. As expected, the weak trapezoidal method demonstrates significantly less bias than the Euler and midpoint methods. Approximation is the 95% confidence interval.

Method	Step size	# paths	Approximation
Exact	N.A.	4,520,000	0.02843 ± 0.00015
Euler	$h = 3^{-7}$	4,750,000	0.02818 ± 0.00015
Euler	$h = 3^{-6}$	4,750,000	0.02782 ± 0.00015
Midpoint	$h = 3^{-4}$	4,650,000	0.02718 ± 0.00015
Midpoint	$h = 3^{-3}$	4,500,000	0.02537 ± 0.00015
Weak trap, $\theta = 1/2$	$h = 3^{-4}$	4,750,000	0.02840 ± 0.00015
Weak trap, $\theta = 1/2$	$h = 3^{-3}$	4,750,000	0.02838 ± 0.00015
Weak trap, $\theta = 1/2$	$h = 3^{-2}$	4,800,000	0.02946 ± 0.00015

arise as much when using the weak trapezoidal method. In Table 7.4 we provide data on the performance of the weak trapezoidal method with various step sizes, combined with a crude Monte Carlo estimator (1.8). We see that for this example the midpoint method and the weak trapezoidal method are, overall, comparable. However, the weak trapezoidal method performs, in terms of bias and required CPU time, significantly better than the midpoint method for $h = 1/3$.

It is worth noting that both the midpoint and the weak trapezoidal methods compare decently on this example with the multilevel Monte Carlo method developed recently for stochastic chemical kinetic systems [4]. The choice of which method (an explicit solver discussed herein or a multilevel Monte Carlo solver) a user wishes to implement will therefore often be problem, and user, specific.

We next used each of the methods above to estimate the probability that the number of dimers at time 1 is greater than or equal to 6,000. Note that this probability is the expected value of the indicator function $1_{\{X_{\mathrm{Dimer}}(1) \geq 6{,}000\}}$. The results are presented in Table 7.5, which provides 95% confidence intervals for a few choices of h for each method. Note that in computing this approximation the weak trapezoidal method has significantly less bias than the midpoint method for comparable step sizes,

making it the method of choice for this particular choice of function f. The necessary CPU time for each of the methods is the same as those reported above.

REFERENCES

[1] D. F. ANDERSON, *A modified next reaction method for simulating chemical systems with time dependent propensities and delays*, J. Chem. Phys., 127 (2007), 214107.

[2] D. F. ANDERSON, *Incorporating postleap checks in tau-leaping*, J. Chem. Phys., 128 (2008), 054103.

[3] D. F. ANDERSON, A. GANGULY, AND T. G. KURTZ, *Error analysis of tau-leap simulation methods*, Ann. Appl. Probab., 21 (2011), pp. 2226–2262.

[4] D. F. ANDERSON AND D. J. HIGHAM, *Multilevel Monte Carlo for continuous time Markov chains, with applications in biochemical kinetics*, Multiscale Model. Simul., 10 (2012), pp. 146–179.

[5] D. F. ANDERSON AND T. G. KURTZ, *Continuous time Markov chain models for chemical reaction networks*, in Design and Analysis of Biomolecular Circuits: Engineering Approaches to Systems and Synthetic Biology, H. Koeppl, D. Densmore, G. Setti, and M. di Bernardo, eds., Springer, New York, 2011, pp. 3–42.

[6] D. F. ANDERSON AND J. C. MATTINGLY, *A weak trapezoidal method for a class of stochastic differential equations*, Commun. Math. Sci., 9 (2011), pp. 301–318.

[7] K. BALL, T. G. KURTZ, L. POPOVIC, AND G. REMPALA, *Asymptotic analysis of multiscale approximations to reaction networks*, Ann. Appl. Probab., 16 (2006), pp. 1925–1961.

[8] Y. CAO, D. T. GILLESPIE, AND L. R. PETZOLD, *Avoiding negative populations in explicit Poisson tau-leaping*, J. Chem. Phys., 123 (2005), 054104.

[9] Y. CAO, D. T. GILLESPIE, AND L. R. PETZOLD, *The slow-scale stochastic simulation algorithm*, J. Chem. Phys., 122 (2005), 014116.

[10] Y. CAO, D. T. GILLESPIE, AND L. R. PETZOLD, *Efficient step size selection for the tau-leaping simulation method*, J. Chem. Phys., 124 (2006), 044109.

[11] Y. CAO AND L. R. PETZOLD, *Trapezoidal tau-leaping formula for the stochastic simulation of biochemical system*, in Proceedings of Foundations of Systems Biology in Engineering (FOSBE) 2005, Santa Barbara, CA, 2005, pp. 149–152.

[12] A. CHATTERJEE AND D. G. VLACHOS, *Binomial distribution based τ-leap accelerated stochastic simulation*, J. Chem. Phys., 122 (2005), 024112.

[13] W. E, D. LIU, AND E. VANDEN-EIJNDEN, *Nested stochastic simulation algorithm for chemical kinetic systems with disparate rates*, J. Chem. Phys., 123 (2005), 194107.

[14] S. N. ETHIER AND T. G. KURTZ, *Markov Processes: Characterization and Convergence*, John Wiley & Sons, New York, 1986.

[15] C. GADGIL, C. H. LEE, AND H. G. OTHMER, *A stochastic analysis of first-order reaction networks*, Bull. Math. Biol., 67 (2005), pp. 901–946.

[16] M. GIBSON AND J. BRUCK, *Efficient exact stochastic simulation of chemical systems with many species and many channels*, J. Phys. Chem. A, 105 (2000), pp. 1876–1889.

[17] D. T. GILLESPIE, *A general method for numerically simulating the stochastic time evolution of coupled chemical reactions*, J. Comput. Phys., 22 (1976), pp. 403–434.

[18] D. T. GILLESPIE, *Exact stochastic simulation of coupled chemical reactions*, J. Phys. Chem., 81 (1977), pp. 2340–2361.

[19] D. T. GILLESPIE, *Approximate accelerated simulation of chemically reaction systems*, J. Chem. Phys., 115 (2001), pp. 1716–1733.

[20] D. T. GILLESPIE AND L. R. PETZOLD, *Improved leap-size selection for accelerated stochastic simulation*, J. Chem. Phys., 119 (2003), pp. 8229–8234.

[21] P. W. GLYNN, *A GSMP formalism for discrete event systems*, Proc. IEEE, 77 (1989), pp. 14–23.

[22] Y. HU, T. LI, AND B. MIN, *The weak convergence analysis of tau-leaping methods: Revisited*, Commun. Math. Sci., 9 (2011), pp. 965–996.

[23] Y. HU, T. LI, AND B. MIN, *A weak second order tau-leaping scheme for simulating chemical reaction systems*, J. Chem. Phys., 135 (2011), 024113.

[24] M. HUTZENTHALER AND A. JENTZEN, *Convergence of the stochastic Euler scheme for locally Lipschitz coefficients*, Found. Comput. Math., 11 (2011), pp. 657–706.

[25] H.-W. KANG AND T. G. KURTZ, *Separation of time-scales and model reduction for stochastic reaction networks*. Ann. Appl. Probab., to appear; available online from http://www.imstat.org/aap/future_papers.html.

[26] P. E. KLOEDEN AND E. PLATEN, *Numerical Solution of Stochastic Differential Equations*, Appl. Math. (N. Y.) 23, Springer-Verlag, Berlin, 1992.

[27] T. G. KURTZ, *Approximation of Population Processes*, CBMS-NSF Regional Conf. Ser. Appl. Math. 36, SIAM, Philadelphia, 1981.

[28] H. LAMBDA, J. C. MATTINGLY, AND A. M. STUART, *An adaptive Euler-Maruyama scheme for SDEs: Convergence and stability*, IMA J. Numer. Anal., 27 (2007), pp. 479–506.