Stochastic operator-splitting method for reaction-diffusion systems

TaiJung Choi,[1,a),b)] Mano Ram Maurya,[2,a),c)] Daniel M. Tartakovsky,[1,d)] and Shankar Subramaniam[2,3,e)]

[1]*Department of Mechanical and Aerospace Engineering, University of California, San Diego, 9500 Gilman Drive, La Jolla, California 92093-0411, USA*

[2]*Department of Bioengineering, University of California, San Diego, 9500 Gilman Drive, La Jolla, California 92093-0412, USA*

[3]*Department of Cellular and Molecular Medicine, Department of Chemistry and Biochemistry and Graduate Program in Bioinformatics, University of California, San Diego, La Jolla, California 92093, USA*

(Received 17 August 2012; accepted 12 October 2012; published online 8 November 2012)

Many biochemical processes at the sub-cellular level involve a small number of molecules. The local numbers of these molecules vary in space and time, and exhibit random fluctuations that can only be captured with stochastic simulations. We present a novel stochastic operator-splitting algorithm to model such reaction-diffusion phenomena. The reaction and diffusion steps employ stochastic simulation algorithms and Brownian dynamics, respectively. Through theoretical analysis, we have developed an algorithm to identify if the system is reaction-controlled, diffusion-controlled or is in an intermediate regime. The time-step size is chosen accordingly at each step of the simulation. We have used three examples to demonstrate the accuracy and robustness of the proposed algorithm. The first example deals with diffusion of two chemical species undergoing an irreversible bimolecular reaction. It is used to validate our algorithm by comparing its results with the solution obtained from a corresponding deterministic partial differential equation at low and high number of molecules. In this example, we also compare the results from our method to those obtained using a Gillespie multi-particle (GMP) method. The second example, which models simplified RNA synthesis, is used to study the performance of our algorithm in reaction- and diffusion-controlled regimes and to investigate the effects of local inhomogeneity. The third example models reaction-diffusion of CheY molecules through the cytoplasm of *Escherichia coli* during chemotaxis. It is used to compare the algorithm's performance against the GMP method. Our analysis demonstrates that the proposed algorithm enables accurate simulation of the kinetics of complex and spatially heterogeneous systems. It is also computationally more efficient than commonly used alternatives, such as the GMP method.
[http://dx.doi.org/10.1063/1.4764108]

I. INTRODUCTION

Randomness plays an important role in the behavior of many biological phenomena, such as cellular signaling and gene regulatory networks.[1–3] While deterministic ordinary differential equations (ODEs) often provide accurate predictions of the dynamics of biochemical pathways with large numbers of reacting molecules, they fail when the concentrations of reactants and/or their products become small and the law of mass action becomes invalid. When this occurs, the randomness associated with the dynamics of individual molecules becomes pronounced, necessitating the use of stochastic simulations. Standard stochastic techniques, e.g., Gillespie's stochastic simulation algorithm[4] and its computationally efficient modifications,[5,6] are routinely used to model biochemical reactions in such systems. Such algorithms assume that reactants and their products are well mixed, i.e., distributed uniformly in space.

The latter assumption is problematic when the number of molecules is small. This is especially so in crowded environments with complex internal geometry, wherein stochasticity and spatial variability are inseparable. Partial differential equations (PDEs) provide accurate macroscopic predictions of the dynamics of spatially heterogeneous systems with large numbers of molecules. Yet, similar to ODE-based models, they fail to account for the randomness inherent in a system comprised of small numbers of molecules. It is essential that computational methods for reaction-diffusion systems with small numbers of molecules are capable of handling both stochasticity and heterogeneity.

A number of micro- and meso-scale methods have been developed for the simulation of reaction-diffusion systems. The micro-scale approaches, e.g., the Green's function reaction dynamics[7] and Smoldyn's algorithm,[8] are based on Brownian dynamics and require the reacting molecules to diffuse within a certain distance from each other in order for bimolecular reactions to take place. The latter requirement necessitates the use of a numerical mesh and the tracking of individual particles and/or distances between them, which renders such algorithms computationally expensive.

[a)] TaiJung Choi and Mano Ram Maurya contributed equally to this work.
[b)] Electronic mail: tjchoi@ucsd.edu.
[c)] Electronic mail: mano@sdsc.edu.
[d)] Author to whom correspondence should be addressed. Electronic mail: dmt@ucsd.edu. Phone: (858) 534-1375. Fax: (858) 534-7599.
[e)] Author to whom correspondence should be addressed. Electronic mail: shankar@ucsd.edu. Phone: (858) 822-0986. Fax: (858) 822-5722.

Mesoscopic approaches, e.g., MesoRD[9] and the Gillespie multi-particle (GMP) method,[10,11] trade representational accuracy for computational efficiency. They are based on a reaction-diffusion master equation,[12] which generalizes a chemical master equation developed for well-mixed chemical reactions by discretizing the space into a collection of cells and treating each cell as a well-mixed system. MesoRD[9] treats diffusion as a unimolecular reaction whose reaction rate is related to the corresponding diffusion coefficient. The GMP method[10,11] employs an operator-splitting scheme in which the Gillespie algorithm and cellular automata[13] handle reactions and diffusion, respectively.

We present a numerical algorithm to simulate stochastic reaction-diffusion processes with a small number of non-uniformly distributed molecules. It employs an operator-splitting, in which the Gillespie algorithm (or its accelerated versions) and Brownian dynamics (or the Smoluchowski equation) are used to simulate reactions and diffusion, respectively. Our algorithm is conceptually similar to the GMP method in that it relies on operator-splitting. However, it offers a number of computational advantages in terms of both accuracy and efficiency. First, the cellular automata used in the GMP method restrict a particle's movement during one fixed time-step to the adjacent cells only; while Brownian motion places no restrictions on the distance particles can travel during one time-step, thus gaining in computational efficiency. Second, Brownian dynamics provides a more accurate representation of diffusion than cellular automata. Third, our algorithm offers the flexibility of an "on-the-fly" adaptive selection of the time-step size for operator-splitting, depending on whether the system is reaction- or diffusion-controlled. The outline of this paper is as follows.

Our stochastic operator-splitting approach is described in Sec. II. This section contains a brief description of the stochastic simulation algorithm for modeling reactions and a comparative analysis of the two approaches—Brownian motion and cellular automata—to deal with diffusion. It also contrasts our operator-splitting algorithm with that used in the GMP method (Sec. II B). Section III presents three computational examples, which demonstrate the accuracy and robustness of the proposed algorithm. The first example (Sec. III A) considers diffusion of two chemical species undergoing an irreversible bimolecular reaction in order to validate our algorithm and to analyze its performance and accuracy in terms of the time-step and the cell size. This is done by comparing the stochastic simulation results with solutions of the corresponding deterministic PDEs. The detailed comparison elucidates the effects of the finite (small) number of molecules and space-time discretization on the simulation accuracy and efficiency. The second example (Sec. III B) models an idealized gene expression system.[7] It serves to investigate the performance of our algorithm in reaction- and diffusion-controlled regimes and the effects of local inhomogeneity. The third example (Sec. III C) considers reaction and diffusion of CheY molecules through the cytoplasm of *Escherichia coli* during chemotaxis.[14] In addition to its biochemical significance, this example poses additional computational challenges by introducing a specific local structure. In all the three cases, we demonstrate that our algorithm outperforms the GMP method in terms of computational time. In Sec. IV, we summarize the simulation results and provide conclusions.

II. METHODS: NUMERICAL APPROACH

A. Operator-splitting method

We consider M species that undergo diffusion and N (bio)chemical reactions. Spatiotemporal evolution of their concentrations $\{c_i(\mathbf{x}, t)\}_{i=1}^{M}$ can be described by a system of reaction-diffusion equations

$$\frac{\partial c_i}{\partial t} = D_i \nabla^2 c_i + f_i(c_1, \ldots, c_M), \quad i = 1, \ldots, M, \tag{1}$$

where D_i is the molecular diffusion coefficient of the ith species, and f_i is the corresponding net production rate through reactions. Our focus is on reaction-diffusion systems with small numbers of molecules, in which continuum representations such as Eq. (1) are inadequate. Such phenomena are typically handled with stochastic simulations. While stochastic, particle-based methods for modeling both reactions in well-mixed environments (e.g., the Gillespie algorithm[4]) and diffusion of chemically inert molecules (e.g., Brownian dynamics) are relatively mature, the same cannot be said about chemical reactions in spatially heterogeneous (reaction-diffusion) systems.

We propose an operator-splitting method that enables one to take advantage of the considerable advances in modeling chemical reactions and molecular diffusion by treating these two phenomena separately. We use the (modified) Gillespie algorithm and Brownian dynamics to represent the reaction and diffusion steps, respectively, in lieu of their continuum representations in Eq. (1). The relative order of these steps is determined dynamically depending on whether the system is in diffusion- or reaction-controlled state.

This *raison d'être* for employing an operator splitting is different from the use of operator-splitting algorithms to model deterministic reaction-diffusion systems. In the latter case, the goal is to handle the stiffness of the reaction-diffusion equations in which diffusion and reaction processes have different time scales. A typical operator-splitting method for solving deterministic reaction-diffusion equations employs an implicit method to handle the (stiff) reaction simulations and an explicit method to handle diffusion. Examples of deterministic operator-splitting approaches include the Douglas-Gunn alternating direction implicit (ADI) method[15] and the method of lines (MOL).[16] The former applies an explicit Euler scheme to diffusion and an implicit Crank-Nicholson method to reactions. The latter converts partial-differential equations (PDEs) into ordinary differential equations by discretizing the spatial derivatives and leaving the time variable continuous.

We employ an operator-splitting algorithm[17] to approximate Eq. (1) with

$$\frac{\partial c_i'}{\partial t} = D_i \nabla^2 c_i', \tag{2a}$$

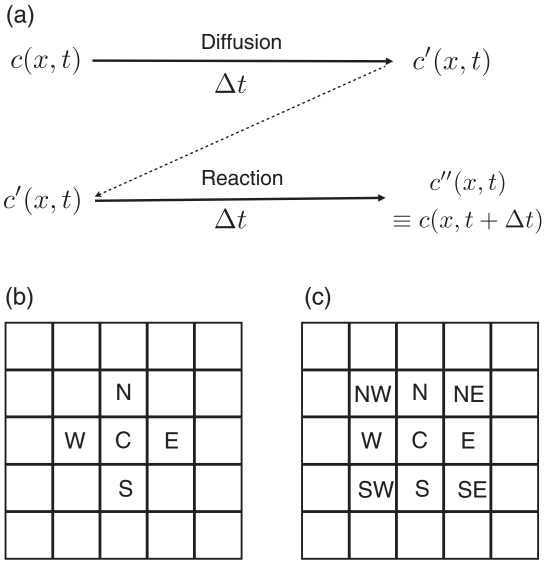

FIG. 1. (a) Schematic representation of the diffusion-reaction operator-splitting. The final value after diffusion process at time $t + \Delta t$ is used as the initial value for the reaction process. Final value of reaction process is the final value at the end of diffusion-reaction process. (b) and (c) Cellular automata neighborhoods in $d = 2$ dimension: in the von Neumann automata the probability of staying in a cell or diffusing to its neighbors is 1/5 (b), in the Moore automata this probability is 1/9 (c).

$$\frac{\partial c_i''}{\partial t} = f_i(c_1'', \ldots, c_M'') \tag{2b}$$

during the time interval [t, $t + \Delta t$]. Here $c_i'(t) = c_i(t)$ and $c_i''(t) = c_i'(t + \Delta t)$, so that the concentration of the ith species at the end of the time-step Δt is $c_i''(t) = c_i(t + \Delta t)$. Fig. 1(a) provides a graphical representation of this operator-splitting algorithm. The resulting stochastic operator-splitting algorithm will enable us to analyze the effects of intrinsic noise in spatially heterogeneous biological systems (Sec. III). Our implementation of the reaction process using Gillespie algorithm, the diffusion process using either Brownian dynamics or cellular automata, and the GMP algorithm is described in the supplementary material.[18] Briefly, in Gillespie algorithm,[4] to advance the system from state $\mathbf{X}(t)$, two random numbers r_1 and r_2 distributed uniformly on the unit interval [0, 1] are generated. Then, a discrete random value j and continuous random value τ are selected probabilistically in accordance with Eq. (S3) (supplementary material[18]) as

$$\tau = \frac{1}{a_{\mathrm{sum}}} \ln\left(\frac{1}{r_1}\right), \quad \sum_{j'=1}^{j-1} a_{j'} \le r_2 a_{\mathrm{sum}} \le \sum_{j'=1}^{j} a_{j'}, \tag{3}$$

where a_{sum} is the sum of all propensity functions. The system state at $t + \tau$ is updated according to $\mathbf{X}(t + \tau) = \mathbf{X}(t) + \boldsymbol{\nu}_j$ where the entries of the vector ν_j are the change in the number of molecules of various species due to the jth reaction.[4]

In Brownian dynamics, a species diffuses from its current location $\mathbf{X}(t) \in \mathbb{R}^3$ at time t to its new location at time ($t + \Delta t$) according to Ref. 19: $\mathbf{X}(t + \Delta t) = \mathbf{X}(t) + \sqrt{2D_i\Delta t}\,\boldsymbol{\xi}$, where $\boldsymbol{\xi} = (\xi_1, \xi_2, \xi_3)^T$ is a normal random displacement vector (supplementary material[18]).

In cellular automata, the ith species can diffuse to one of its neighboring cells (Figs. 1(b) and 1(c)) during the time interval equal to its diffusion-time constant τ_{D_i} given by $\tau_{D_i} = (\Delta x)^2/(2D_i d)$ (supplementary material[18]).

B. Algorithm for the stochastic operator-splitting method

To deal with reaction-diffusion systems composed of a small number of molecules, we propose the following stochastic operator-splitting algorithm.

1. Lattice: The space is discretized into a lattice of cells. Within each cell (lattice element), each species is assumed to be distributed homogeneously.
2. System state: Determine whether the system is at diffusion- or reaction-controlled state to decide the time-step size Δt_j at the jth time-step.
3. Diffusion process: Diffusion of species between cells is modeled via Brownian dynamics with a fixed time-step by treating the space as a continuum.
4. Reaction process: Reactions within each cell are simulated via the Gillespie algorithm or its accelerated versions.
5. Time is increased by the time-step size and the above steps are repeated till the final desired time.

1. Dynamic identification of system's state

A key feature of our algorithm is its ability to determine at each time-step the system's state (reaction- or diffusion-controlled) and to set the time-step size accordingly. For an ith cell ($i = 1, \ldots, C$ where C is the number of cells in a numerical grid) at the jth time-step Δt_j, we define a macroscopic time constant

$$T_{R_{ij}} = \frac{1}{a_{\mathrm{sum}}^{ij}}, \quad a_{\mathrm{sum}}^{ij} \equiv \sum_{k=1}^{N} a_k(\mathbf{X}_{ij}), \tag{4}$$

where $\mathbf{X}_{ij}$ is the state $\mathbf{X}$ of the ith cell at the jth time-step, and $a_k(\mathbf{X}_{ij})$ is the propensity function for the kth reaction. At each time-step, we find the minimum value of the macroscopic time constants over all the cells,

$$T_{R_j}^{\min} \equiv \min_i T_{R_{ij}} \tag{5}$$

and define

$$\tau_{R_j} = T_{R_j}^{\min} \ln\left(\frac{1}{r}\right). \tag{6}$$

Figs. 2(a) and 2(b) show a frequency chart of $\ln(1/r)$ and the corresponding cumulative probability distribution. They reveal that the cumulative probability of $\ln(1/r) \le 1$ is 0.63 (also see Table I), i.e., the probability of $\tau_{R_j} \le T_{R_j}^{\min}$ is 63%. Then, a time fraction

$$F \equiv \frac{T_{R_j}^{\min}}{\tau_D} \tag{7}$$

can be used to classify the system as reaction- or diffusion-controlled as explained below.

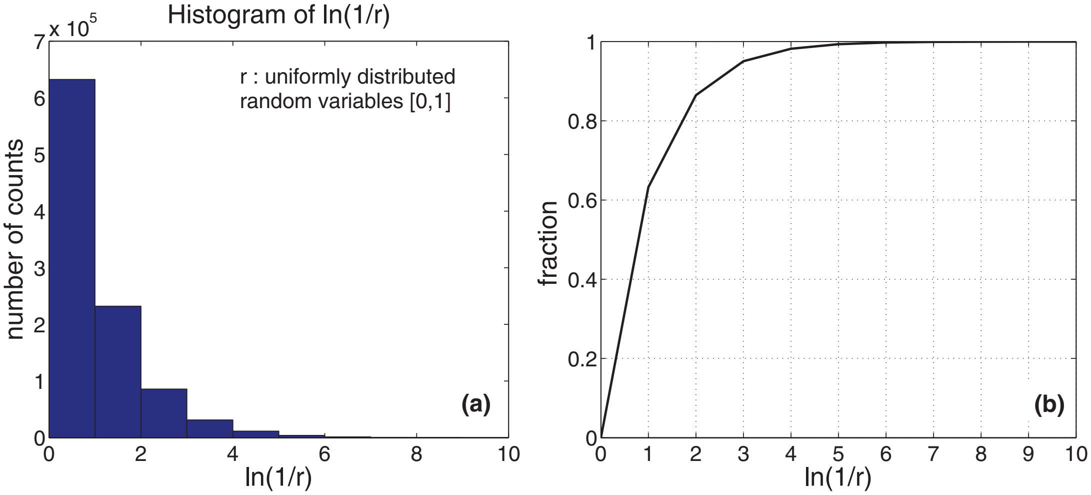

FIG. 2. (a) Histogram of $\ln(1/r)$, where r is a uniformly distributed random variable in [0,1]. (b) Cumulative fraction of counts out of total counts (one million). About 63% of the numbers have values less than 1 and 86% of the numbers are less than 2.

It follows from Eqs. (6) and (7) that

$$\frac{\tau_{R_j}}{\tau_D} = F \ln\left(\frac{1}{r}\right), \tag{8}$$

which allows one to compute the cumulative probability of $\tau_{R_j}/\tau_D \leq 1$ as

$$\mathrm{P}\left[\frac{\tau_{R_j}}{\tau_D} \leq 1\right] = \mathrm{P}\left[F \ln\left(\frac{1}{r}\right) \leq 1\right] = \mathrm{P}\left[\ln\left(\frac{1}{r}\right) \leq \frac{1}{F}\right] = 1 - \mathrm{e}^{-1/F}. \tag{9}$$

This is the same as the waiting time probability in the Gillespie algorithm.[4] It becomes clear that the magnitude of F determines the state of the system. For example, $F = 1$ corresponds to $\mathrm{P}[\ln(1/r) \leq 1] = 0.63$ (Fig. 2(b)), so that $\mathrm{P}[\tau_{R_j}/\tau_D \leq 1] = 0.63$ as well. In other words, $F = 1$ implies that $\tau_{R_j} \leq \tau_D$ in about 63% cases (Table I), i.e., the system is diffusion-controlled. Similarly, $F = 0.5$ (even faster reactions) translates into $\mathrm{P}[\ln(1/r) \leq (1/F) = 2] = 0.86$ (Fig. 2(b)) and $P[\tau_{R_j}/\tau_D \leq 1] = 0.86$. We classify a system as diffusion-controlled, if $P[\tau_{R_j}/\tau_D \leq 1] \geq 0.5$. According to Table I, this corresponds to $F \leq 1/\ln(2) = 1.44$. We introduce a parameter $0 < k_1 \leq 1/\ln(2)$ and say that the system is diffusion-controlled if $F < k_1$. The smaller the value of k_1, the more stringent the criterion becomes. Essentially, as the probability of $\tau_R < \tau_D$ increases, i.e., k_1 decreases, the system becomes more diffusion-controlled. Similarly, we define a related parameter $k_1^{'}$ so that if $F > k_1^{'}$, then the system is reaction-controlled.

TABLE I. The tunable parameter k_1 is used as a criterion to decide if the system is diffusion- or reaction-controlled. As the probability of τ_R being less than τ_D increases, the system becomes more diffusion-controlled. The other parameter, k_2, is related to the probability of a reaction taking place during Δt. As k_2 increases, the probability of a reaction occurrence during Δt increases. In our algorithm, $k_1 = 0.5$, $k_1^{'} = 3$, $k_2 = 2$, and $k_2^{'} = 3$ are used. Please refer to Fig. 2.

F, k_1, or $k_1^{'}$	Relation	Meaning
0.5	$\bar{T}_R = 0.5\tau_D$	86% of τ_R is less than τ_D
1	$\bar{T}_R = \tau_D$	63% of τ_R is less than τ_D
1.44	$\bar{T}_R = 1.44\tau_D$	50% of τ_R is less than τ_D
3	$\bar{T}_R = 3\tau_D$	28% of τ_R is less than τ_D
k_2	Probability for the reaction to occur during Δt	
1	63%	
2	86%	
3	95%	

In diffusion-controlled systems, many reactions may be fired during Δt_j. We set the time-step $\Delta t_j = k_2 \tau_D$, where k_2 is a tunable parameter representative of the cut-off (or critical) value of $\ln(1/r)$ for a desired cumulative probability (Fig. 2(b) and Table I). For example, $k_2 = 2$ corresponds to 0.86 probability of a reaction taking place during Δt_j.

For the reaction-controlled system, τ_{R_j} (or $T_{R_j}^{\min}$) is much larger than τ_D. For example, $k_1^{'} = 3$ corresponds to $\mathrm{P}[\ln(1/r) \leq (1/F) = 1/3] = 0.28$ (Fig. 2(b)), i.e., $P[\tau_{R_j}/\tau_D \leq 1] = 0.28$. To ensure the firing of some reactions, larger Δt_j should be chosen. Based on several simulations, we found that $\Delta t_j = 10\tau_D$ provides good results.

We also define an intermediate regime that is characterized by values of F that prevent one from classifying a system as being diffusion- or reaction-controlled. In this regime, the time-step Δt_j should be chosen between $k_2\tau_D$ and $10\tau_D$. Our numerical experimentation suggests that setting $k_2^{'} = 3$ provides a good balance between accuracy and computational efficiency.

2. Algorithm

A detailed algorithm for the numerical implementation of the above steps of our stochastic operator-splitting method is provided below.

```
1. For a given space dimension $d$ and cell size $\Delta x$, calculate the diffusion time $\tau_{D_i} = (\Delta x)^2/(2D_i d)$ of diffusing species $i = 1, \ldots, M$ and set $\tau_D = \min\{\tau_{D_i}\}$.
2. Initialize $t = 0$.
3. While $t \leq t_{\mathrm{final}}$
```

```
(a) Define whether system is diffusion- or reaction-controlled at every time-step.
    • Calculate $T_{R_j}^{\min}$ through Eq. (5).
    • Calculate $F$ through Eq. (7).
(b) Compute the time-step according to the classification of the system. The multiplicative factors $k_1$, $k_1^{'}$, $k_2$, and $k_2^{'}$ are selected based on Fig. 2 and Table I.
    (i) If $F < k_1$ (diffusion-controlled),
        • Set $\Delta t_j = k_2 \tau_D$.
    (ii) Elseif $k_1 < F < k_1^{'}$ (mixed zone),
        • Set $\Delta t_j = k_2^{'} \tau_D$.
    (iii) Else $F > k_1^{'}$ (reaction-controlled)
        • Set $\Delta t_j = 10 \tau_D$.
(c) Reset $t_{\text{old}} = t$.
(d) Perform the diffusion step first followed by the reaction step.
    (i) Diffusion step: Use Brownian dynamics to advance the species with time-step $\Delta t_j$.
    (ii) For each cell: Reaction step:
        A. While $(t - t_{\text{old}}) \le \Delta t_j$
               Calculate $\tau_R$ using Eq. (3).
               • If $\Delta t_j \ge \tau_R$, find which reaction takes place within $\tau_R$ using Eq. (3). Update the number of molecules of different species and time as
                     $\mathbf{x} \leftarrow \mathbf{x} + \boldsymbol{\nu}_j, \quad t \leftarrow t + \tau_R.$    (10)
               • Else, do not update the state vector since no reaction was fired.
               end *while*
        B. Reset $t = t_{\text{old}}$ for the next cell.
        end *for*
(e) Set: $t = t_{\text{old}} + \Delta t_j$ (synchronize $t$ across all cells).
end *while*
```

In the supplementary material,[18] our approach and the GMP algorithm are compared. A synthetic example has also been used to demonstrate the salient features of both Brownian dynamics and cellular automata (supplementary material,[18] Section "Comparison with GMP method," Table S1, and Fig. S1).

III. RESULTS: CASE STUDIES

We start with a synthetic example (Sec. III A) to validate our stochastic operator-splitting method by comparing its results with both the GMP approach and a deterministic solution of the underlying reaction-diffusion equation computed with COMSOL. Next, we use our algorithm to model a gene expression system (Sec. III B) and CheY diffusion in *E. coli* (Sec. III C). The last two examples were carried out on a Linux-based Triton cluster of the San Diego Supercomputer Center at University of California, San Diego.

A. Synthetic reaction-diffusion case study

Suppose that at time $t = 0$, A_0 molecules of species A and B_0 molecules of species B are distributed uniformly over the left-half of the computational domain in Fig. 3(a). At $t > 0$, they diffuse into the rest of the domain and undergo a (bio)chemical reaction whose reaction product is species C,

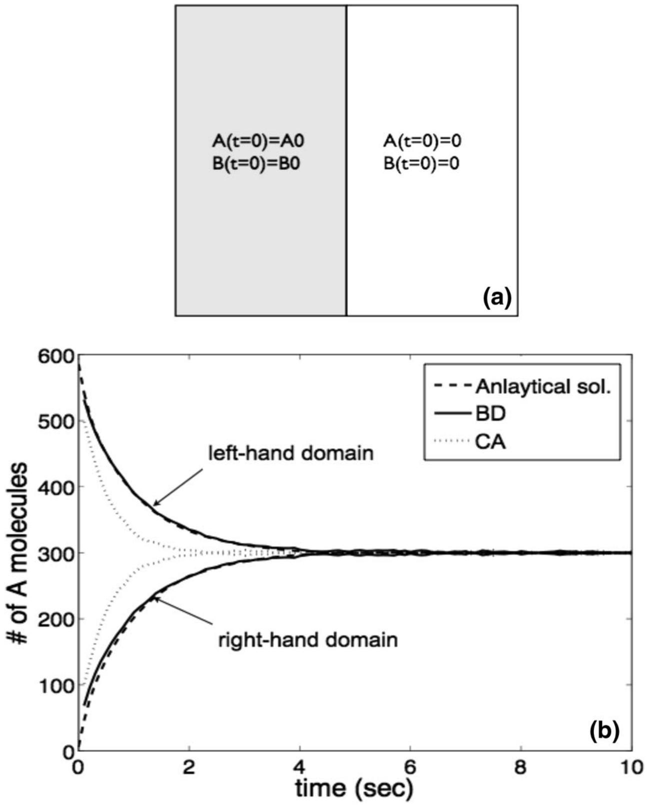

FIG. 3. $A + B \rightarrow C$ case study: (a) Initially, species A and B exist only in left-hand side. All A and B molecules and their product P diffuse with the same diffusion constant. (b) Comparison of results from analytical solution, cellular automata (CA) and Brownian dynamics (BD). The Brownian dynamics results agree with the analytical solution, while the cellular automata results do not. The increasing curves represent the number of molecules in the right half of the domain and the decreasing ones in the left half.

$$A + B \xrightarrow{k} C. \qquad (11)$$

The three species are assumed to have the same molecular diffusion coefficient $D = 10^{-13}$ m^2/s. In a more biologically realistic case study of CheY diffusion in *E. coli* (Sec. III C), the diffusion coefficients are different for different species. Here, we use a forward reaction rate constant of $k = 3 \times 10^9$ M^{-1} s^{-1}. The computational volume is $V = 10^{-15}$ L ($V = 10^{-18}$ m^3).

1. Performance analysis

First, we simulate diffusion (no reactions) with the cellular automaton and Brownian dynamics. Fig. 3(b) demonstrates that the numbers of molecules of A and B predicted with cellular automata approach their equilibrium values faster than those computed with the Brownian dynamics. The results from Brownian dynamics approach are more accurate and are in good agreement with the PDE solution. This is because Brownian dynamics provides a better approximation of the diffusion process. Furthermore, Brownian dynamics simulations are computationally more efficient than their cellular automaton counterparts (Table S1).[18]

TABLE II. A synthetic reaction-diffusion system $A + B \rightarrow C$ with diffusion constant $D = 10^{-12}$ m^2/s. The system is reaction-controlled for $L = 4$ and 8. In these cases, we set the average value of all Δt to $\overline{\Delta t} = 10\tau_D$. Computational time increases with smaller Δt or larger L (smaller cell size). For $L = 16$, the system undergoes transitions between the mixed and diffusion-controlled regimes. In this case, $\overline{\Delta t} \in [2\tau_D, 10\tau_D]$. As L increases, τ_D (or Δt) decreases and computational time increases for both algorithms. However, for any L, our method is faster than the GMP method. The relative error-rate is also shown (see Eq. (12) and related text). As L increases, the relative error-rate decreases for both methods. However, our algorithm is more accurate than the GMP method. In a similar way, for a given L, as D increases, τ_D (or Δt) decreases, and computational time increases for both algorithms.

$D = 10^{-12}$ m^2/s		Computational time (s) ($\overline{\Delta t}$ (s))		Relative error-rate (%)	
L	τ_D (s)	GMP	Our method	GMP	Our method
4	1×10^{-2}	16	1.4 (1×10^{-1})	5.2	1.17
8	2.6×10^{-3}	167	32 (2.6×10^{-2})	4.8	0.95
16	6.5×10^{-4}	4602	4022 (1.4×10^{-3})	4.7	0.91
$L = 8$			Computational time (s) ($\overline{\Delta t}$ (s))		
D (m^2/s)	τ_D (s)		GMP		Our method
10^{-11}	2.6×10^{-4}		1660		304 (2.6×10^{-3})
10^{-12}	2.6×10^{-3}		167		33 (2.6×10^{-2})
10^{-13}	2.6×10^{-2}		17		3.6 (2.6×10^{-1})
10^{-14}	2.6×10^{-1}		1.7		0.7 (1.56)

Second, we analyze the impact of a time-step and lattice size on simulations of the full reaction-diffusion system. A numerical solution (obtained using COMSOL software) of the corresponding PDEs is treated as a yardstick. First, we analyze the effect of lattice size and diffusion constant on computational time (Table II) and then we study the accuracy. For a fixed diffusion constant $D = 10^{-12}$ m^2/s, as L increases, τ_D (or Δt) decreases, and computational time increases for both algorithms. However, for any L, our method is faster than the GMP method. Similarly, for increasing diffusion constant for a given cell size ($L = 8$), τ_D (or Δt) decreases and computational time increases for both algorithms (Table II and Fig. 4). Our algorithm is faster than the GMP method because our algorithm can apply larger time-steps according to the state of the system. For example, for $D = 10^{-12}$ m^2/s, τ_D ($= 2.6 \times 10^{-3} s$) is 10 times larger than that for $D = 10^{-11}$ m^2/s. Hence, the computational time for $D = 10^{-12}$ m^2/s is about 10 times smaller than that for $D = 10^{-11}$ m^2/s. In Fig. 4, as D increases, τ_D (or Δt) decreases (Fig. 4(a)), and computational time increases (Fig. 4(b)) for both algorithms. For $D = 10^{-14}$ m^2/s, the system transitions from diffusion-controlled ($\Delta t = k_2\tau_D$; $k_2 = 2$) to reaction-controlled regime during the time-course. For $D \geq 10^{-13}$ m^2/s, the system becomes reaction-controlled ($\Delta t = 10\tau_D$). This explains the increase in the absolute value of the slope of Δt or computational time vs. D plots at $D = 10^{-13}$ m^2/s for our method.

As should be expected, the accuracy of our stochastic operator-splitting algorithm increases as the time-step and/or the cell size become smaller for both the diffusion-controlled ($D = 10^{-13}$ m^2/s) and reaction-controlled ($D = 10^{-12}$ m^2/s) scenarios in Figs. 5(a) and 5(b). The results are based on average of 8 realizations. The relative error-rate, defined as the ratio of the integrated absolute difference between a method and the PDE solution to the integrated absolute value of the

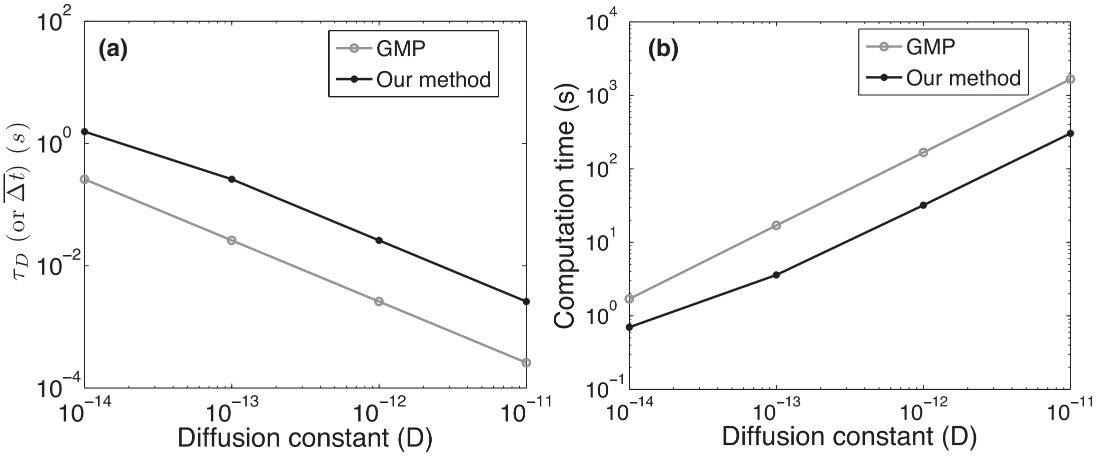

FIG. 4. $A + B \rightarrow C$ case study: Effect of diffusion constant, D (m^2/s), on (a) τ_D (or Δt) and (b) computational time for our method and the GMP method. As D increases, τ_D (or Δt) decreases and computational time increases for both algorithms. For $D = 10^{-14}$ m^2/s, the system transitions from diffusion-controlled ($\Delta t = k_2\tau_D$; $k_2 = 2$) to reaction-controlled regime during the time-course. For $D \geq 10^{-13}$ m^2/s, the system becomes reaction-controlled ($\Delta t = 10\tau_D$), explaining the increase in the absolute value of the slope of Δt or computational time vs. D plots at $D = 10^{-13}$ m^2/s for our method.

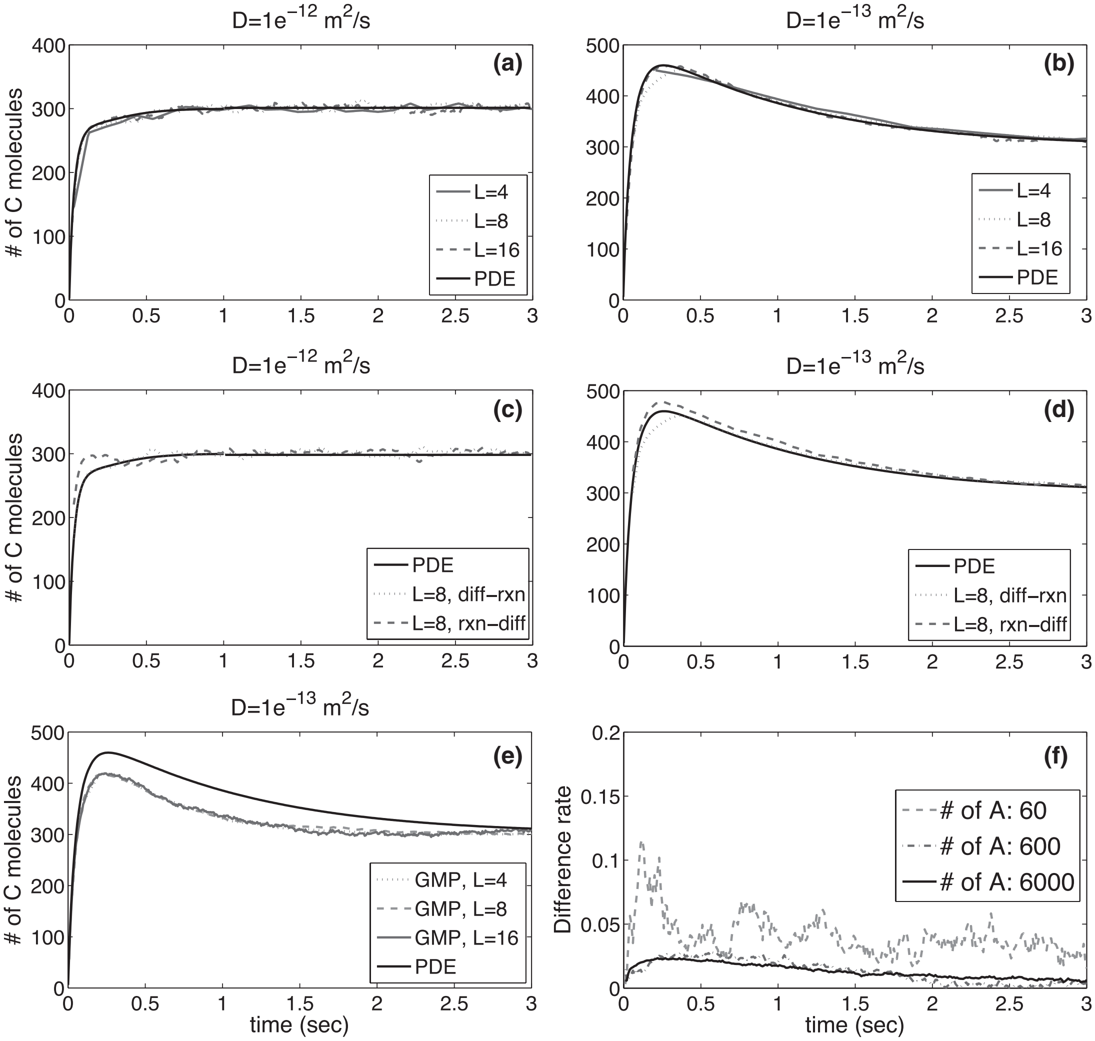

FIG. 5. $A + B \rightarrow C$ case study: (a) and (b) As cell size becomes finer, the results from our approach converge to the numerical solution. L is the number of cells along each direction (larger L represents finer cell size). The vertical axis shows the number of C molecules produced in the left half of the domain. The results are based on average of 8 realizations. (a) shows the results of reaction-controlled system, whereas (b) is for a diffusion-controlled system. (c) and (d) Dashed line is the result of reaction-first and diffusion-later order and dotted line is the reverse order. Diffusion-reaction ordering has better agreement with PDE solution than the reaction-diffusion order. It is because molecules cannot diffuse during the time-step if we treat reaction first. This ordering is effective for both diffusion-controlled and reaction-controlled systems. (e) Black line denotes the result of PDE solution. Three gray lines are for the GMP method. In the GMP method, time-step Δt is related to the cell size. So, it takes longer time to simulate the system with finer cell size. (f) As the initial number of molecules gets lower, the difference rate between deterministic solution and our stochastic solution increases. The fluctuations also increase. This result proves that the stochastic solution approaches the deterministic solution as the number of molecules (or concentration if volume is fixed) increases. In other words, randomness or stochasticity is less important at higher concentrations.

PDE solution over the time-course (ratio of the areas),

$$\text{Relative error-rate} = \frac{\sum_t |(\text{method - deterministic PDE})|}{\sum_t |(\text{deterministic PDE})|}, \tag{12}$$

is shown in Table II. For a given D, as L increases, the time step and the relative error-rate decrease for both methods. The smaller the time-step, the smaller the errors introduced by the operator-splitting procedure. However, for any L, our algorithm is more accurate than the GMP method.

Third, we investigate the impact of ordering the diffusion and reaction steps on the simulation accuracy (Figs. 5(c) and 5(d)). Both diffusion-controlled and reaction-controlled systems are considered. If the reaction step is selected to be the first part of the operator-splitting algorithm, then a diffusion process does not contribute to the system evolution during the first time-step. Hence, the reaction-first approach introduces larger errors if there is excessive inhomogeneity at the beginning. Thus, the diffusion-first (followed by the reaction step) approach is suited for both diffusion-controlled as well as reaction-controlled processes.

We further compare the accuracy of the results from our algorithm and the GMP method. Simulation results in Fig. 5(b) demonstrate an excellent agreement between our solution and the PDE solution, while the GMP method significantly underestimates both the peak number of molecules and the time it takes for the system to equilibrate (Fig. 5(e)). This finding is consistent with the results shown in Fig. 3(b), which reveal that the number of molecules estimated with cellular automata reach their equilibrium levels faster than those computed with Brownian dynamics.

2. Effect of number of molecules

Having established the agreement between our stochastic (discrete) operator-splitting algorithm and its continuum (PDE-based) counterpart for a large number of molecules, we proceed to analyze their ability to handle reaction-diffusion systems composed of small numbers of molecules. The premise here is that the smaller the number of molecules, the more inadequate the deterministic (continuum) models become and the more pronounced are the stochastic effects.

We rely on an absolute difference rate (DR),

$$\mathrm{DR} = \frac{|\text{our method - deterministic PDE}|}{\text{deterministic PDE}}, \tag{13}$$

to quantify the difference between the concentrations (relative numbers of molecules) computed with the two approaches. As expected, the DR decreases as the initial number of reacting molecules increases (Fig. 5(f)). It drops from DR ≈ 0.1 for $A_0 = 60$ to DR ≈ 0.01 for $A_0 = 600$ or 6000. Hence, stochastic and deterministic simulations yield similar results, when the number of molecules becomes large. This expected result is consistent with many other studies of randomness in reacting perfectly-mixed systems.[2]

B. Gene expression case study

The van Zon and ten Wolde model[7] of gene regulation serves as an ideal model system for studying the stochasticity effects due to both the low number of molecules and the spatial inhomogeneity. Similar to Fig. S1(a) (supplementary material[18]), RNAp molecules initially occupy the left-bottom cell of a numerical mesh, and at $t > 0$ diffuse towards a DNA molecule that is fixed in the center cell ("operator site"). Upon reaching the operator site, the RNAp molecules bind with DNA with a forward reaction rate constant k_a, forming the DNA-RNAp complex and this complex can dissociate with a backward rate constant k_d (supplementary material,[18] Table S2). In addition, it can produce a mRNA at a production rate constant k_{prod} and mRNA degrade with a decay rate constant k_{dec}. In the following, we use A, B, C, and P to denote DNA, RNAp, DNA-RNAp, and the produced mRNA, respectively.

Assuming that RNAp is the only diffusing species (i.e., DNA-RNAp and the produced mRNA do not leave the operator site), and that the molecular diffusion coefficient and reaction rates are constant (i.e., neglect anomalous diffusion due to the crowding effect and hydrodynamic effect), a continuum representation of the process is provided by a system of three ordinary differential equations and one partial differential equation,

$$\frac{d[A]}{dt} = -k_a[A][B] + k_d[C] + k_{prod}[C], \tag{14}$$

$$\frac{\partial[B]}{\partial t} = D\nabla^2[B] - k_a[A][B] + k_d[C] + k_{prod}[C], \tag{15}$$

$$\frac{d[C]}{dt} = k_a[A][B] - k_d[C] - k_{prod}[C], \tag{16}$$

$$\frac{d[P]}{dt} = k_{prod}[C] - k_{dec}[P], \tag{17}$$

where the square brackets denote concentrations of the respective species.

1. Reaction- vs. diffusion-limited processes

We set the molecular diffusion coefficient to $D = 10^{-12}$ m^2/s. Then the average time for RNAp molecules to arrive at the operator site is 0.04 s, i.e., RNAp molecules diffuse quickly throughout the system that becomes "well-mixed." Diffusion does not have a significant impact on the system's dynamics since the system is reaction-controlled. In other words, $D = 10^{-12}$ m^2/s can result in a reaction-limited (reaction-controlled) system.

Let us define a dimensionless Damköhler number Da as the ratio of typical diffusion (τ_D) and reaction ($\bar{\tau}_R$) time scales,

$$Da = \frac{\tau_D}{\bar{\tau}_R}. \tag{18}$$

A system is diffusion-limited, if $Da \gg 1$ and reaction-limited otherwise. For $D = 10^{-12}$ m^2/s, the average diffusion time $\tau_D \in [10^{-2}$ s, 10^{-1} s]. Since $\bar{\tau}_R$ is of the same order of magnitude, the system is reaction-limited. On the other hand, the diffusion coefficient $D = 10^{-15}$ m^2/s corresponds to $Da \sim 10^3$, resulting in the diffusion-limited behavior.

Fig. 6(a) demonstrates the salient features of these two transport regimes with $L = 5$. For $D = 10^{-12}$ m^2/s, the number of protein molecules computed with the Gillespie algorithm (a perfectly mixed system with no diffusion) and with our operator-splitting algorithm are in close agreement. For $D = 10^{-15}$ m^2/s, diffusion becomes important with the protein beginning to burst around 20 s after the RNAp molecules encounter DNA at the central operator site. Our results differ from their counterparts obtained by the Gillespie algorithm mainly in terms of fluctuations.

2. Time-step selection

The magnitude of the molecular diffusion coefficient D affects the choice of the time-step Δt in the stochastic operator-splitting algorithm. Figs. 6(a) and 6(b) show the number of mRNA molecules computed for a wide range of the diffusion coefficients, $10^{-15} \leq D \leq 10^{-12}$ with $L = 5$ and $L = 20$, respectively. In the case study with $L = 5$ and $D = 10^{-12}$ m^2/s, the time scales are $\bar{\tau}_R = 0.012$ s and $\tau_D = 6 \times 10^{-3}$ s; fast diffusion quickly homogenizes the system so that its behavior is reaction-controlled. Our numerical experiments suggest that setting $\Delta t = 10\tau_D$ decreases the simulation time and guarantees that a reasonable number of reactions take place during the simulation time-step.

For small diffusion coefficients ($D = 10^{-15}$ m^2/s), $\tau_D = 6.67$ s, and $\bar{\tau}_R = 0.012$ s, which means that almost all $\tau_R < \tau_D$. To ensure that a sufficient number of reactions take place during the time interval Δt, we selected $\Delta t = 2\tau_D$. Similar rules are applied for $L = 20$ as well.

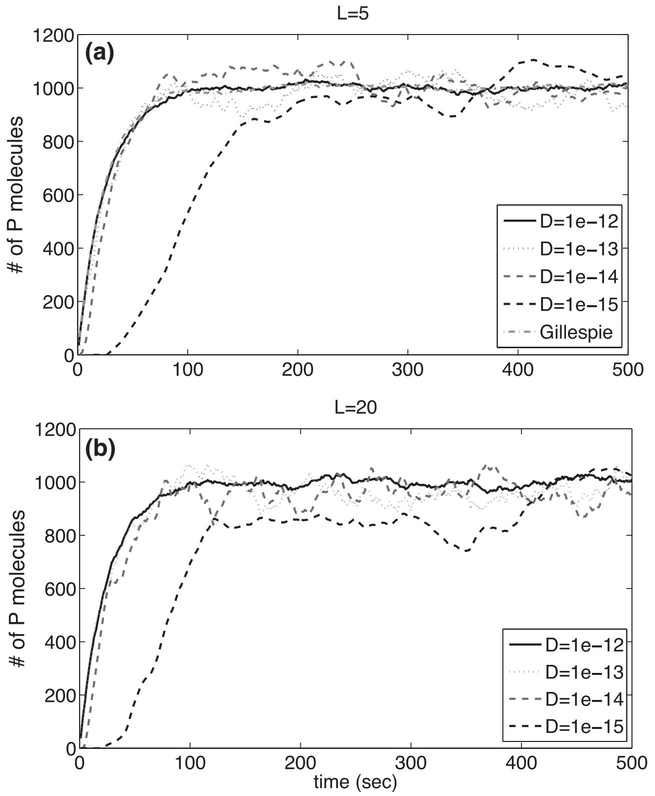

FIG. 6. Gene expression case study: (a) Dash-dotted line shows the result of Gillespie algorithm which deals with only reaction process. For $D = 10^{-12}$ m^2/s with $L = 5$, our results are similar to those obtained with the Gillespie algorithm, because it is a reaction-limited process so that diffusion does not have serious impact on the system. On the contrary, the case of $D = 10^{-15}$ m^2/s with $L = 5$ exhibits long time lag to reach the steady-state value due to diffusion effect and has much larger fluctuations. (b) In case of $L = 20$, the mesh is much finer than the above cases. The results are similar to those in (a) as our algorithm is able to adjust time-steps according to the system characteristics even though L is increased from 5 to 20.

3. Comparison with GMP method and stochastic effect

The mRNA production, predicted with the GMP algorithm and our approach on the meshes with several degrees of refinement ($L = 5$, 10, 20), are shown in Figs. 7(a) and 7(b), respectively. In a display of the lack of self-consistency, the finest mesh ($L = 20$) results in predictions that are quantitatively wrong in that the system fails to reach its equilibrium state of about 1000 proteins. It is worthwhile recalling that in the GMP algorithm, Δt is defined as the minimal diffusion time (supplementary material[18]) that cannot be adjusted. In the system under consideration, $\bar{\tau}_R > \Delta t$ so that all the reactions cannot take place during the time interval Δt ($\bar{\tau}_R = 0.65$ s and $\tau_D = 0.417$ s). These results demonstrate one of the advantages of our algorithm: unlike the GMP algorithm, our approach is capable of handling different mesh sizes by adapting appropriate time-steps.

The effects of stochasticity (noise) become apparent in predictions averaged over a smaller number of realizations (Fig. 7(c)). As should be expected from the central limit theorem, the standard deviation from the mean prediction decreases as $1/\sqrt{N_r}$. By ignoring the spatial variability, the Gillespie algorithm dampens considerably the noise present

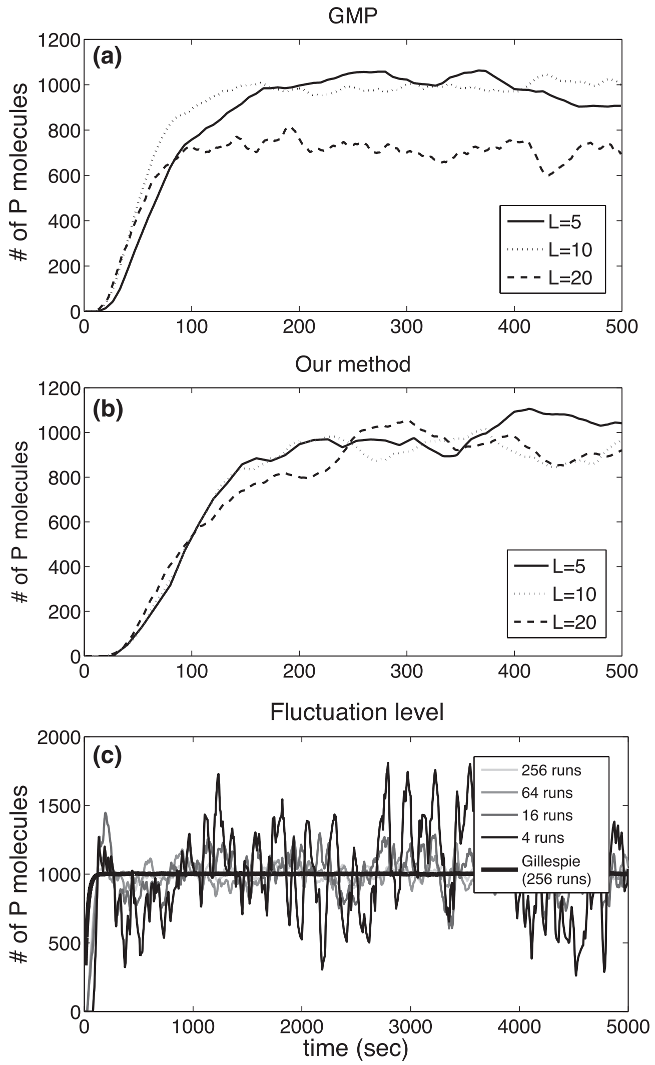

FIG. 7. Gene expression case study: (a) The result of GMP algorithm for various L and the corresponding Δt ($=\tau_D$) values. Diffusion constant has a fixed value, $D = 10^{-15}$ m^2/s. For $L = 20$ ($\bar{\tau}_R = 0.65$ s and $\tau_D = 0.417$ s), the number of P molecules does not reach its steady-state value of around 1000 because $\bar{\tau}_R > \Delta t = \tau_D$ (Table III). This means reactions cannot fully take place during Δt. (b) Our algorithm performs well for both cases of L because it classifies the system as diffusion- or reaction-controlled and decides the appropriate time-steps accordingly. (c) Fluctuations become smaller as the number of realizations become larger. In comparison to the Gillespie algorithm, our method has much higher fluctuations because it considers spatial randomness as well as randomness due to the small number of molecules.

TABLE III. Gene expression case study: Reaction time is averaged over 256 realizations of a simplified gene expression process. As cell sizes become smaller reaction times increase, since fewer molecules in each cell imply lower probability for reactions to take place within a cell. The time-step Δt in the GMP method equals τ_D, whereas Δt in our method can vary according to the system classification as reaction- or diffusion-controlled. Since the cases of $L = 5$ and 10 are diffusion-controlled, we set $\overline{\Delta t} = 2\tau_D$. For $L = 20$, the system changes from diffusion- to reaction-controlled as time progresses.

	$D = 10^{-15}$ m^2/s		GMP	Our method
L	$\Delta x\,(\mu\text{m})$	$\bar{\tau}_R$	$\tau_D = \frac{\Delta x^2}{2Dd}$ (s)	$\overline{\Delta t}$ (s)
5	0.2	0.012	6.67	13.33
10	0.1	0.095	1.667	3.33
20	0.05	0.65	0.417	2.34

in the system. The protein production continues to fluctuate in time even after it reaches its equilibrium (steady-state) value because it depends on the frequency of the encounter of RANp and DNA in the central cell. The statistics of the equilibrium protein production, i.e., its mean μ, standard deviation σ, and noise level (coefficient of variation) $\nu = \sigma/\mu$, are presented in Table S3 (supplementary material[18]).

C. CheY diffusion case study

1. System description

As a final example, we consider a chemotaxis pathway in *E. coli*. A mathematical model of this process has been developed earlier.[14] The species included in the model[14] and their simplified spatial arrangement within a cell are presented in Table S4 (supplementary material[18]) and Fig. 8(a), respectively. Table S5 (supplementary material[18]) lists a set of reactions considered in this model. The diffusing species are CheY, CheYp, and CheZ. The species CheA* (active CheA) and CheAp do not diffuse into the cytoplasm, being confined in the inner receptor cluster. The molecules of CheY and CheA* are phosphorylated in the receptor cluster located on the anterior cell wall. Once diffused into the cytoplasm, the CheYp molecules bind with four flagellar motors FliM1, … , FliM4 and the FliM · CheYp complex is produced. The four motors are located on the side walls, ordered FliM1 to FliM4 from the anterior wall (Fig. 8(a)). The reactions in the FliMs induce *E. coli*'s forward or backward motion and/or rotation.

The diffusion step in our stochastic operator-splitting algorithm is implemented in a way that the molecules reaching the cell's surface are reflected back into the cell without loss of momentum. The diffusion step is followed by the reaction step, which employs the Gillespie algorithm to simulate reactions between the molecules within each cell of a numerical mesh. We investigate the effects of varying the length of cell, L_{sv} (*sv* denotes subvolume) and time-step Δt on the performance of our algorithm, and compare it to that of the GMP algorithm.

Both GMP and our approach are conceptually different from the Smoldyn method.[14] The latter approach simulates diffusion with Brownian dynamics and keeps track of individual molecules. Unlike our algorithm, it allows for multimolecular reactions between two or three molecules only within a certain radius from each other. This reduces the computational speed and increases storage requirements, because positions of all molecules have to be stored and distances between all molecules must be calculated at each step in order to check if reactions can take place.

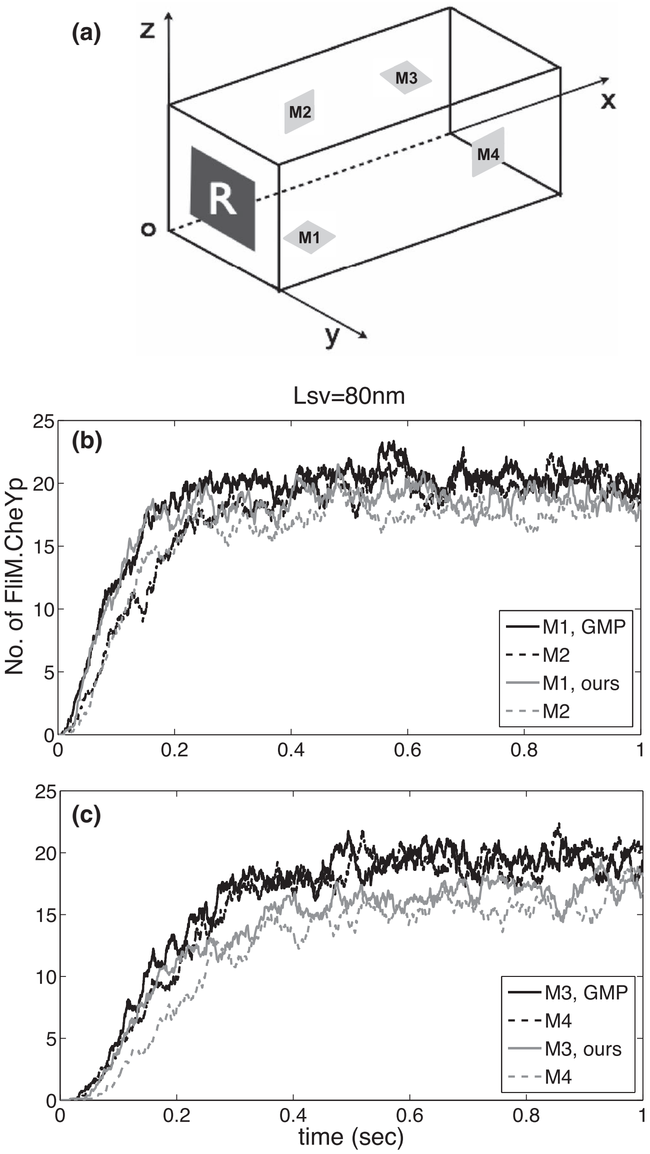

FIG. 8. CheY diffusion case study: (a) *E. coli* has length [2.48 0.88 0.88] μm along x, y, and z direction. R denotes receptor cluster located on the anterior wall and is represented by [x_min x_max; y_min y_max; z_min z_max] = [0 0.08; 0.16 0.64; 0.16 0.64] μm. CheY molecules are phosphorylated in the receptor cluster and diffuse into the cytoplasm. M1∼M4 show the location of the four flagellar motors on side walls and are located in M1 = [0.48 0.56; 0.40 0.48; 0 0.08] μm, M2 = [0.96 1.04; 0 0.08; 0.40 0.48] μm, M3 = [1.44 1.52; 0.40 0.48; 0.80 0.88] μm, and M4 = [1.92 2.00; 0.80 0.88; 0.40 0.48] μm. The remaining domain is considered as cytoplasm. (b) and (c) The simulation results from both the GMP method and our method are in good agreement although there are some differences in rise-time because the GMP method and our method use cellular automata and Brownian dynamics, respectively, to model the diffusion process (as explained in Fig. 3).

2. Simulation results

The time-course of FliM · CheYp complexes simulated with both the GMP algorithm and our stochastic operator-splitting approach is shown in Figs. 8(b) and 8(c). The GMP algorithm overestimates the equilibrium levels of the FliM · CheYp complexes and underestimates the transition-to-equilibrium times in both Fig. 8(b) (M1 and M2) and Fig. 8(c) (M3 and M4). As established in the two previous computational examples, this discrepancy is due to the errors associated with the cellular automaton treatment of diffusion in the GMP algorithm. In addition to being more accurate, our approach is also computationally more efficient than the GMP algorithm. In both algorithms, the reaction step consumes close to 99% of the total computational time. Therefore, in order to reduce the simulation time, larger Δt should be selected because the execution of the Gillespie algorithm accounts for most of the computational time. For a given mesh size L_{sv}, the time-step in the GMP algorithm is fixed by the molecular diffusion coefficient D, while in our algorithm it is more flexible according to whether the system is reaction-controlled or diffusion-controlled. The simulation time for our algorithm is 12 h, whereas it is 26 h for the GMP method. Computational time increases with decreasing L_{sv} and Δt according to $O(L_{sv}^{-3} \cdot \Delta t^{-1})$.

IV. SUMMARY AND DISCUSSION

Complex multi-scale biological systems can be analyzed with microscopic approaches, such as Green's-function reaction dynamics and the Smoldyn algorithm. These methods are accurate albeit computationally expensive and often prohibitive. On the other hand, macroscopic kinetic modeling approaches that use PDEs are amenable to numerical computation, but fail to model the physics of systems with small number of molecules accurately. Mesoscopic approaches, e.g., reaction-diffusion master equation and MesoRD, discretize space into a collection of lattice elements and extend the chemical master equation normally used in well-mixed chemical reactions into the stochastic regime for inhomogeneous systems. To facilitate faster and more accurate solutions within the mesoscopic scale framework, we have developed a stochastic simulation method which is based on operator-splitting for modeling the reaction-diffusion system. In our methodology, the time-step size is chosen automatically at each step depending upon whether the system is reaction- or diffusion-controlled. We use the Gillespie stochastic simulation algorithm for modeling the reactions and Brownian dynamics approach for modeling the diffusion process. We thus account for both spatial heterogeneity and the fluctuation in concentrations arising from the small number of molecules. Our method yields highly accurate results and has the merit of modeling both the reaction and diffusion processes in the system.

In order to validate accuracy and efficiency of our algorithm, a simple reaction-diffusion system, $A + B \rightarrow C$, is studied first. We concluded that Brownian dynamics provides much more accurate results while being faster than a Cellular automata approach. For example, Table II reveals that the error-rates for our approach and the GMP algorithm are about 1% and 5%, respectively. The average speed-up by using our method is about 5 times as compared to the GMP method for a wide range of the values of the diffusion constant. Moreover, we compared the stochastic ensemble average with the deterministic result and found out that our results converge to the deterministic result when smaller Δt and larger L are used in the simulation. We also concluded that the fluctuations become larger in case of smaller number of molecules and spatial inhomogeneity.

Towards modeling biologically realistic systems, a simplified gene expression system and CheY diffusion in *E. coli* bacteria are studied. In gene expression case study, the system is classified based on the Damköhler number, Da. If it is larger than 1, it is regarded as a diffusion-limited system and reaction-limited otherwise. In order to simulate the system accurately, the time-step, Δt, should be selected according to the dominant process. In addition, noise levels concerning the number of molecules and number of realizations are studied. It is shown that as the number of molecules or number of realizations become smaller, the noise level increases. We then simulated a more complicated system, viz., CheY diffusion in *E. Coli*, through both the GMP method and our operator-splitting algorithm. We have shown that the operator-splitting approach provides more accurate results and is faster as compared to the GMP algorithm. For a more accurate analysis of movement of *E. coli* bacteria, the chemotaxis process in which molecules move towards higher or lower concentration according to the concentration gradient should also be analyzed.[20]

In conclusion, we present a hybrid numerical method, also known as, operator-splitting method, for stochastic reaction-diffusion process with a small number of heterogeneously distributed molecules. Our approach is conceptually similar to the GMP algorithm that applies Gillespie algorithm for reaction process and Cellular automata for diffusion process. However, our method provides computational advantages in terms of accuracy and efficiency. First, molecules in Brownian dynamics can move freely without the restriction of lattice or time-step, whereas molecules in cellular automata move only to the adjacent lattices during the fixed time-step. Second, Brownian dynamics offers a more accurate simulation result than the cellular automata approach. Third, our algorithm has the flexibility of changing time-steps, depending on whether the system is reaction- or diffusion-controlled.

ACKNOWLEDGMENTS

We acknowledge the UCSD Triton Resource of San Diego Supercomputer Center (SDSC) used in this work. This research was supported by the National Heart, Lung and Blood Institute (NHLBI) Grant No. 5 R33 HL087375-02; National Science Foundation (NSF) Grant Nos. DBI-0641037, DBI-0835541, and STC-0939370; and by the Department of Energy (DOE) Office of Science, Advanced Scientific Computing Research (ASCR) program in Applied Mathematical Sciences.

NOMENCLATURE

ODE	ordinary differential equation
PDE	partial differential equation
SSA	stochastic simulation algorithm
CA	Cellular automata
BD	Brownian dynamics
GMP	Gillespie multi-particle method
ADI	alternating direction implicit
MOL	method of line
DR	Difference rate
E. Coli	*Escherichia Coli*
Da	Damköhler number
N_N	Neumann neighborhood
N_M	Moore neighborhood
N_r	number of realizations of simulations
$P[\mathbf{X}; t]$	the probability of the system being in the state $\mathbf{X}$ at time t
$P_0[\tau \mid \mathbf{X}, t]$	the conditional probability that no reactions occur during the time interval $[t, t + \tau)$
D	diffusion coefficient

F	fraction value
L	number of cells along each direction
L_{sv}	cell length or sub-volume length
T_R	time-constant, reciprocal of a_{sum} (time-interval which is not affected by random variables)
V	fixed cellular volume
Δt	time-step
$\mathbf{X(t)}$	state vector representing number of molecules of each species
$\mathbf{Y(t)}$	continuous counterpart of $\mathbf{X(t)}$
Z_j	independent random variables on (0,1)
$a_j(\mathbf{X})$	the propensity function of the j-th reaction channel
$k_1, k_1^{'}, k_2, k_2^{'}$	parameters used to decide Δt
$p(\tau, j\|\mathbf{X}, t)$	the probability that the next reaction will be the jth reaction and will occur during $[t + \tau, t + \tau + d\tau)$

Greek Letters

ν	noise level
ξ	normally distributed random number
τ_D	diffusion time constant
$\overline{\tau_R}$	reaction time constant (averaged value of reaction time over all firings)
τ_R	time-interval until next reaction takes place
σ	standard variation
μ	mean value
τ	time-interval

[1] U. S. Bhalla, P. T. Ram, and R. Iyengar, Science **297**, 1018 (2002).

[2] T. Choi, M. R. Maurya, D. M. Tartakovsky, and S. Subramaniam, J. Chem. Phys. **133**, 165101 (2010).

[3] J. M. Pedraza and A. van Oudenaarden, Science **307**, 1965 (2005).

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

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

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

[7] J. S. van Zon and P. R. ten Wolde, J. Chem. Phys. **123**, 234910 (2005).

[8] S. S. Andrews and D. Bray, Phys. Biol. **1**, 137 (2004).

[9] J. Hattne, D. Fange, and J. Elf, Bioinformatics **21**, 2923 (2005).

[10] M. Dobrzyński, J. V. Rodríguez, J. A. Kaandorp, and J. G. Blom, Bioinformatics **23**, 1969 (2007).

[11] J. V. Rodríguez, J. A. Kaandorp, M. Dobrzyński, and J. G. Blom, Bioinformatics **22**, 1895 (2006).

[12] F. Baras and M. M. Mansour, Phys. Rev. E **54**, 6139 (1996).

[13] B. Chopard and M. Droz, *Cellular Automata Modeling of Physical Systems* (Cambridge University Press, New York, 1998).

[14] K. Lipkow, S. S. Andrews, and D. Bray, J. Bacteriol. **187**, 45 (2005).

[15] J. Douglas and J. E. Gunn, Numer. Math. **6**, 428 (1964).

[16] W. E. Schiesser, *The Numerical Method of Lines: Integration of Partial Differential Equations* (Academic, San Diego, 1991).

[17] W. Hundsdorfer and J. G. Verwer, *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations* (Springer, New York, 2003).

[18] See supplementary material at http://dx.doi.org/10.1063/1.4764108 for additional text, Tables S1– S5, and Fig. S1.

[19] N. G. Van Kampen, *Stochastic Processes in Physics and Chemistry* (Elsevier, San Diego, 2007).

[20] J. Adler and W.-W. Tso, Science **184**, 1292 (1974).

Supporting Information to
Stochastic Operator-Splitting Method for Reaction-Diffusion Systems

TaiJung Choi, Mano R. Maurya, Daniel M. Tartakovsky
and Shankar Subramaniam

October 9, 2012

1 Methods

1.1 Reaction process: Gillespie algorithm

We consider a well mixed system consisting of $X_i(t)$ molecules of the i-th species ($i = 1, \dots, M$) at time t, i.e., it is at a state $\mathbf{X}(t) \equiv (X_1, \dots, X_M)$. Let $P_0[\tau|\mathbf{X}, t]$ denote the (conditional) probability of no reactions taking place during the time interval $[t, t+\tau)$ provided that the system is at state $\mathbf{X}$ at time t. Furthermore, let us assume that the reacting system is Markovian, i.e., the probability that no reactions occur during $[t, t+\tau+\mathrm{d}\tau)$ equals the product of the probabilities of no reactions occuring during $[t, t+\tau)$ and during $[t+\tau, t+\tau+\mathrm{d}\tau)$. Recalling the definition of the propensity function $a_j(\mathbf{X})$, i.e., $a_j(\mathbf{X})\mathrm{d}\tau$ is the probability that both the next reaction will be j-th reaction and it will occur during $[t+\tau, t+\tau+\mathrm{d}\tau)$, one obtains [3, 4]

$$P_0[\tau + \mathrm{d}\tau|\mathbf{X}, t] = P_0[\tau|\mathbf{X}, t]\,[1 - a_{\mathrm{sum}}(\mathbf{X})\mathrm{d}\tau]\,, \qquad a_{\mathrm{sum}}(\mathbf{X}) \equiv \sum_{j=1}^{N} a_j(\mathbf{X}) \tag{S1}$$

where N is the number of chemical reactions. Taking the limit as $\mathrm{d}\tau \to 0$ and solving the resulting ODE leads to

$$P_0(\tau|\mathbf{X}, t) = \mathrm{e}^{-a_{\mathrm{sum}}(\mathbf{X})\tau}. \tag{S2}$$

It follows from the definitions of P_0 and a_j [3, 4] that the joint probability density function $P(\tau, j|\mathbf{X}, t)$, which describes the probability that the next reaction will be the j-th reaction and will occur during $[t+\tau, t+\tau+\mathrm{d}\tau)$ given the present state of the system $\mathbf{X}(t)$, is $P(\tau, j|\mathbf{X}, t) = P_0[\tau|\mathbf{X}, t]a_j(\mathbf{X})$. Accounting for Eq. S2 gives

$$P(\tau, j|\mathbf{X}, t) = \frac{a_j(\mathbf{X})}{a_{\mathrm{sum}}(\mathbf{X})} a_{\mathrm{sum}}(\mathbf{X})\,\mathrm{e}^{-a_{\mathrm{sum}}(\mathbf{X})\tau}. \tag{S3}$$

The ratio $a_j(\mathbf{X})/a_{\mathrm{sum}}(\mathbf{X})$ represents the probability density function of a discrete random variable, and serves to determine the next reaction. The remainder of the right-hand-side of Eq. S3, $a_{\mathrm{sum}}(\mathbf{X})\exp[-a_{\mathrm{sum}}(\mathbf{X})\tau]$, is the exponential density function of a continuous random variable, which corresponds to the time at which the next reaction will occur.

To advance the system from state $\mathbf{X}(t)$, the Gillespie algorithm [3] generates two random variables r_1 and r_2 distributed uniformly on the unit interval $[0, 1]$. A discrete random value j and continuous random value τ are selected probabilistically in accordance with Eq. S3 as

$$\tau = \frac{1}{a_{\text{sum}}} \ln\left(\frac{1}{r_1}\right), \qquad \sum_{j'=1}^{j-1} a_{j'} \leq r_2 a_{\text{sum}} \leq \sum_{j'=1}^{j} a_{j'} \tag{S4}$$

where a_{sum} is the sum of all propensity functions. The system state at $t+\tau$ is updated according to $\mathbf{X}(t+\tau) = \mathbf{X}(t) + \boldsymbol{\nu}_j$ where the entries of the vector ν_j are the change in the number of molecules of various species due to the j-th reaction [3].

1.2 Diffusion process: Brownian dynamics

In cells, molecules such as proteins and metabolites, have a non-zero instantaneous speed at room temperature or at the temperature of the human body. A typical protein molecule is immersed in the aqueous medium of a living cell. It collides with other molecules in the solution, exhibiting a random walk or Brownian motion.

Let $\mathbf{X}(t) \in \mathbb{R}^3$ denote the position of a diffusing molecule at time t. Diffusive spreading of molecules of the i-th species $(i = 1, \ldots, M)$ is characterized by a molecular diffusion coefficient D_i, whose value depends on the molecule size, absolute temperature and the viscosity of a solution. The molecule's position at the end of the time interval $\triangle t$ is computed as follows [5].

1. Generate three normally distributed random numbers ξ_1, ξ_2, and ξ_3 that serve as components of the random displacement vector $\boldsymbol{\xi} = (\xi_1, \xi_2, \xi_3)^T$.

2. Compute the molecule's position at time $t + \triangle t$ as

$$\mathbf{X}(t + \triangle t) = \mathbf{X}(t) + \sqrt{2D_i \triangle t}\, \boldsymbol{\xi}. \tag{S5}$$

3. Set $t = t + \triangle t$ and go to step 1.

1.3 Diffusion process: Cellular automata

In general, cellular automata depend on mesh size and diffusion constant. Simulation accuracy and computational time vary according to neighborhood types [1]. For the two-dimensional example in Figs. 1B-C (main manuscript), molecules can diffuse to four adjacent cells (voxel) or stay in the original voxel in the von Neumann neighborhood, whereas in the Moore neighborhood they can diffuse to eight adjacent cells or stay in the original voxel. If $(0, 0)$ denotes the original voxel, the von Neumann neighborhood is a set $N_N = \{(-1, 0), (0, -1), (0, 0), (0, 1), (1, 0)\}$. The Moore neighborhood is a set $N_M = N_N \cup \{(-1, -1), (-1, 1), (1, -1), (1, 1)\}$.

The Gillespie multi-particle (GMP) algorithm [2] employs cellular automata to simulate diffusion. A diffusion-time constant τ_{D_i}, the time during which a molecule of the i-th species remains in one cell of a mesh, is given by [6]

$$\tau_{D_i} = \frac{1}{2d} \frac{(\Delta x)^2}{D_i}, \tag{S6}$$

where D_i is the diffusion coefficient for the i-th species. Moreover, a reaction-time constant $\bar{\tau}_R$ is defined as the ensemble average of the equivalent time constants for all reactions related to diffusing molecules.

1.4 Gillespie multi-particle (GMP) method

We implemented the following GMP algorithm based on [6].

```
1. Set $t_S = \Delta t = \min_i\{\tau_{D_i}\}$ for all diffusing species $i$.
2. Initialize $t = 0$ and $n_i = 1$ for all diffusing species.
3. While $t \leq t_{\text{final}}$
    • Reset $t_S = \min_i\{\tau_{D_i} \cdot n_i\}$ for all diffusing species.
    • Reset $t_{\text{old}} = t$.
    • For each cell, use the Gillespie algorithm to simulate reactions.
        (a) While $t \leq t_S$
            Calculate $\tau_R$ using Eq. S4.
              – If $t \leq t_S$, find which reaction takes place within $\tau_R$ using Eq. S4. Update number
                of species and time:
                    $\mathbf{x} \leftarrow \mathbf{x} + \boldsymbol{\nu}_j, \qquad t \leftarrow t + \tau_R$      (S7)
                where $\boldsymbol{\nu}_j$ is defined in Section 1.1.
              – Else; do not update the state vector $\mathbf{x}$ since no reaction has occured.
            end *while*
        (b) Reset $t = t_{\text{old}}$ for the next cell.
        end *for*
    • Use the cellular automata to diffuse the species.
    • Reset $n_i \leftarrow n_i + 1$ for the diffused species.
    • Set $t = t_S$.
  end *while*
```

1.5 Comparison of our method with GMP method

The GMP method [2] provides an alternative implementation of the operator-splitting approach shown in Fig. 1A (main manuscript). While our approach relies on Brownian dynamics, the GMP method models diffusion with cellular automata. This difference is significant and has far-reaching implications. First, the time step in a cellular automaton is fixed and determined by Eq. S6 in terms of the diffusion coefficient and cell size. This is because during one time step molecules in cellular automata can move from a cell only to its immediate neighbors. By relying on Brownian dynamics, our approach allows the time step to vary between the diffusion (τ_D) and reaction ($\bar{T}_R$ or $\bar{\tau}_R$) time scales. This significantly speeds up the simulations, especially when the diffusion coefficient is large and/or the cell size is small. Second, the GMP method uses the diffusion times for each species to determine when their respective molecules move from one cell of the lattice to the adjacent cells. In our algorithm, diffusing molecules of all species move during the same time step.

The following synthetic example demonstrates the salient features of both Brownian dynamics and cellular automata. We place 18 molecules of a substance P in the bottom-left cell of a lattice and allow them to diffuse towards its center cell/element (Fig. S1A). Fig. S1 shows the average number of molecules at the center cell as a function of time for several degrees of mesh refinement. Mesh refinement (the increased number of cells in each direction) does not significantly affect the accuracy of the simulation results (Fig. S1B-D) but increases the computational time (Table S1). Figs. S1E and F reveal that the Brownian dynamics reproduces a solution of the corresponding diffusion equation more accurately than the cellular automata does. This is because a particle in Brownian dynamics can move any distance in any directions while the cellular automata limits its displacement to 9 adjacent cells.

Finally, for a given degree of accuracy the Brownian dynamics simulations provide a significant computational speed-up relative to their cellular automata counterparts (Table S1). This is because the diffusion time τ_D in the cellular automata is fixed by the lattice size, whereas Brownian dynamics allow for larger time steps Δt. The run time of the Brownian dynamics simulations reported in Table S1 correspond to the same time step $\Delta t = 10$ s regardless of the lattice size, while τ_D in the cellular automata simulations varied with the mesh size and diffusion coefficient in accordance with Eq. S6. As a result, the cellular-automata simulation time increases significantly with the number of cells in each direction (L).

References

[1] B. Chopard and M. Droz. *Cellular automata modeling of physical systems.* Cambrige University Press, New York, 1998.

[2] Maciej Dobrzyński, Jordi Vidal Rodríguez, Jaap A. Kaandorp, and Joke G. Blom. Computational methods for diffusion-influenced biochemical reactions. *Bioinformatics*, 23(15):1969–1977, 2007.

[3] Daniel T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.*, 81(25):2340–2361, 12 1977.

[4] Desmond J. Higham. Modeling and simulating chemical reactions. *SIAM Rev.*, 50:347–368, 2008.

[5] N. G. Van Kampen. *Stochastic processes in physics and chemistry.* Elsevier, San Diego, 2007.

[6] J. Vidal Rodríguez, Jaap A. Kaandorp, Maciej Dobrzyński, and Joke G. Blom. Spatial stochastic modelling of the phosphoenolpyruvate-dependent phosphotransferase (pts) pathway in escherichia coli. *Bioinformatics*, 22(15):1895–1901, 2006.

		Cellular automata	Brownian dynamics
	$D = 10^{-15}$ (m^2/s)	$\Delta t = \tau_D$ (s)	$\Delta t = 10$ (s)
L	$\tau_D = (\Delta x)^2/(2Dd)$	Computational time (s)	
5	6.67	2.98	6.72
10	1.667	13.39	6.73
20	0.417	55.09	6.68

Table S1: Comparison of computational time for cellular automata and Brownian dynamics. The total of 2048 realizations are considered in order to emphasize the difference in computational time. L is the number of cells along each axis, d is the spatial dimension, and τ_D is the diffusion time constant. Brownian dynamics uses the same time step $\Delta t = 10$ s for all cell sizes, whereas cellular automata has different time steps depending on the cell size and the value of diffusion coefficient D. The simulation time increases with L. Brownian dynamics is more efficient than cellular automata.

Reaction			Rate	I.C. [nM]
DNA + RNAp	$\xrightarrow{k_a}$	C	3×10^9 M^{-1}s^{-1}	1.67
C	$\xrightarrow{k_d}$	DNA +RNAp	21.5 s^{-1}	30
C	$\xrightarrow{k_{prod}}$	P + DNA + RNAp	89.55 s^{-1}	0
P	$\xrightarrow{k_{dec}}$	ϕ	0.04 s^{-1}	0

Table S2: Gene expression case study: DNA has 1 molecule and RNAp has 18 molecules. C is the DNA·RNAp complex. System volume is 1×10^{-15} l and diffusion coefficient of RNAp is $D = 10^{-12}$ m^2/s (reaction-limited system) or $D = 10^{-15}$ m^2/s (diffusion-limited system). Abbreviation: I.C.: initial condition.

Number of realizations N_r	Mean value μ	Standard deviation σ	Noise level ν
4	1006.3	324.3	0.322
16	1015.8	149.6	0.147
64	1010.4	81.9	0.088
256	1004.4	41.2	0.041
256 (Gillespie)	1001.4	1.76	0.0018

Table S3: Gene expression case study. According to the central limit theorem, noise level or standard deviation decreases as $1/\sqrt{N_r}$. The mean values remain around 1000. The standard deviation predicted with our algorithm is much higher than that computed with the Gillespie algorithm, because our algorithm accounts for randomness due to both a small number of molecules and spatial inhomogeneity.

Species	Initial # of molecules	Diffusion constant
CheA*	1260	position fixed
CheAp	0	position fixed
CheY	8200	$D = 10^{-11}$ m^2/s
CheYp	0	$D = 10^{-11}$ m^2/s
CheZ	1600	$D = 6 \times 10^{-12}$ m^2/s
FliM$_i$ ($i = 1, \ldots, 4$)	34	position fixed
FliM$_i$·CheYp ($i = 1, \ldots, 4$)	0	position fixed

Table S4: CheY diffusion case study: 13 species and 13 reactions in the *E. coli* system. Only CheY, CheYp and CheZ molecules can diffuse; others are fixed within their original cells. Initial values are expressed in terms of number of molecules, and i denotes the index of flagellar motors.

Compartment	Reaction	Reaction constant
Receptor cluster	CheA* $\rightarrow$ CheAp	$k_f = 3.4 \times 10^1$ s^{-1}
	CheAp+CheY $\rightarrow$ CheA*+CheYp	$k_f = 10^8$ M^{-1}s^{-1}
Cytoplasm	CheY $\rightleftharpoons$ CheYp	$k_f = 5.0 \times 10^{-5}$ s^{-1} $k_b = 8.5 \times 10^{-2}$ s^{-1}
	CheZ+CheYp $\rightarrow$ CheZ+CheY	$k_f = 1.6 \times 10^6$ M^{-1}s^{-1}
FliM$_i$, $(i = 1, \ldots, 4)$	FliM$_i$+CheYp $\rightleftharpoons$ FliM$_i$·CheYp	$k_f = 5.0 \times 10^6$ M^{-1}s^{-1} $k_b = 2.0 \times 10^1$ s^{-1}

Table S5: CheY diffusion case study: k_f and k_b denote respectively forward and backward reaction rate constants for the *E. coli* system. Unimolecular and bimolecular reaction rates have dimensions [s^{-1}] and [M^{-1}s^{-1}], respectively. i denotes the index of flagellar motors.

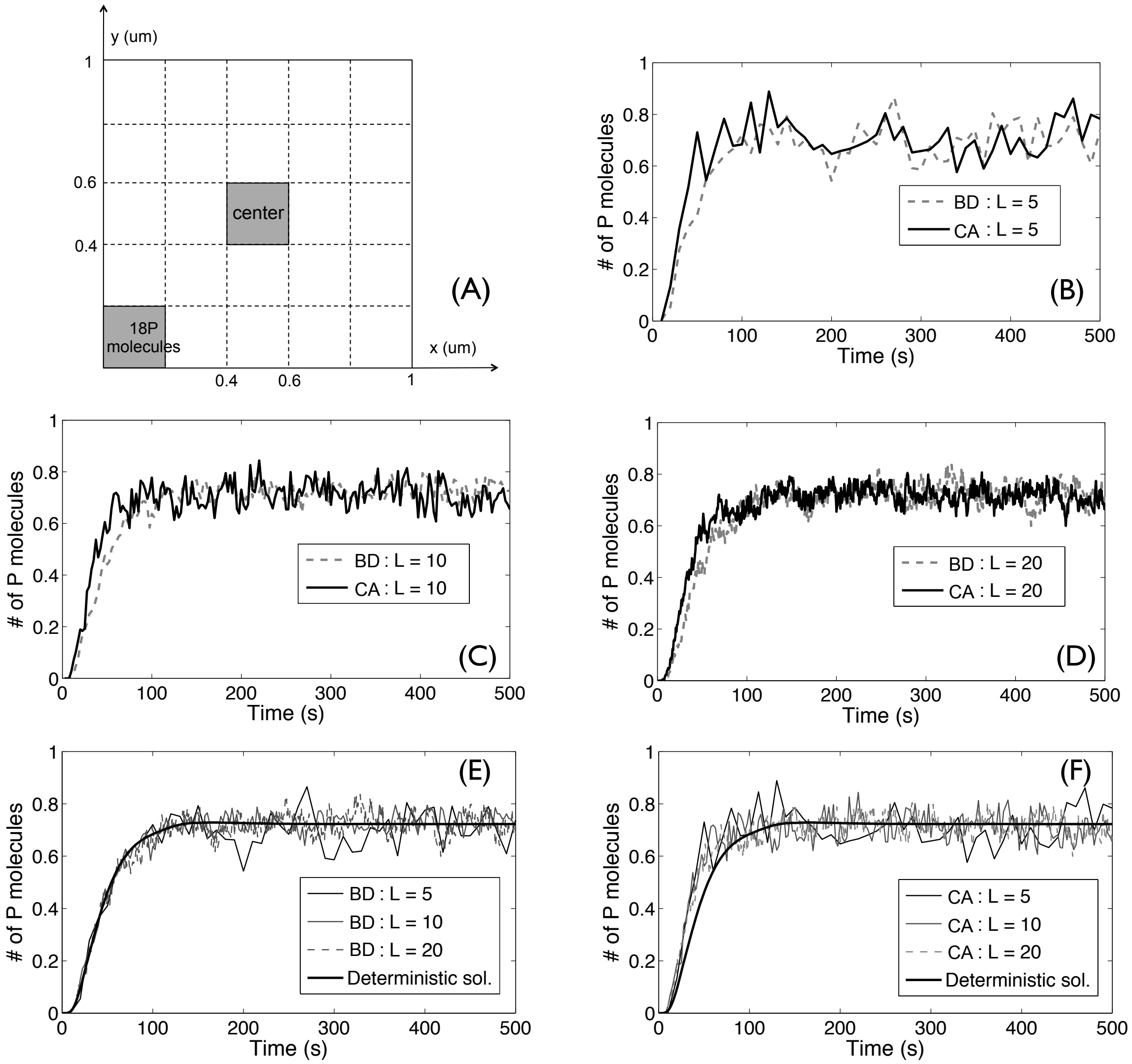

Figure S1: Temporal evolution of the count of molecules at the center cell averaged over 512 realizations of cellular automata (CA) and Brownian dynamics (BD) for several degrees of mesh refinement (L denotes the number of cells in each direction). $D = 10^{-15}$ m^2/s, $L_x = L_y = 1$ μm. (A) Initially, 18 P molecules are placed into the bottom-left cell. As time increases, they diffuse and number of P's in the center cell is counted. (B)-(D) For various values of L, the cellular automata simulation results have faster rising times than those of Brownian dynamics. (E)-(F) The simulation results are independent of the cell size (L). The Brownian dynamics results are in better agreement with the deterministic PDE solution than those of cellular automata.