Quantifying stochastic effects in biochemical reaction networks using partitioned leaping

Leonard A. Harris,[1,*] Aaron M. Piccirilli,[1,2] Emily R. Majusiak,[1] and Paulette Clancy[1,†]
[1] *School of Chemical and Biomolecular Engineering, Cornell University, Ithaca, New York 14853, USA*
[2] *Department of Computer Science, Cornell University, Ithaca, New York 14853, USA*

(Received 9 December 2008; revised manuscript received 17 February 2009; published 12 May 2009)

"Leaping" methods show great promise for significantly accelerating stochastic simulations of complex biochemical reaction networks. However, few practical applications of leaping have appeared in the literature to date. Here, we address this issue using the "partitioned leaping algorithm" (PLA) [L. A. Harris and P. Clancy, J. Chem. Phys. **125**, 144107 (2006)], a recently introduced multiscale leaping approach. We use the PLA to investigate stochastic effects in two model biochemical reaction networks. The networks that we consider are simple enough so as to be accessible to our intuition but sufficiently complex so as to be generally representative of real biological systems. We demonstrate how the PLA allows us to quantify subtle effects of stochasticity in these systems that would be difficult to ascertain otherwise as well as not-so-subtle behaviors that would strain commonly used "exact" stochastic methods. We also illustrate bottlenecks that can hinder the approach and exemplify and discuss possible strategies for overcoming them. Overall, our aim is to aid and motivate future applications of leaping by providing stark illustrations of the benefits of the method while at the same time elucidating obstacles that are often encountered in practice.

DOI: 10.1103/PhysRevE.79.051906 PACS number(s): 82.39.−k, 87.10.Rt, 87.18.Tt, 87.18.Vf

I. INTRODUCTION

Biological systems are inherently noisy or *stochastic*. A primary source of this noise is the random nature of molecular interactions that predominates when molecular copy numbers are low, so-called "intrinsic" noise [1–8]. Intrinsic noise has been implicated as the source of nongenetic variability in clonal cell populations [9–11] and can profoundly affect the dynamical behavior of a biological system both to its benefit as well as its detriment [2]. For example, stochastic effects in gene expression have been shown to speed the response of yeast cells to a challenge, allowing them to survive in conditions where they otherwise could not [12]. Conversely, noise can interfere with the workings of circadian clocks [13,14], systems whose reliability is essential for survival. As such, nature has developed regulatory mechanisms to attenuate the noise [15,16]. All in all, accounting for the effects of stochasticity is *essential* for gaining a clear understanding of the design principles underlying many biological processes.

Nevertheless, biological-systems modeling today continues to be performed primarily at the continuous-deterministic scale, usually in the form of ordinary differential equations (ODEs). These formalisms explicitly *ignore* stochastic fluctuations [17,18]. However, the practice understandably persists because the methods are well established and numerous easy-to-use algorithms are available at little to no cost. Moreover, established stochastic methods, such as Gillespie's stochastic simulation algorithm (SSA) [19–21], are in many cases simply too computationally intensive to apply to realistic models of biological networks [22,23].

There is great interest, therefore, in developing accelerated stochastic simulation methods that can accurately capture noise effects but at significantly reduced computational cost relative to standard approaches. Ultimately, it is hoped that these methods will supplant ODEs as the default method of choice in computational systems biology. Approaches that have been proposed in this regard include (but are not limited to) (i) modification and optimization of Gillespie's original SSA [24–27], (ii) "leaping" methods which ignore the exact moments at which reaction firings occur [23,28–45], and (iii) "hybrid" methods which couple different simulation techniques (e.g., the SSA and ODEs) into a single, overarching algorithmic framework [46–54]. Of these, leaping methods are particularly popular presumably because of the sound theoretical foundation on which they stand [23,55].

Despite their popularity, however, very few practical applications of leaping have appeared in the literature to date [38]. This is a curious fact that has yet to be fully explored or explained. Moreover, those applications that have appeared (e.g., [56–58]) are generally brief in their presentation of the algorithms used and do not report much by way of the capabilities and limitations of the method. Thus, it is difficult, if not impossible, to project the potential utility of the approach onto other and more complex biological networks.

Here, we address this issue by using the "partitioned leaping algorithm" (PLA) [45], a recently introduced extension and variant of the τ-leaping method of Gillespie and coworkers [23,28,33,34], to systematically investigate the effects of stochasticity in two model biochemical reaction networks. The systems that we consider are intuitively simple yet they contain attributes that are ubiquitous to complex biological networks, such as enzyme catalysis and feedback control. We perform a detailed and in-depth investigation using leaping with the aim of illuminating both the capabilities and limitations of the method and, hence, aiding and motivating future applications of the approach. The PLA operates on the same basic principles that underlie all leaping algorithms and its performance with respect to these systems can thus be seen as generally reflective of the entire class of method. That being said, there are certain aspects of the

[*] Present address: Department of Computational Biology, University of Pittsburgh School of Medicine, Pittsburgh, Pennsylvania 15260, USA; lharris@pitt.edu

[†] pqc1@cornell.edu

TABLE I. Kummer *et al.* [59] core model for calcium oscillations in hepatocytes. "∅" represents a source or a sink and $k_2 = 2.85\ \mathrm{s}^{-1}$ puts the system into the "periodic-bursting" regime (see text). G_α represents the activated α subunit of the intracellular receptor-bound G protein, PLC* the activated form of phospholipase C, and Ca cytosolic calcium ions. Note that to perform stochastic simulations all parameters must be devoid of molar units (M). Parameters with molar units are thus multiplied by $N_A\Omega$ (Avogadro's number × system volume) prior to runtime. Initial conditions: $[G_\alpha]=[\mathrm{PLC}^*]=[\mathrm{Ca}]=0.01M$.

	Reaction	Rate expression	Parameter value(s)
1	$\varnothing \rightarrow G_\alpha$	k_1	$k_1=0.212M\ \mathrm{s}^{-1}$
2	$G_\alpha \rightarrow 2G_\alpha$	$k_2[G_\alpha]$	$k_2=2.85\ \mathrm{s}^{-1}$
3	$G_\alpha+\mathrm{PLC}^* \rightarrow \mathrm{PLC}^*$	$k_3[G_\alpha][\mathrm{PLC}^*]/(K_4+[G_\alpha])$	$k_3=1.52\ \mathrm{s}^{-1}$, $K_4=0.19M$
4	$G_\alpha+\mathrm{Ca} \rightarrow \mathrm{Ca}$	$k_5[G_\alpha][\mathrm{Ca}]/(K_6+[G_\alpha])$	$k_5=4.88\ \mathrm{s}^{-1}$, $K_6=1.18M$
5	$G_\alpha \rightarrow G_\alpha+\mathrm{PLC}^*$	$k_7[G_\alpha]$	$k_7=1.24\ \mathrm{s}^{-1}$
6	$\mathrm{PLC}^* \rightarrow \varnothing$	$k_8[\mathrm{PLC}^*]/(K_9+[\mathrm{PLC}^*])$	$k_8=32.24M\ \mathrm{s}^{-1}$, $K_9=29.09M$
7	$G_\alpha \rightarrow G_\alpha+\mathrm{Ca}$	$k_{10}[G_\alpha]$	$k_{10}=13.58\ \mathrm{s}^{-1}$
8	$\mathrm{Ca} \rightarrow \varnothing$	$k_{11}[\mathrm{Ca}]/(K_{12}+[\mathrm{Ca}])$	$k_{11}=153.0M\ \mathrm{s}^{-1}$, $K_{12}=0.16M$

PLA, which we will expound upon below, that make it particularly appealing from a practical point of view.

We begin in Sec. II by introducing the biochemical reaction networks that we investigate in this paper. We then briefly describe in Sec. III the PLA, the time series analysis method, and the statistical tests employed in this work. Detailed results for the two networks are subsequently presented in Sec. IV. We conclude in Sec. V with a discussion of the implications of these results, a possible explanation for why practical applications of leaping are so scarce in the literature, and the future outlook for leaping methods in computational systems biology.

II. NETWORKS

The systems that we investigate are a core model for calcium oscillations in hepatocytes introduced by Kummer *et al.* [59] and the three-gene "repressilator" of Elowitz and Leibler [13]. These systems are relatively simple, yet they are not "toy" problems in the sense that they contain nontrivial features that are ubiquitous to biochemical systems, such as enzyme catalysis and feedback control. Moreover, both systems emit large-amplitude oscillations which give rise to the kinds of wide disparities in species populations that leaping algorithms are specifically designed to cope with [23,45]. All in all, these systems provide an ideal test bed for investigating the practical utility of leaping methods in computational systems biology.

Our investigation entails using the PLA to probe behavioral changes that arise in these systems due to changes in various system properties. Specifically, we investigate the transition from stochastic to deterministic behavior that accompanies increases in the system volume (i.e., population levels) in the calcium-oscillations model and increases in the gene-protein binding and unbinding rate constants in the repressilator. The salient feature of our investigation is that we are able to ascertain, in a systematic way, the performance characteristics of the leaping algorithm over a wide spectrum of conditions. We thus identify cases where leaping proves particularly beneficial, where it "bogs down," and various points in between. Further details of the networks are provided in the sections below.

A. Calcium oscillations

Intracellular calcium is an important second messenger for the functioning of many cell types, both in plants and in animals. It is involved in a multitude of functions during the lifetime of a cell, including fertilization, development, and death [60]. The dynamics of intracellular calcium are not smooth and continuous, however. Rather, they are driven by small numbers of receptors and ion channels that can give rise to highly stochastic behavior. Indeed, experiments have shown that calcium waves are triggered by elementary stochastic events known as "blips" and "puffs" [61]. Incorporating stochasticity into models of calcium oscillations is thus of high interest.

Many theoretical models have been proposed to describe the oscillatory dynamics of intracellular calcium [61,62]. Kummer *et al.* [59] proposed a model for calcium oscillations in hepatocytes (liver cells) that displays a rich variety of behaviors. The model features self-enhanced activation of the G_α subunit of the receptor complex and is able to capture many aspects of experimentally observed behavior that eluded previous models. The authors also presented a simplified version of the model that displays the same basic behaviors as the full model, thus emphasizing the "core" mechanisms driving the oscillations [59].

In Table I, we show the Kummer *et al.* [59] core model for calcium oscillations in hepatocytes. The model consists of eight reactions involving three species: the activated form of phospholipase C (PLC*), the α subunit of the receptor-bound G protein (G_α), and cytosolic calcium ions (Ca). Note that the model is in a reduced form, with degradation processes described in terms of Michaelis-Menten kinetics. Reaction 2, which is the prime feature of this model, describes the agonist-initiated [e.g., adenosine triphosphate (ATP)] autocatalytic activation of the G_α subunit. The parameter k_2 thus amounts to the product of the second-order association constant and the agonist concentration and is a primary determinant of the system behavior. Kummer *et al.* [59,63] showed that with increasing k_2 the system behavior transitions from simple Ca^{2+} spiking oscillations, to complex oscillations, to chaotic behavior, and finally, to an elevated steady state.

We also see in Table I that the model contains various feedback loops which drive the oscillatory behavior of the network. Specifically, PLC^* and Ca are created autocatalytically in reactions 5 and 7, respectively, through the action of G_α. In reactions 3 and 4, however, G_α is degraded enzymatically by the actions of PLC^* and Ca, respectively. Thus, in the correct parameter range, increased levels of G_α lead to increased levels of PLC^* and Ca which, in turn, lead to increased degradation of G_α, which leads to decreased levels of PLC^* and Ca, and so on and so forth.

In Ref. [63], Kummer *et al.* compared the deterministic behavior of this model to results of stochastic simulations performed using the SSA. The goal was to determine points of transition to determinism for various dynamical regimes of the model (e.g., "periodic spiking," "periodic bursting," "chaos") and to provide general insight as to when a deterministic treatment is applicable and when a stochastic approach is necessary. SSA simulations were performed for various system sizes (with fixed concentrations) and the point of transition to determinism was estimated via *visual* comparison of stochastic and deterministic time courses. Visual inspection was necessary because of the high computational expense of the SSA [63].

Here, we extend the analysis of Kummer *et al.* for the periodic-bursting regime, a main focus of Ref. [63]. The regime is characterized by complex Ca^{2+} oscillations comprised of three-peak complexes (see below), behavior that is reminiscent of that seen experimentally in hepatocytes stimulated by ATP [59,63,64]. Using the PLA and the peak-analysis tool described in Sec. III, we collect large amounts of peak amplitude and peak-to-peak distance data at various system volumes and *quantify* the relationship between stochasticity and system size, something that was not feasible in Ref. [63] because of the limitations of the SSA. This allows us to pinpoint, from a statistical perspective, the points of transition to determinism. As we shall see, these differ, to some extent, from those reported in [63].

B. Repressilator

Synthetic biology is a relatively new and rapidly growing scientific field [65–68]. In analogy with electrical circuit design, synthetic biologists attempt to use their knowledge of fundamental biological principles to design and construct artificial biological "circuits" that confer novel function unto their host. In this way, one can isolate and control specific aspects of a biological process and circumvent the immense complexity of natural biological systems, providing a means by which current theoretical understanding can be tested and scrutinized. Moreover, the long-term goal is to develop protocols for logical control. One can envision a time when microorganisms are "programmed" at the genetic level to carry out important functions, such as cleaning up oil spills or delivering tumor-suppressing drugs to specific locations within the body [65,66].

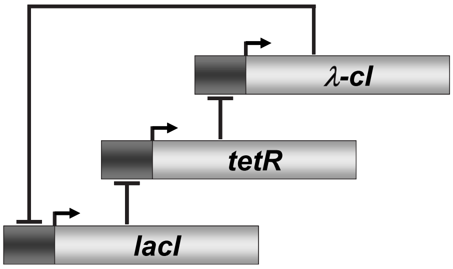

FIG. 1. Schematic diagram of the repressilator. Each gene (*lacI*, *tetR*, λ-*cI*) produces a protein which binds to the operator site of the promoter driving expression of the next gene in the sequence, thus repressing it. Within the correct region of parameter space the repressilator oscillates, a so-called "ring oscillator" [66].

Numerous artificial biological circuits have been constructed in bacteria and demonstrated to perform as designed. One such network is the repressilator, a three-gene synthetic genetic regulatory network developed by Elowitz and Leibler [13]. Each gene in the repressilator produces a protein which represses the next gene in the sequence; the protein product of the last gene represses the first gene, thus closing the loop. This construct is known in microelectronics as a "ring oscillator" [66]. As implemented experimentally in *Escherichia coli* [13], the repressilator consists of the genes *lacI*, *tetR*, and λ-*cI*; LacI protein represses *tetR*, etc. (Fig. 1). [Standard convention is to denote genes in italicized font beginning with a lowercase letter (e.g., *tetR*), mRNA transcript in nonitalicized font beginning with a lowercase letter (e.g., tetR), and proteins in nonitalicized font beginning with a capital letter (e.g., TetR).]

Under the right conditions, i.e., within the correct region of parameter space, the repressilator oscillates, acting as a biological clock. However, determining the conditions for oscillation is nontrivial and theoretical modeling was employed to identify the appropriate design criteria [13]. Once functional, a particularly interesting experimental observation was the significant fluctuations in amplitude and period exhibited by the circuit. Natural oscillators, such as circadian clocks, do not exhibit such variability [16,69,70] and subsequent modeling indicated that nature must employ some form of regulatory control in order to overcome the problem [14,15]. The repressilator thus succeeded in providing valuable insight regarding the design principles underlying an important biological process.

The extensive use of modeling in the design and analysis of the repressilator, as well as the highly stochastic behavior exhibited by the network, motivates our investigation using the PLA. In Table II, we show the basic form of one-third of the repressilator model (all three genes are considered equivalent). This corresponds to the "stochastic" model of Elowitz and Leibler [13]. Here, all reactions are treated as elementary using simple mass-action kinetics (i.e., rates directly proportional to the reactant population levels). Each gene is assumed to have two binding sites for repressor protein, with binding occurring sequentially, and the unbound gene transcribes mRNA 1000 times faster than the singly- or doubly-bound gene. mRNA also translates protein autocatalytically and mRNA and protein degrade with half lives of 120 and 600 s, respectively [13].

TABLE II. Basic form of one third of the full repressilator model (all three genes are considered equivalent) [13]. g_x represents one of the three repressilator genes (*lacI*, *tetR*, or λ-*cI*) and p_r the corresponding repressor protein (LacI for *tetR*, etc.; see Fig. 1). m_x and p_x represent the mRNA and protein products of g_x, respectively. All reactions are treated using simple mass-action kinetics and all parameters with inverse molar units (M^{-1}) are divided by $N_A\Omega$ prior to runtime. k_1,k_2 are rate constants for forward repressor binding while k_{-1},k_{-2} are for the reverse reactions. Also shown (see text for explanation) are the "telegraph factor" γ, the "RNA factor" η, and the "protein factor" ρ (equivalent for all genes) [71,72]. Here, we set $\eta=\rho=1000$ and vary $10^{-4}\leq\gamma\leq 1$. Initial conditions: $[m_{tetR}]=3.8\ \mu M$; $[m_{CI}]=8.1\ \mu M$; $[m_{lacI}]=0.15\ \mu M$; $[p_{tetR}]=0.22$ mM; $[p_{CI}]=2.4$ mM; $[p_{lacI}]=0.20$ mM; $g_{tetR}=g_{CI}=g_{lacI}=1$ (molecule); all $\{g_x\cdot p_r\}$ and $\{g_x\cdot p_r\cdot p_r\}=0$.

	Reaction	Parameter value
$\vec{1},\overleftarrow{2}$	$g_x+p_r\rightleftharpoons\{g_x\cdot p_r\}$	$\begin{cases} k_1=10^9\boldsymbol{\gamma}/\boldsymbol{\rho}M^{-1}\ \text{s}^{-1} \\ k_{-1}=224.0\boldsymbol{\gamma}\ \text{s}^{-1}\end{cases}$
$\vec{3},\overleftarrow{4}$	$\{g_x\cdot p_r\}+p_r\rightleftharpoons\{g_x\cdot p_r\cdot p_r\}$	$\begin{cases} k_2=10^9\boldsymbol{\gamma}/\boldsymbol{\rho}M^{-1}\ \text{s}^{-1} \\ k_{-2}=9.0\boldsymbol{\gamma}\ \text{s}^{-1}\end{cases}$
5	$g_x\rightarrow g_x+m_x$	$k_3=0.5\boldsymbol{\eta}\ \text{s}^{-1}$
6	$\{g_x\cdot p_r\}\rightarrow\{g_x\cdot p_r\}+m_x$	$k_4=5\times10^{-4}\boldsymbol{\eta}\ \text{s}^{-1}$
7	$\{g_x\cdot p_r\cdot p_r\}\rightarrow\{g_x\cdot p_r\cdot p_r\}+m_x$	$k_5=5\times10^{-4}\boldsymbol{\eta}\ \text{s}^{-1}$
8	$m_x\rightarrow m_x+p_x$	$k_6=0.167\boldsymbol{\rho}/\boldsymbol{\eta}\ \text{s}^{-1}$
9	$m_x\rightarrow\varnothing$	$k_7=\ln(2)/120\ \text{s}^{-1}$
10	$p_x\rightarrow\varnothing$	$k_8=\ln(2)/600\ \text{s}^{-1}$

TABLE III. Basic form of one third of the *reduced* repressilator model. Parameter values are the same as in Table II. The *Adair* functional forms [73] describing mRNA production are similar to the well-known Hill forms, but are formally correct for $\gamma\rightarrow\infty$ (see Appendix A). $K_i\equiv k_{-i}/k_i$ $(i=1,2)$, $f([p_r])\equiv K_1K_2+K_2[p_r]+[p_r]^2$.

	Reaction	Rate expression
1	$g_x\rightarrow g_x+m_x$	$k_3K_1K_2/f([p_r])$
2	$\{g_x\cdot p_r\}\rightarrow\{g_x\cdot p_r\}+m_x$	$k_4K_2[p_r]/f([p_r])$
3	$\{g_x\cdot p_r\cdot p_r\}\rightarrow\{g_x\cdot p_r\cdot p_r\}+m_x$	$k_5[p_r]^2/f([p_r])$
4	$m_x\rightarrow m_x+p_x$	$k_6[m_x]$
5	$m_x\rightarrow\varnothing$	$k_7[m_x]$
6	$p_x\rightarrow\varnothing$	$k_8[p_x]$

We also include in Table II various multiplicative factors: a "telegraph factor" γ, an "RNA factor" η, and a "protein factor" ρ [71,72]. These factors allow us to control and tune the various sources of noise in the system. For example, increasing η increases the rates of gene transcription, resulting in larger mRNA populations and less mRNA-related "shot noise," i.e., noise arising from the fact that the system is comprised of discrete numbers of interacting entities (in electrical circuits, shot noise arises from discrete numbers of charge carriers; in optical devices, from discrete numbers of photons) [71,72]. The translation rate is divided by η, however, thus canceling out the effect of increased mRNA levels on the protein production rates. Protein-related shot noise is controlled similarly through the protein factor ρ while the amount of "telegraph noise," i.e., that associated with the random switching between the ON and OFF states of the genes (reminiscent of an electronic telegraph transmitting Morse code) [71,72], is controlled through the parameter γ.

In this paper, we focus primarily on the telegraph factor γ. We do so because the performance of the leaping algorithm is strongly affected by this parameter: at small values the method performs exceptionally well but falters as it is increased, approaching the performance of the SSA (see Sec. IV A). With the system volume $\Omega=1.4\times10^{-15}$ l (the volume of a typical *E. Coli* cell) and η and ρ set to high values (i.e., 1000) in order to dampen the mRNA- and protein-related noise sources, we investigate how the system behavior changes for $10^{-4}\leq\gamma\leq1$. We thus observe how the actual values of the gene-protein binding and unbinding rate constants, as opposed to simply their ratios, affect the overall dynamical behavior of the system as well as the performance of the PLA.

We also find it convenient to investigate a reduced form of the repressilator model obtained by applying the "partial equilibrium assumption" (PEA) to the first four reactions in Table II. Assuming each reversible reaction to be in rapid equilibrium, simple algebra leads to effective rate expressions of the *Adair* form [73] for mRNA production from the free, singly-bound, and doubly-bound genes (see Appendix A for derivations). These expressions are strictly valid in the limit $\gamma\rightarrow\infty$. Doing so reduces the 30 reactions of Table II to 18 in Table III. Note that the reduced model in Table III differs from the "deterministic" model of Elowitz and Leibler [13] in that the expressions in Table III are directly derivable from the reactions in Table II via application of the PEA while those in Ref. [13] are not.

III. METHODS

To carry out our investigations, we use the partitioned leaping algorithm [45], an extension and variant of the τ-leaping method of Gillespie and co-workers [23,28,34,55]. The PLA operates on the same basic principles that underlie all leaping methods: calculate a time step τ over which all reaction rates in the system remain "essentially constant" and then determine the number of times each reaction fires within that interval by sampling from an appropriate probability distribution. The primary difference between our approach and other leaping algorithms is that we utilize the *entire* theoretical framework developed by Gillespie [23,55] for bridging from the discrete-stochastic description of reaction dynamics to the more familiar continuous-deterministic representation [45].

At each step of a PLA simulation, reactions are partitioned, using theoretically sound criteria [23,55], into four categories based on the calculated time step and the current reactant population levels. The categories correspond to different levels of approximation; reactions with small reactant populations garner a detailed "exact-stochastic" classification (i.e., a SSA treatment) while those with larger populations receive coarser descriptions. The coarse classifications range

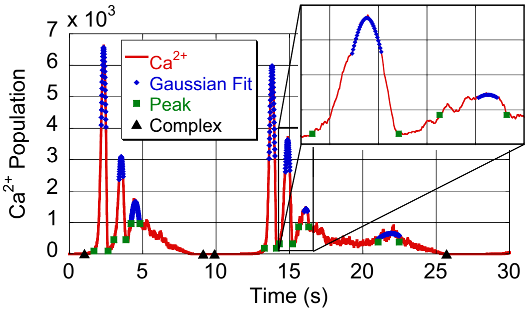

FIG. 2. (Color online) Example calcium-oscillations time course and the Gaussian fits obtained using the peak-analysis software employed in this work. Results are for a system volume $\Omega = 10^{-21}$ l. Also shown are peak and peak-complex bracket points identified by the fitting algorithm. Notice that stochastic effects lead at this small volume to the identification of a fourth peak in the second peak complex. (Inset) Blown up view of the second and third peaks in the second peak complex. Squares correspond to where fitting began; diamonds to where fitting concluded.

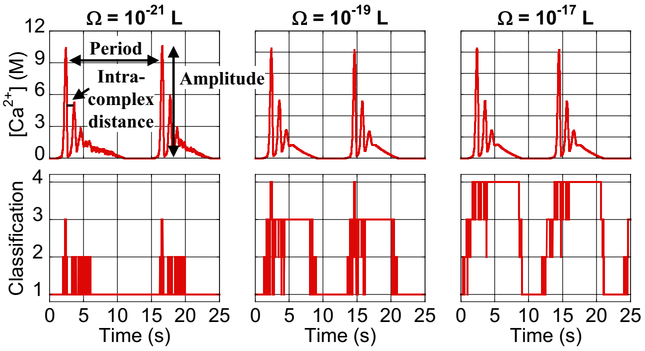

FIG. 3. (Color online) Example Ca^{2+} periodic-bursting time courses and associated classifications for $G_\alpha + Ca \rightarrow Ca$ (Table I, reaction 4) obtained using the PLA at three different system volumes. Classifications are (1) exact stochastic, (2) Poisson, (3) Langevin, and (4) deterministic. Also shown (*top-left panel*) are the three system attributes investigated: first-peak amplitudes, first-to-second intracomplex distances, and first-to-first intercomplex periods.

from "Poisson" to "Langevin" to "deterministic," with the levels of stochasticity decreasing and approximation increasing as one moves up the hierarchy. The result is a truly multiscale method, where fluctuations associated with rare events are correctly described while frequent events "leapt" over multiple reaction firings at a time. As such, the PLA accomplishes what so-called hybrid methods [46–54] aim to do but in a much more simple and theoretically sound way. For even moderately sized systems the computational gains of the PLA relative to the SSA can be significant [45]. We refer the interested reader to Ref. [45] for further details regarding the theoretical foundations and implementation of the PLA.

The promise of the PLA, and leaping algorithms in general, is that long-time stochastic simulations can be performed, allowing for large-scale data collection and quantitative statistical analyses of the resulting time series. However, leaping simulations produce noisy time-evolution trajectories and automated data collection tools that can compensate for noise are needed. Here, we use in-house time-domain peak-analysis software for this purpose. Borrowing ideas from the automated identification of peaks in mass spectral data [74,75], the software identifies "significant" peaks within a time series and fits Gaussians to the data in order to wash out the noise. An example calcium-oscillations time series and the Gaussian fits achieved using the peak-analysis software are shown in Fig. 2.

Using this tool, we collect large amounts of peak amplitude and peak-to-peak distance data from simulated time series and perform various statistical analyses. We calculate averages and variances from long-time PLA and deterministic simulation runs (see [76] for an explanation of variance in the deterministic case) and perform z tests on the differences in means and F tests on the ratios of variances [77]. We also calculate coefficients of variation (COVs), defined as the ratio of the standard deviation to the mean [4], in order to quantify the relative *importance* of the noise. Finally, we put the data into the form of smoothed histograms [45] and calculate "histogram distances," D, and "self-distances," $D_{\mathrm{Ref}}^{\mathrm{self}}$ [45,78], so as to account for any particulars in the *shapes* of the distributions (e.g., long tails, bimodal features, etc.). We do all of this for various system properties (i.e., volumes, telegraph factors) in order to quantify changes in the system behavior and to identify points of transition to determinism.

All PLA simulations reported in this work were performed using the parameters "≈ 1"=3, "$\gg 1$"=100 [45], and using the species-based τ-selection procedure of Cao *et al.* [34], as modified in Harris and Clancy [45], with "error control parameter" $\epsilon = 0.03$. These represent "typical" values for these parameters. Derivations of the g_i values [34,45] used in τ selection for the Michaelis-Menten and Adair reactions of Tables I and III are given in Appendix B. It should also be noted that an attractive feature of the PLA is its ability, via simple manipulation of the classification parameters, to force simulations at any level of description [45]. Thus, both deterministic and exact-stochastic simulations reported in this work were performed using the same code as for the PLA simulations. The PLA segues to an explicit Euler method in the deterministic limit and to the next-reaction method [24] in the exact-stochastic limit.

IV. RESULTS

A. Calcium oscillations

The periodic-bursting regime of the Kummer *et al.* [59] calcium-oscillations model (Table I) is characterized by large-amplitude complex oscillations in which the Ca^{2+} repeating unit is a three-peak complex. In Fig. 3, we show example time courses at three different system volumes spanning 4 orders of magnitude obtained using the PLA. Also shown are the classifications achieved along the time courses for the reaction $G_\alpha + Ca \rightarrow Ca$ (Table I, reaction 4). The classifications range from 1 to 4, with 1 being the finest level of description (exact stochastic) and 4 the coarsest (deterministic).

The plots in Fig. 3 starkly illustrate why this system is ideally suited for treatment via the PLA: *the classifications oscillate in time along with the reactant species populations*.

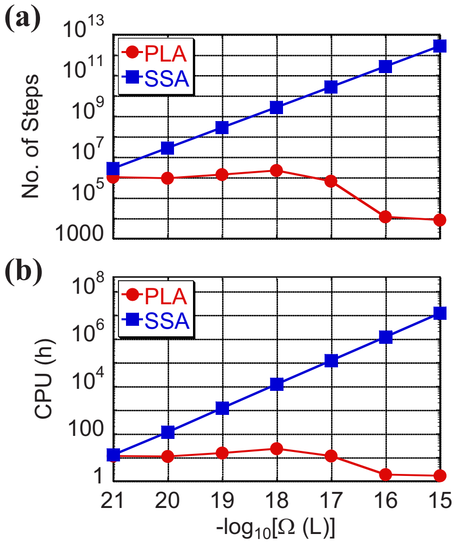

FIG. 4. (Color online) (a) Average numbers of steps from and (b) total CPU times for 10 000 PLA and SSA simulation runs of 20 s for the Kummer *et al.* [59] core model for calcium oscillations (Table I). Both graphs are shown as log-log plots. SSA values at $\Omega = 10^{-20}$ and 10^{-19} l are based on 1000 and 100 simulation runs, respectively. SSA values at $\Omega \geq 10^{-18}$ l are extrapolations (not based on actual data). Note that the PLA steps and CPU times go through maxima at $\sim\Omega = 10^{-18}$ l. Similar behavior was observed for the example systems in Ref. [45]. Also note that in the case of the SSA, the linear relationship between computational expense and system size [19–21], which has the form $y = mx$, with m being the slope (the y intercept is zero since, obviously, a system of zero size requires zero computational effort), appears here as a line with a slope of unity and y intercept of $\log_{10}(m)$. All simulations were performed on a 3.60 GHz Pentium Xeon processor.

When the Ca^{2+} population is small we see that the reaction gets classified at the exact-stochastic level, while coarser descriptions are employed when the population is large (similar behavior is seen for other reactions in the system as well: data not shown). As such, the PLA is able to accurately capture stochastic effects that arise in this system when the species populations become small without suffering from the characteristic inefficiency of the SSA when the populations become large.

This is evident in Fig. 4, where we show results of a step and timing analysis comparing the performance of the PLA to the SSA. As expected, we see a linear increase in the computational expense of the SSA with increasing system size (see Fig. 4, caption) [19–21]. The PLA, on the other hand, exhibits more complex behavior, with the expense initially remaining constant, then increasing slightly, going through a maximum at $\sim\Omega = 10^{-18}$ l and then dropping off sharply before finally leveling off. Interestingly, similar behavior was seen for the simple example systems in Ref. [45], which were specifically designed to showcase the strengths of the algorithm. Most importantly, however, is that Fig. 4 clearly illustrates that for all but the smallest system size considered the PLA *far* outperforms the SSA by as many as 8 orders of magnitude in simulation steps at $\Omega = 10^{-15}$ l. It is these types of accelerations that make quantifying stochastic effects in this system possible, something that was unachievable in Ref. [63] because of the limitations of the SSA.

Our statistical results are shown in Fig. 5. In all cases, we compare results obtained from both PLA and SSA simulations to deterministic predictions for the three attributes considered, namely, first-peak amplitudes, first-to-second intracomplex distances, and first-to-first intercomplex periods (see Fig. 3, top-left panel). In the case of the SSA, we were only able to obtain data for the three smallest system sizes considered because of the computational expense of the method.

In Figs. 5(a)–5(c), we compare averages and modes obtained from the PLA and SSA to deterministic predictions. The results are shown as percent deviations from determinism. In all cases, we see small yet statistically significant deviations from determinism at small volumes and, in the case of the PLA, a rapid convergence to the deterministic limit with increasing system size. Close inspection reveals that full convergence is achieved for all attributes by $\Omega = 10^{-18}$ l. It is also clear in Figs. 5(a) and 5(b) that there are discrepancies between the PLA results and the SSA results. The discrepancies are small, however, on the order of 1% or less in all cases, and decrease with decreasing ϵ (data not shown). Interestingly, there are virtually no discrepancies between the PLA results and the SSA results in Fig. 5(c), the intercomplex periods. We cannot at present explain why the PLA achieves greater accuracy for this attribute over the others. Understanding the sources of error in leaping algorithms and developing strategies for attenuating them is an area of current interest [29,30]. Suffice it to say that in this case the PLA achieves very good to excellent accuracy for all quantities considered.

In Figs. 5(d) and 5(e), we consider the *distributions* of the attributes. Figure 5(d) shows data for standard deviations, a simple point statistic, while Fig. 5(e) considers the shapes of the distributions through the histogram distance [45,78]. In Fig. 5(d), we see almost perfect correspondence between the PLA and the SSA results. In Fig. 5(e), however, we see discrepancies in the histogram distances for the amplitude and the intracomplex distance (see inset). Taken together, along with Figs. 5(a) and 5(b), this indicates that the PLA is accurately capturing the shapes of the distributions but they are shifted slightly relative to those obtained with the SSA.

As far as convergence to determinism, both Figs. 5(d) and 5(e) give the same result: the different attributes converge to the deterministic limit at different rates and with different transition points. The intracomplex distance converges the fastest, followed by the peak amplitude, and finally the intercomplex period. The amplitude and intracomplex distance statistically converge to the deterministic limit at $\Omega = 10^{-17}$ l while the period converges at 10^{-15} l. These convergence points differ from those for the averages by 1–3 orders of magnitude [cf. Figs. 5(a)–5(c)] and indicate a persistence of noise in this system at volumes much larger than expected based on the analysis of Ref. [63].

Finally, in Fig. 5(f) we consider the relative "importance" of the noise through the COV. The idea is that even if noise in an attribute is significant from a statistical perspective it might be so subtle as to be of little practical import. For example, in this case we see that for $\Omega \geq 10^{-20}$ l the COVs for all attributes are less than a few percent [the discrepan-

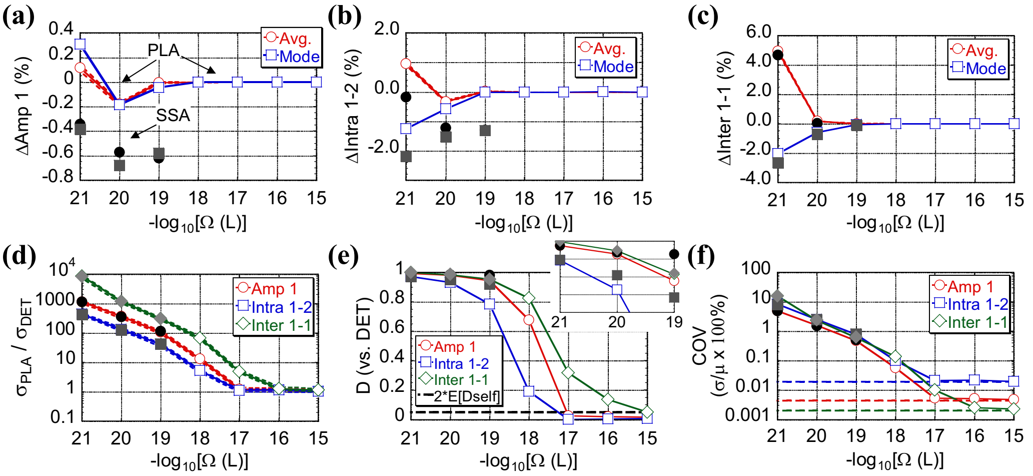

FIG. 5. (Color online) Statistical results for the Ca^{2+} periodic-bursting regime. Results of PLA and SSA simulations are compared to deterministic (DET) predictions. PLA results are shown as empty symbols (circle, square, diamond) connected by lines. SSA results are shown as disconnected, filled symbols. PLA and SSA points designated with the same symbol correspond to the same quantity [e.g., in (a), the PLA averages are given as empty circles while the SSA averages are shown as filled circles]. All PLA and deterministic values are based on over 10 000 collected data points. For the SSA, over 10 000 data points were collected for $\Omega = 10^{-21}$ and 10^{-20} l and ~ 1000 were collected for $\Omega = 10^{-19}$ l. No SSA results are given for $\Omega \geq 10^{-18}$ l due to computational expense. (a)–(c) Deviations from determinism, shown as percentages [({PLA or SSA}−DET)÷DET×100%], for averages and modes of Ca^{2+} first-peak amplitudes (Amp 1), first-to-second intracomplex distances (Intra 1–2), and first-to-first intercomplex periods (Inter 1–1), respectively (see Fig. 3, *top-left panel*). Dashed lines denote 95% confidence intervals on the PLA averages [difficult to see in (b) and (c)]. Note that long-tailed distributions lead to averages and modes on opposite sides of the deterministic predictions at small volumes in (b) and (c). (d) Ratios of standard deviations ({PLA or SSA}÷DET) for the three attributes in (a)–(c). Dashed lines denote 80% confidence intervals (because of the relative weakness of the *F* test [77]). (e) Histogram distances ({PLA or SSA} vs DET) for the three attributes in (a)–(c). The dashed line denotes *twice* the deterministic self-distance ($2 \times E[D_{DET}^{self}]$) (see [79]). The self-distances for all three attributes are essentially identical in this case. (f) COVs obtained from PLA and SSA simulations, shown as percentages (standard deviation÷average×100%), for the three attributes in (a)–(c). Deterministic limits are given as dashed lines.

cies between the PLA and the SSA seen in Figs. 5(a) and 5(b) are virtually indiscernible on this scale]. The noise effects clearly persist up until 10^{-17} l [as seen in Figs. 5(d) and 5(e) as well] but it seems unlikely that in any realistic setting, e.g., an embedding within a larger "whole-cell" model, they would be of much practical consequence. Whether or not this is true (it is debatable [6,7]), it is certainly the case that it would be difficult, if not impossible, to perceive these effects visually. This explains, therefore, why Kummer *et al.* [63] reported the stochastic-to-deterministic transition point for this model to be at $\sim 10^{-20}$ l (tens of thousands of Ca^{2+} ions). Our results thus largely corroborate their claim that a deterministic treatment is justified for volumes larger than this.

B. Repressilator

Our analysis of the repressilator focuses on behavioral changes that arise when the intermittent rates of switching between the transcriptional ON and OFF states of the genes are varied. The parameter that controls this is the telegraph factor γ. From an intuitive standpoint, we expect to observe large deviations from determinism at small values of γ and a convergence toward deterministic behavior with increasing γ because of the "averaging out" of the states of the genes [4]. Moreover, by making the RNA and protein factors, η and ρ, large we minimize the effects of shot noise. However, we cannot eliminate it completely and thus we expect to encounter some residual effects. Finally, we also expect that the PLA simulations will begin to bog down as γ is increased because of the growing disparities between the gene-protein binding and unbinding rates and the rates of all other reactions in the system [45].

In Figs. 6–9, these expectations are realized. In Fig. 6, we show example time courses for TetR protein (taken as representative of the system behavior) that illustrate how "deviant effects" [6] arise in the repressilator at small values of γ. With $\gamma = 10^{-4}$, we see in Fig. 6 that the true behavior of the system, as captured by both the PLA and the SSA, differs markedly from that predicted deterministically. Rather than

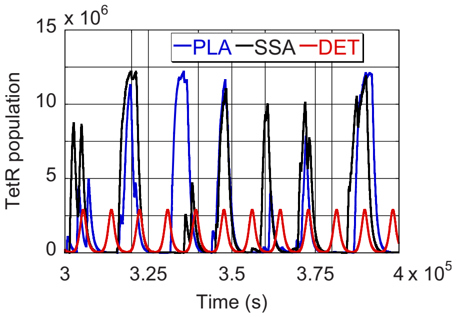

FIG. 6. (Color online) Example time courses (TetR protein) illustrating "deviant effects" [6] in the repressilator at small values of γ. With $\gamma = 10^{-4}$ and $\eta = \rho = 1000$, stochastic realizations (PLA and SSA) differ markedly from the deterministic prediction.

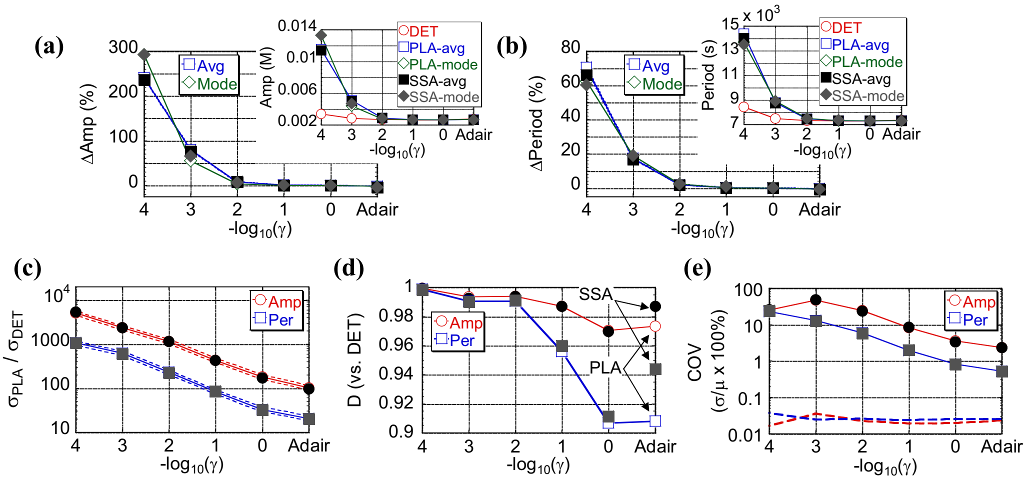

FIG. 7. (Color online) Statistical results for the repressilator. At various values of the telegraph factor γ and at the Adair limit ($\gamma \rightarrow \infty$), results of both PLA and SSA simulations are compared to deterministic (DET) predictions. As in Fig. 5, PLA (and deterministic) results are shown as empty symbols connected by lines, SSA results are shown as disconnected, filled symbols, and PLA and SSA points designated with the same symbol correspond to the same quantity. All PLA and deterministic values are based on over 10 000 collected data points while all SSA values are based on ~1000 collected data points. Note that the only clearly discernible discrepancies between the PLA and SSA results are the histogram distances in (d) at the Adair limit. (a) and (b) Averages and modes for the TetR-protein peak amplitude and period, respectively. In the main plots, results are given as percent deviations from determinism (95% confidence intervals on the PLA averages are difficult to see). In the insets, results are shown in absolute form, illustrating the dependencies of the amplitude and period on γ. (c) Ratios of standard deviations ({PLA or SSA}÷DET) for the TetR-protein peak amplitude and period. Dashed lines denote 80% confidence intervals. (d) Histogram distances ({PLA or SSA} vs DET). Note that the self-distances are off the chart. (e) Coefficients of variation, given as percentages, obtained from both PLA and SSA simulations. In principle, the deterministic limits (dashed lines) vary with γ [see (a) and (b), *insets*], though here they are very nearly constant.

emitting smooth and regular oscillations, the system produces large-amplitude intermittent "bursts" of (mRNA and) protein production. This is a direct consequence of the slow stochastic switching between the ON and OFF states of the genes and is consistent with gene-expression behavior often observed in eukaryotes [4,12]. Note that due to stochasticity the PLA and SSA traces differ from each other. As we shall see, however, they are virtually identical from a statistical standpoint.

In Fig. 7, we present results of our statistical analyses of the repressilator. At various values of γ, as well as at the Adair limit ($\gamma \rightarrow \infty$), we compare the stochastic behavior of the system, as captured by both the PLA and the SSA, to deterministic predictions. In Figs. 7(a) and 7(b), we consider averages and modes for the TetR-protein peak amplitude and period, respectively. In both cases, the PLA and SSA results coincide almost perfectly and show large deviations from determinism at small values of γ and a convergence toward the deterministic limit with increasing γ. Close inspection of the PLA results reveals that statistical convergence to the deterministic limit is achieved for both attributes by $\gamma=1$. It is also evident from these plots that the behavior of the full model (Table II) approaches that of the reduced model (Table III) with increasing γ, as we would expect.

In Figs. 7(c) and 7(d), we consider the distributions of the amplitude and the period. Again, we look at ratios of standard deviations and histogram distances and again we see a convergence toward determinism with increasing γ. However, in this case the deterministic limit is never reached; even at the Adair limit we see considerable deviation from determinism. Furthermore, we see very good correspondence between the PLA and the SSA results. In fact, the only significant differences that we see are the small discrepancies in the histogram distances at the Adair limit in Fig. 7(d). This is interesting in light of the discrepancies seen between the PLA and the SSA in Figs. 5(a) and 5(b) for the calcium-oscillations model, which also contains reduced reaction types (see Table I). This suggests that reduced reactions might be the source of the various inaccuracies seen in Figs. 5 and 7. We plan to investigate this issue further in the future.

In Fig. 7(e), we consider the noise strength through the COV. Here, as in Figs. 7(c) and 7(d), we see almost perfect agreement between the PLA and the SSA results and an incomplete convergence toward the deterministic limit with increasing γ. It is clear, therefore, that significant shot-noise effects persist in this system even as $\gamma \rightarrow \infty$. Moreover, it is interesting to note the elevated levels of noise in the amplitude as compared to the period. We see an approximately order-of-magnitude difference in the COVs for these two attributes at all values of $\gamma > 10^{-4}$ and at the Adair limit. Contrast this with Fig. 5(f), which shows no appreciable difference between the COVs for the amplitude and the period in the calcium-oscillations model. This is an example of the type of fine-level insight that we can garner via the leaping algorithm.

It is clear from Figs. 7(c)–7(e) that the repressilator never behaves in a fully deterministic manner under the conditions that we consider. However, it is also clear that the behavior does approach that of the reduced model with increasing γ.

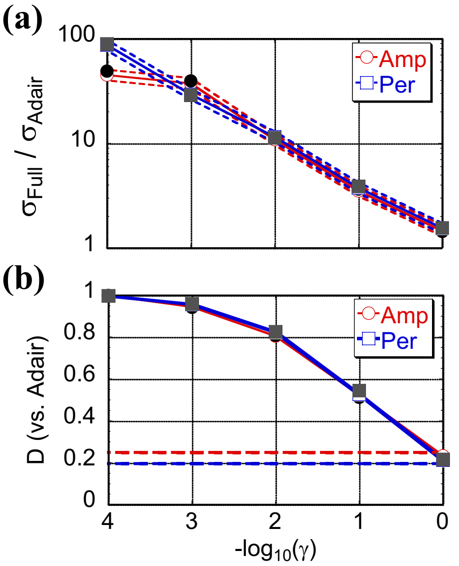

FIG. 8. (Color online) Convergence of the full repressilator model to the Adair limit with increasing γ. At each value of γ, PLA and SSA results of the full model (Table II) are compared to PLA and SSA results, respectively, of the reduced model (Table III). In all cases, the PLA and SSA values (empty and filled symbols, respectively) coincide almost perfectly. (a) Ratios of standard deviations (full÷reduced) for the TetR-protein peak amplitude and period. Dashed lines denote 80% confidence intervals. (b) Histogram distances (full vs reduced). Dashed lines denote twice the *Adair* self-distances.

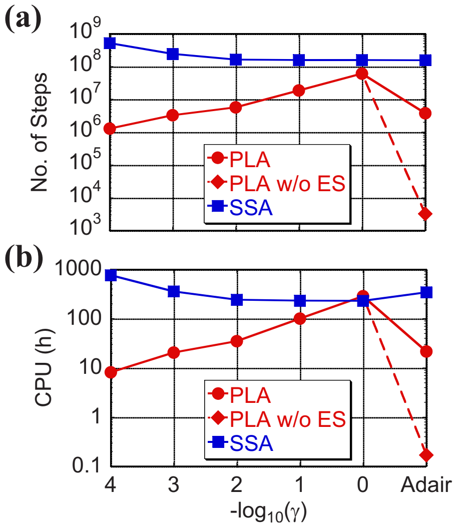

FIG. 9. (Color online) (a) Average numbers of steps from and (b) total CPU times for 1000 PLA and SSA simulation runs of 30 000 s of the full (Table II) and reduced (Table III) repressilator models. All SSA points are based on 100 simulation runs (due to computational expense). Note that the CPU curves in (b) cross at $\gamma = 1$ because each PLA step is more expensive than each SSA step. At the Adair limit, results are given for PLA simulations that both include and exclude the ES classification (see [80]). All simulations were performed on a 3.60 GHz Pentium Xeon processor.

Therefore, in Fig. 8 we quantify this convergence to the Adair limit by repeating the statistical tests of Figs. 7(c) and 7(d) but using the PLA and SSA results for the reduced model, rather than the deterministic results at each γ, as our reference. The results clearly confirm the (near) convergence of the system behavior to the Adair limit at $\gamma = 1$.

Finally, in Fig. 9 we present results of a step and timing analysis comparing the performance of the PLA to the SSA for simulations of both the full (Table II) and reduced (Table III) repressilator models. For the full model, we see the convergence in computational expense of the PLA and the SSA that we anticipated [45]. In Fig. 9(a), the numbers of steps required for PLA and SSA simulations converge asymptotically with increasing γ. In Fig. 9(b), we see a similar trend for the CPU times, although interestingly the curves here cross at $\gamma = 1$ because each PLA step is more computationally expensive than each SSA step. Also worth noticing is that both plots indicate that the expense of the SSA *decreases* with increasing γ while the opposite is true for the PLA. This is because the protein (and mRNA) populations, which are the prime bottleneck for the SSA, tend to be larger at small values of γ [cf. Figs. 6 and 7(a)]. Leaping algorithms are not affected by population sizes, however, having been developed specifically to cope with this problem [23,55]. Hence, we see that when stochastic effects in this system are most pronounced (small γ) the PLA far outperforms the SSA.

In Ref. [45], it was posited that large disparities in rate constants would prove to be the prime bottleneck for leaping algorithms. This is confirmed in Fig. 9 by the declining performance of the PLA with increasing γ. It is for exactly this reason that we consider the reduced model of Table III. In Figs. 7 and 8, we have seen that the behavior of the full model approaches that of the reduced model with increasing γ. Now, in Fig. 9 we see that the performance of the PLA is greatly enhanced by the model reduction. Depending on exactly how we choose to implement the PLA (see [80] for an explanation), we can achieve gains of between 1 and 4 orders of magnitude in both simulation steps and run times. Additionally, it is important to note that reducing the model has very little effect on the performance of the SSA. In fact, we see in Fig. 9 that while the numbers of simulation steps required for the SSA remain virtually unchanged upon reducing the model, the CPU time actually *increases* by ~50% because of the higher complexity rate expressions in Table III which impose additional computational burdens on the algorithm. Our results indicate, therefore, that there is a distinct advantage to using model reduction in conjunction with leaping which is absent with regards to the SSA.

V. DISCUSSION AND CONCLUSIONS

Using the scarcity of published practical applications of leaping as a backdrop, our aim in this paper was to investigate the performance characteristics of a particular leaping algorithm, the PLA, when applied to two nontrivial biological models under a variety of conditions. Our hope was to identify the conditions under which leaping proves particularly beneficial and where it falters and, hence, provide a kind of guide that will aid and motivate future applications of the method. Our use of the PLA, as opposed to a different leaping algorithm, was based purely on expediency, given that we developed the method [45]. However, the PLA operates on the same basic principles as all leaping algorithms and its performance can thus be seen as generally reflective of the entire class of method. In other words, the accelerations that we have reported here are not wholly unique to the PLA but are characteristic of the leaping methodology in general. Similarly, the bottlenecking that we experienced in the face of fast-reversible reactions in the repressilator system can be expected to afflict all leaping algorithms.

That being said, there are attributes of the PLA that we believe set it apart from its various counterparts and we would be remiss in not emphasizing these. Foremost among these is its simplicity of implementation and ease of use. The algorithm is concise, straightforward, and overcomes various technical difficulties (e.g., negative populations [31–33]) without the need for extensive auxiliary machinery [45]. Using the PLA requires little more than a system definition (reactions), rate expressions (elementary or nonelementary), and definition of three simple model-independent parameters [“≈ 1,” “$\gg 1$,” “$\ll 1$” (i.e., ϵ)] [45]. Also significant is the ability to force the algorithm to perform both deterministic and exact-stochastic simulations by simple manipulation of the classification parameters (e.g., setting “≈ 1”$=\infty$ or “$\gg 1$”$=0$). In our case, this significantly simplified the noise quantification and step and timing analyses.

Tangibly speaking, our results clearly illustrate the great potential that leaping methods hold in computational systems biology. For both the calcium-oscillations model and the repressilator, we observed orders-of-magnitude accelerations relative to the SSA (Figs. 4 and 9) that made quantifying stochastic effects in these systems possible. In the calcium-oscillations case, this gave us access to subtle effects of stochasticity that would have been indiscernible otherwise (Fig. 5). For the repressilator, we actually saw the greatest gains in situations where stochastic effects were *most* prevalent (small γ: Fig. 7). This is a particularly intriguing result. Gene regulation is a common feature of many biological models and our results indicate a great potential advantage to using leaping in cases of slow transcription-factor binding and unbinding (such as observed in Ref. [12]).

A critical aspect of the present study was our ability to identify conditions under which the leaping algorithm did *not* perform particularly well. In many ways, this may be more valuable in terms of advancing the use of leaping methods than is highlighting its strengths. The leaping algorithm clearly falters when applied to the full repressilator model (Table II) with large telegraph factor γ (Fig. 9). Intuitively, it is easy to understand why this is. The basic strategy underlying all leaping algorithms is to allow, at each simulation step, as many reaction firings as possible without the reaction rates in the system changing “appreciably” [21,23,55]. However, in this case there is only a single copy of each gene. Thus, only a single binding or unbinding event is possible at each simulation step because one firing changes the binding and unbinding rates from either finite values to zero or vice versa, which is obviously appreciable. When γ is small, this is not a problem because the time interval between successive binding and unbinding events is large enough so that many transcription, translation, and degradation reactions can fire. When γ is large, however, this is no longer the case. The numbers of reaction firings become limited due to the high frequency of binding and unbinding and in the extreme limit the effect is such that the performance of the algorithm approaches that of the SSA (i.e., one reaction firing per step, Fig. 9). We can generalize this observation by saying that small reaction subnetworks (pairs of reversible reactions in this case) that have small populations and large rate constants are prime bottlenecks for leaping algorithms.

Fortunately, our results also illustrate how one can surmount such problems. By applying a simple rapid-equilibrium assumption to the first four reactions of Table II, we were able to recover the behavior of the full model for $\gamma \geq 1$ (Fig. 8) at significantly reduced computational cost (Fig. 9) [80]. This includes accurately capturing stochastic effects associated with finite numbers of mRNAs and proteins. Interestingly, we also showed that reducing the model has little effect on the performance of the SSA (Fig. 9). Thus, the chief benefit to using model reduction in this case was *not* in reducing the number of reactions that had to be considered, but rather in increasing the size of the time step that could be traversed at each simulation step. This is a different perspective on the issue than is usual and strongly suggests that leaping and model reduction should be viewed, not as alternative approaches to the problem of time scale separation (as is common), but as *complementary*. Integrating leaping with advanced model-reduction schemes (e.g., [81–88]) is thus an area of great future interest. As a final note, we did observe some (small) disagreements between the PLA and the SSA results [Figs. 5(a), 5(b), and 7(d)] which may be due to the inclusion of the reduced reaction types. This is an issue that will be investigated further in the future.

So, given the effectiveness of the leaping algorithm as demonstrated in this paper, why are practical applications of leaping so scarce in the literature? The answer is likely multifaceted. First, the approach is relatively new and many researchers may simply be unaware, or only vaguely aware, of its existence. Second, newer incarnations of the method are becoming increasingly complex to the point that even if aware of their existence, a nonexpert may be unable to implement them. Third, it has been our experience that there is a common misperception that stochastic simulation algorithms can only be applied to sets of *elementary* reaction types. Indeed, it is common practice when investigating the stochastic characteristics of an established biochemical model to first “deconstruct” it into elementary reaction steps. While this is not, in fact, strictly necessary, it is possible that attempts to use leaping algorithms in this way have befallen the bottleneck of fast-reversible reactions illustrated in Fig. 9.

It is our hope that this paper alleviates, to some extent, each of these hindrances to the expanded use of leaping algorithms in computational systems biology. For we believe that the future of these methods is bright. Leaping methods represent a small but important piece of the larger puzzle that is comprehensible and actionable models of complex biochemical processes. Coupled with advanced model-reduction techniques that address the problem of rate constant disparities, they can provide a sound and practical means by which the problem of time scale separation in biological systems can be overcome. Further, imbedded into larger modeling and simulation frameworks that include methods for addressing combinatorial complexity [89], spatial localization [90], and parameter uncertainty [91,92], the promises of *in silico* biology [93] might finally be within reach.

ACKNOWLEDGMENTS

We thank H. Lee, J. E. Goose, K. A. Iyengar, F. P. Casey, and J. P. Sethna for useful discussions regarding this work and acknowledge financial support from the Semiconductor Research Corporation. L.A.H. extends special thanks to Professor J. R. Faeder. A.M.P. and E.R.M. further thank the Intel Corporation for funding through Cornell's Learning Initiatives for Future Engineers (LIFE) program.

APPENDIX A: THE ADAIR REDUCTION

For large telegraph factor γ, the gene-protein binding and unbinding reactions,

$$g + r \underset{\gamma k_{-1}}{\overset{\gamma k_1}{\rightleftharpoons}} \{g \cdot r\}, \tag{A1}$$

$$\{g \cdot r\} + r \underset{\gamma k_{-2}}{\overset{\gamma k_2}{\rightleftharpoons}} \{g \cdot r \cdot r\}, \tag{A2}$$

from Table II can be assumed to be in rapid equilibrium. [Here, we use simpler notation for convenience: g for the gene promoter, r for the repressor protein and m for mRNA (below).] Setting the forward and reverse rates of reaction pairs (A1) and (A2) equal to each other, it is easy to show that

$$[g] = K_1 K_2 [g \cdot r \cdot r]/[r]^2, \tag{A3}$$

where $K_i \equiv k_{-i}/k_i$ and $[\cdot]$ denotes concentration (or, more correctly, *occupancy probability*). Assuming that the total number of genes, g_T, is constant,

$$g_T = [g] + [g \cdot r] + [g \cdot r \cdot r], \tag{A4}$$

simple algebra leads to

$$[g] = \frac{K_1 K_2 g_T}{K_1 K_2 + K_2[r] + [r]^2}. \tag{A5}$$

It is then straightforward to show that

$$[g \cdot r] = [g]\frac{[r]}{K_1} = \frac{K_2[r] g_T}{K_1 K_2 + K_2[r] + [r]^2}, \tag{A6}$$

$$[g \cdot r \cdot r] = [g \cdot r]\frac{[r]}{K_2} = \frac{[r]^2 g_T}{K_1 K_2 + K_2[r] + [r]^2}. \tag{A7}$$

The mRNA transcription reactions are

$$g \xrightarrow{k_{\text{tr}}} g + m, \tag{A8}$$

$$\{g \cdot r\} \xrightarrow{k'_{\text{tr}}} \{g \cdot r\} + m, \tag{A9}$$

$$\{g \cdot r \cdot r\} \xrightarrow{k''_{\text{tr}}} \{g \cdot r \cdot r\} + m. \tag{A10}$$

The effective rate expressions for mRNA production given in Table III are obtained by multiplying the rate constants in reactions (A8)–(A10) by the expressions in Eqs. (A5)–(A7). These effective expressions are of the *Adair* form [73].

Notice that Eq. (A7) is similar in form to a Hill equation [73] with a Hill coefficient of +2 (positive cooperativity), except for the $K_2[r]$ term in the denominator. However, this term vanishes as $[r] \to 0$, dominated by $K_1 K_2$, and is overwhelmed by $[r]^2$ as $[r] \to \infty$. Thus, this term can effectively be ignored in all cases other than a small range of intermediate levels of $[r]$. This can be seen, therefore, as a type of "derivation" of the Hill equation. Similarly, Eq. (A5) can be rewritten as

$$[g] = \frac{[r]^{-2} g_T}{[r]^{-2} + (\sqrt{K_1[r]})^{-2} + (\sqrt{K_1 K_2})^{-2}}, \tag{A11}$$

which is similar to a negative-cooperativity Hill equation with a Hill coefficient of −2, again except for the second term in the denominator. Note that the expression for $[g \cdot r]$ in Eq. (A6) is intermediate between Eqs. (A7) and (A11) and does not have an analogous Hill form.

APPENDIX B: MICHAELIS-MENTEN, ADAIR, AND τ SELECTION

At each step of a PLA simulation, a time step τ is calculated based on the current state of the system. This time step corresponds to the time interval over which we expect the reaction rates a_μ ("propensities" in the stochastic jargon) for all reactions in a system to remain *essentially* constant [23,28,34,45]. We quantify the concept of "essentially constant" by imposing a constraint on the relative change of each reaction propensity [34],

$$|a_\mu(t+\tau) - a_\mu(t)|/a_\mu(t) = \epsilon \quad (0 < \epsilon \ll 1). \tag{B1}$$

In practice, there are two approaches for determining τ. The first is a "reaction-based" approach in which the constraint in Eq. (B1) is used directly [23,28,34]. The second, which is used in this paper, is a "species-based" approach where the relative changes in each reactant population X_i are constrained such that Eq. (B1) is satisfied for all reactions [34],

$$|X_i(t+\tau) - X_i(t)|/X_i(t) = \epsilon/g_i. \tag{B2}$$

Here, ϵ is the same as in Eq. (B1) and g_i depends on the highest-order reaction species S_i is involved in.

Procedures for determining g_i which account for elementary reaction types up to third order are given in Cao *et al.* [34] and, in modified form, in Harris and Clancy [45]. Basically, each reaction type has associated with it a value of g_i for each reactant species S_i. Thus, one merely sifts through all reactions in which S_i appears as a reactant and sets g_i equal to the largest of these values. This need be done only once, at the outset of a simulation. In this paper, we consider nonelementary reactions of the Michaelis-Menten (MM) and Adair types (Tables I and III) and must, therefore, derive appropriate g_i expressions for them.

In Table I, we consider two different types of Michaelis-Menten reactions, which we can term first-order (e.g., PLC$^* \to \varnothing$) and second-order (e.g., G_α+PLC$^* \to$PLC*) MM types. It is easy to show that g_i for each of these is the same as for the corresponding elementary reaction [34,45], i.e., $g_i=1$ for first-order MM and $g_i=2$ for second-order MM. For first-order MM reactions we have

$$a_\mu = \left(\frac{c_{\rm cat}}{X_i + C_M}\right) X_i,$$

$$\Delta a_\mu = \frac{da_\mu}{dX_i}\Delta X_i = a_\mu\left(\frac{\Delta X_i}{X_i} - \frac{\Delta X_i}{X_i + C_M}\right),$$

$$\frac{|\Delta a_\mu|}{a_\mu} = \left(\frac{C_M}{X_i + C_M}\right)\frac{|\Delta X_i|}{X_i} \le \frac{|\Delta X_i|}{X_i}. \tag{B3}$$

Equation (B3) shows that if we constrain $|\Delta X_i|/X_i=\epsilon$, then $|\Delta a_\mu|/a_\mu \le \epsilon$, i.e., $g_i=1$ will suffice. Similarly, for second-order MM reactions,

$$a_\mu = \left(\frac{c_{\rm cat}}{X_i + C_M}\right) X_i X_j,$$

$$\Delta a_\mu \approx \frac{\partial a_\mu}{\partial X_i}\Delta X_i + \frac{\partial a_\mu}{\partial X_j}\Delta X_j,$$

$$\frac{|\Delta a_\mu|}{a_\mu} \lesssim \left(\frac{C_M}{X_i + C_M}\right)\frac{|\Delta X_i|}{X_i} + \frac{|\Delta X_j|}{X_j}. \tag{B4}$$

Thus, if we constrain both $|\Delta X_i|/X_i=|\Delta X_j|/X_j=\epsilon/2$, then $|\Delta a_\mu|/a_\mu \le \epsilon$, i.e., $g_i=g_j=2$.

The Adair reactions in Table III represent effective rates of mRNA production from the unbound, singly-bound, and doubly-bound genes. In general terms, we can think of the rates of mRNA production from an n-bound gene with m binding sites ($0 \le n \le m$). We then have

$$a_\mu^{n,m}(r) = \frac{r^n g_T c_{\rm cat}\prod_{i=n}^{m-1} C_i}{\sum_{i=0}^{m}\left(r^i \prod_{j=i}^{m-1} C_j\right)}, \tag{B5}$$

where r is the repressor protein population, g_T is the total number of genes (unity in this case), and $C_i=K_i N_A \Omega$. It is easy to show that for $m=2$ and $n=\{0,1,2\}$, Eq. (B5) reduces to the expressions given in Table III.

Following the same procedure as above, we get

$$\frac{|\Delta a_\mu^{n,m}|}{a_\mu^{n,m}} = \left|\frac{da_\mu^{n,m}}{dr}\frac{\Delta r}{a_\mu^{n,m}}\right| = \left| n - \frac{m\sum_{i=1}^{m}\left(\frac{i}{m} r^i \prod_{j=i}^{m-1} C_j\right)}{\prod_{j=0}^{m-1} C_j + \sum_{i=1}^{m}\left(r^i \prod_{j=i}^{m-1} C_j\right)}\right| \frac{|\Delta r|}{r}. \tag{B6}$$

From Eq. (B6), we see that as $r \to 0$, $\frac{|\Delta a_\mu^{n,m}|}{a_\mu^{n,m}} \to n\frac{|\Delta r|}{r}$, and as $r \to \infty$, $\frac{|\Delta a_\mu^{n,m}|}{a_\mu^{n,m}} \to |n-m|\frac{|\Delta r|}{r}$. This means that if we constrain $|\Delta r|/r=\epsilon/\max\{n,|n-m|\}$, then $|\Delta a_\mu|/a_\mu \le \epsilon$ in all cases, i.e., $g_i=\max\{n,|n-m|\}$. For the three Adair reactions in Table III, this gives $g_i=\{2,1,2\}$ for $n=\{0,1,2\}$, respectively.

[1] H. H. McAdams and A. Arkin, Trends Genet. **15**, 65 (1999).

[2] C. V. Rao, D. M. Wolf, and A. P. Arkin, Nature (London) **420**, 231 (2002).

[3] J. M. Raser and E. K. O'Shea, Science **309**, 2010 (2005).

[4] M. Kærn, T. C. Elston, W. J. Blake, and J. J. Collins, Nat. Rev. Genet. **6**, 451 (2005).

[5] B. Di Ventura, C. Lemerle, K. Michalodimitrakis, and L. Serrano, Nature (London) **443**, 527 (2006).

[6] M. S. Samoilov and A. P. Arkin, Nat. Biotechnol. **24**, 1235 (2006).

[7] M. S. Samoilov, G. Price, and A. P. Arkin, Sci. STKE **2006**, re17 (2006).

[8] N. Maheshri and E. K. O'Shea, Annu. Rev. Biophys. Biomol. Struct. **36**, 413 (2007).

[9] J. L. Spudich and D. E. Koshland, Jr., Nature (London) **262**, 467 (1976).

[10] M. B. Elowitz, A. J. Levine, E. D. Siggia, and P. S. Swain, Science **297**, 1183 (2002).

[11] N. Fedoroff and W. Fontana, Science **297**, 1129 (2002).

[12] W. J. Blake, G. Balázsi, M. A. Kohanski, F. J. Isaacs, K. F. Murphy, Y. Kuang, C. R. Cantor, D. R. Walt, and J. J. Collins, Mol. Cell **24**, 853 (2006).

[13] M. B. Elowitz and S. Leibler, Nature (London) **403**, 335 (2000).

[14] N. Barkai and S. Leibler, Nature (London) **403**, 267 (2000).

[15] J. M. G. Vilar, H. Y. Kueh, N. Barkai, and S. Leibler, Proc. Natl. Acad. Sci. U.S.A. **99**, 5988 (2002).

[16] D. Gonze, J. Halloy, and A. Goldbeter, Proc. Natl. Acad. Sci. U.S.A. **99**, 673 (2002).

[17] B. B. Aldridge, J. M. Burke, D. A. Lauffenburger, and P. K. Sorger, Nat. Cell Biol. **8**, 1195 (2006).

[18] C. J. Tomlin and J. D. Axelrod, Nat. Rev. Genet. **8**, 331 (2007).

[19] D. T. Gillespie, J. Comput. Phys. **22**, 403 (1976).

[20] D. T. Gillespie, J. Phys. Chem. **81**, 2340 (1977).

[21] D. T. Gillespie, Annu. Rev. Phys. Chem. **58**, 35 (2007).

[22] D. Endy and R. Brent, Nature (London) **409**, 391 (2001).

[23] D. T. Gillespie, J. Chem. Phys. **115**, 1716 (2001).

[24] M. A. Gibson and J. Bruck, J. Phys. Chem. A **104**, 1876 (2000).

[25] H. Resat, H. S. Wiley, and D. A. Dixon, J. Phys. Chem. B **105**, 11026 (2001).

[26] Y. Cao, H. Li, and L. Petzold, J. Chem. Phys. **121**, 4059 (2004).

[27] J. M. McCollum, G. D. Peterson, C. D. Cox, M. L. Simpson, and N. F. Samatova, Comput. Biol. Chem. **30**, 39 (2006).

[28] D. T. Gillespie and L. R. Petzold, J. Chem. Phys. **119**, 8229 (2003).

[29] M. Rathinam, L. R. Petzold, Y. Cao, and D. T. Gillespie, J. Chem. Phys. **119**, 12784 (2003).

[30] Y. Cao, L. R. Petzold, M. Rathinam, and D. T. Gillespie, J. Chem. Phys. **121**, 12169 (2004).

[31] T. Tian and K. Burrage, J. Chem. Phys. **121**, 10356 (2004).

[32] A. Chatterjee, D. G. Vlachos, and M. A. Katsoulakis, J. Chem. Phys. **122**, 024112 (2005).

[33] Y. Cao, D. T. Gillespie, and L. R. Petzold, J. Chem. Phys. **123**, 054104 (2005).

[34] Y. Cao, D. T. Gillespie, and L. R. Petzold, J. Chem. Phys. **124**, 044109 (2006).

[35] H. Wagner, M. Möller, and K. Prank, J. Chem. Phys. **125**, 174104 (2006).

[36] A. Auger, P. Chatelain, and P. Koumoutsakos, J. Chem. Phys. **125**, 084103 (2006).

[37] X. Cai and Z. Xu, J. Chem. Phys. **126**, 074102 (2007).

[38] M. F. Pettigrew and H. Resat, J. Chem. Phys. **126**, 084101 (2007).

[39] X. Peng, W. Zhou, and Y. Wang, J. Chem. Phys. **126**, 224109 (2007).

[40] Y. Cao, D. T. Gillespie, and L. R. Petzold, J. Chem. Phys. **126**, 224101 (2007).

[41] M. Rathinam and H. El Samad, J. Comput. Phys. **224**, 897 (2007).

[42] D. F. Anderson, J. Chem. Phys. **128**, 054103 (2008).

[43] Z. Xu and X. Cai, J. Chem. Phys. **128**, 154112 (2008).

[44] A. Leier, T. T. Marquez-Lago, and K. Burrage, J. Chem. Phys. **128**, 205107 (2008).

[45] L. A. Harris and P. Clancy, J. Chem. Phys. **125**, 144107 (2006).

[46] E. L. Haseltine and J. B. Rawlings, J. Chem. Phys. **117**, 6959 (2002).

[47] T. R. Kiehl, R. M. Mattheyses, and M. K. Simmons, Bioinformatics **20**, 316 (2004).

[48] K. Takahashi, K. Kaizu, B. Hu, and M. Tomita, Bioinformatics **20**, 538 (2004).

[49] K. Vasudeva and U. S. Bhalla, Bioinformatics **20**, 78 (2004).

[50] K. Burrage, T. Tian, and P. Burrage, Prog. Biophys. Mol. Biol. **85**, 217 (2004).

[51] J. Puchałka and A. M. Kierzek, Biophys. J. **86**, 1357 (2004).

[52] H. Salis and Y. Kaznessis, J. Chem. Phys. **122**, 054103 (2005).

[53] M. Griffith, T. Courtney, J. Peccoud, and W. H. Sanders, Bioinformatics **22**, 2782 (2006).

[54] D. C. Wylie, Y. Hori, A. R. Dinner, and A. K. Chakraborty, J. Phys. Chem. B **110**, 12749 (2006).

[55] D. T. Gillespie, J. Chem. Phys. **113**, 297 (2000).

[56] A. Chatterjee, K. Mayawala, J. S. Edwards, and D. G. Vlachos, Bioinformatics **21**, 2136 (2005).

[57] M. Perc, M. Gosak, and M. Marhl, Chem. Phys. Lett. **437**, 143 (2007).

[58] A. Handel, I. M. Longini, Jr., and R. Antia, PLOS Comput. Biol. **3**, e240 (2007).

[59] U. Kummer, L. F. Olsen, C. J. Dixon, A. K. Green, E. Bornberg-Bauer, and G. Baier, Biophys. J. **79**, 1188 (2000).

[60] M. J. Berridge, M. D. Bootman, and P. Lipp, Nature (London) **395**, 645 (1998).

[61] M. Falcke, Adv. Phys. **53**, 255 (2004).

[62] S. Schuster, M. Marhl, and T. Höfer, Eur. J. Biochem. **269**, 1333 (2002).

[63] U. Kummer, B. Drajnc, J. Pahle, A. K. Green, C. J. Dixon, and M. Marhl, Biophys. J. **89**, 1603 (2005).

[64] C. J. Dixon, N. M. Woods, K. S. R. Cuthbertson, and P. H. Cobbold, Biochem. J. **269**, 499 (1990).

[65] J. Hasty, D. McMillen, F. Isaacs, and J. J. Collins, Nat. Rev. Genet. **2**, 268 (2001).

[66] J. Hasty, D. McMillen, and J. J. Collins, Nature (London) **420**, 224 (2002).

[67] D. Sprinzak and M. B. Elowitz, Nature (London) **438**, 443 (2005).

[68] S. A. Benner and A. M. Sismour, Nat. Rev. Genet. **6**, 533 (2005).

[69] A. Goldbeter, Nature (London) **420**, 238 (2002).

[70] D. Gonze and A. Goldbeter, Chaos **16**, 026110 (2006).

[71] J. P. Sethna, *Statistical Mechanics: Entropy, Order Parameters, and Complexity* (Oxford University Press, Oxford, England, 2006).

[72] J. P. Sethna and C. R. Myers, http://www.physics.cornell.edu/sethna/StatMech/ComputerExercises/Repressilator/Repressilator.html

[73] A. Cornish-Bowden, *Fundamentals of Enzyme Kinetics*, 3rd ed. (Portland, London, 2004).

[74] W. E. Wallace, A. J. Kearsley, and C. M. Guttman, Anal. Chem. **76**, 2446 (2004).

[75] A. J. Kearsley, W. E. Wallace, J. Bernal, and C. M. Guttman, Appl. Math. Lett. **18**, 1412 (2005).

[76] Obviously, deterministic simulations should exhibit zero variance in their results. However, due to sampling and curve-fitting inaccuracies we do see slight variations. It is these variations that we use as the criteria for determining when a system attribute has converged to the deterministic limit. Clearly, if the PLA results show equal or less variation than the deterministic results then we can deem that the property has converged to determinism.

[77] J. S. Milton and J. C. Arnold, *Introduction to Probability and Statistics: Principles and Applications for Engineering and the Computing Sciences*, 3rd ed. (McGraw-Hill, New York, 1995).

[78] Y. Cao and L. Petzold, J. Comput. Phys. **212**, 6 (2006).

[79] The self-distance is a measure of the difference between a sample histogram (i.e., one based on a finite amount of data) and the "true" (unattainable) histogram. Since the measure is based on *absolute* differences [45,78] two sample histograms can have equal self-distances but arising from opposite sources (e.g., one histogram might be slightly taller and thinner, while the other shorter and wider, than the true histogram). This means that two sample histograms can be as dissimilar as *twice* the self-distance and still be considered indistinguishable from the true histogram and hence each other. In the Appendix of Ref. [45], it was incorrectly stated that two histograms can be considered distinct if they differ by only a single self-distance.

[80] We found that significant speed ups can be achieved in the PLA simulations of the reduced repressilator model (Table III) if we remove the ES classification (see Fig. 9). The problem lies in the iterative τ-selection procedure [45] designed to account for the randomness of the ES reactions. In this particular case, we experienced an unexpected "classification cascade," whereby reactions classified as ES led to a reduced τ, which

then led to more ES reactions (via reclassification), which further reduced τ, and so on and so forth. Removing the ES classification eliminated this problem with no major effect on the accuracy. However, this cannot be done in all cases. Removing the ES classification when simulating the full model led to numerous instances of negative populations, specifically for the species g_x, $\{g_x \cdot p_r\}$, and $\{g_x \cdot p_r \cdot p_r\}$, which can only have populations of zero or unity. These required costly reversals that significantly increased the run time. Further investigation of this issue is warranted and will be undertaken in the near future. Also note that all results reported in Figs. 7 and 8 were performed *with* the ES classification included.

[81] T. Shibata, J. Chem. Phys. **119**, 6629 (2003).

[82] R. Bundschuh, F. Hayot, and C. Jayaprakash, Biophys. J. **84**, 1606 (2003).

[83] Y. Cao, D. T. Gillespie, and L. R. Petzold, J. Chem. Phys. **122**, 014116 (2005).

[84] J. Goutsias, J. Chem. Phys. **122**, 184102 (2005).

[85] A. Samant and D. G. Vlachos, J. Chem. Phys. **123**, 144114 (2005).

[86] W. E, D. Liu, and E. Vanden-Eijnden, J. Chem. Phys. **123**, 194107 (2005).

[87] H. Salis and Y. N. Kaznessis, J. Chem. Phys. **123**, 214106 (2005).

[88] M. J. Morelli, R. J. Allen, S. Tănase-Nicola, and P. R. ten Wolde, J. Chem. Phys. **128**, 045105 (2008).

[89] W. S. Hlavacek, J. R. Faeder, M. L. Blinov, R. G. Posner, M. Hucka, and W. Fontana, Sci. STKE **2006**, re6 (2006).

[90] C. Lemerle, B. Di Ventura, and L. Serrano, FEBS Lett. **579**, 1789 (2005).

[91] K. S. Brown and J. P. Sethna, Phys. Rev. E **68**, 021904 (2003).

[92] R. Gunawan, Y. Cao, L. Petzold, and F. J. Doyle III, Biophys. J. **88**, 2530 (2005).

[93] B. Palsson, Nat. Biotechnol. **18**, 1147 (2000).