Hierarchical algorithm for the reaction-diffusion master equation

View Online   Export Citation   CrossMark

Stefan Hellander[a] and Andreas Hellander[b] (iD)

AFFILIATIONS

Department of Information Technology, Uppsala University, Box 337, SE-755 01 Uppsala, Sweden

[a]stefan.hellander@it.uu.se
[b]andreas.hellander@it.uu.se

ABSTRACT

We have developed an algorithm coupling mesoscopic simulations on different levels in a hierarchy of Cartesian meshes. Based on the multiscale nature of the chemical reactions, some molecules in the system will live on a fine-grained mesh, while others live on a coarse-grained mesh. By allowing molecules to transfer from the fine levels to the coarse levels when appropriate, we show that we can save up to three orders of magnitude of computational time compared to microscopic simulations or highly resolved mesoscopic simulations, without losing significant accuracy. We demonstrate this in several numerical examples with systems that cannot be accurately simulated with a coarse-grained mesoscopic model.

I. INTRODUCTION

Spatial stochastic modeling of reaction-diffusion kinetics is a popular tool to study the fine-grained molecular details of intracellular regulation. By being able to capture the inherent spatial aspects of signal transduction as well as the discrete and stochastic nature of molecular interaction in the low-copy number regime, these types of models offer the possibility of insights not attainable by either more detailed molecular dynamics models (due to their computational cost) or phenomenological macroscopic models (due to the deterministic description).[1–6]

Spatial stochastic simulation algorithms (SSA) used in systems biology have to carefully balance the need for high spatial resolution with the need for a low computational cost, in order to study regulatory processes over physiologically relevant time scales (entire cell cycles). Two model formalisms have attracted particular attention in the field: the mesoscopic on-lattice Reaction-Diffusion Master Equation (RDME) and the microscopic off-lattice Collins-Kimball-Smoluchowski (CKS) model. In the former, proteins are modeled as point particles and are diffusing on the grid according to a discrete jump process, and they are able to react when finding themselves in the same voxel, whereas in the latter, reacting proteins are modeled as individual hard spheres and diffuse continuously in space according to Brownian motion. Due to the popularity of these modeling frameworks, several capable open-source software frameworks have been developed to support spatial stochastic modeling both for the RDME[7–9] and the particle-based model.[10–12] Some software support simulation on multiple levels and integrate one or more spatial stochastic simulators, such as VCell[13] and StochSS.[14]

The CKS model is generally considered being a more accurate model than the RDME, although there is no formal relationship between these two models in the sense that one arises as an approximation of the other. By choosing reaction rates in the RDME so that the properties of the microscopic model are captured, it is possible to relate the two models to each other formally.[15–17] With the choice of mesoscopic rate constants from Ref. 15, it is possible to match the mean binding time between two molecules in the two models, down to a critical size of the mesh.[16] For meshes finer than this critical size, the on-lattice RDME deteriorates in accuracy compared to the CKS model and thus cannot capture the microscopic dynamics accurately.

The mesoscopic on-lattice model offers superior simulation speed, assuming sufficient accuracy can be obtained with a relatively

coarse mesh. However, studies have highlighted scenarios where a very high spatial resolution is necessary to capture microscopic properties such as the rebinding time distributions accurately, in order to capture the correct macroscopic behavior.[3,17,18] For these systems, the computational cost becomes substantial both for microscopic particle-based methods and for on-lattice simulations with high spatial resolution.[15]

The reason for the rapid growth in computational cost differs in simulation based on the CKS and RDME models. Implementations of the microscopic model, being a many-body problem, scales poorly with the number of particles in the simulation. The Green's Function Reaction Dynamics (GFRD) algorithm improves performance over naive Brownian dynamics for sparse systems with relatively few particles,[19] but the computational cost still becomes overwhelming for systems with many interacting particles. If the mesh used in the RDME can be chosen relatively coarse, simulations on the mesoscopic scale is typically orders of magnitudes faster than simulations based on the CKS model and scales linearly with the number of particles. However, the on-lattice RDME suffers from stiffness, leading to a quickly growing computational cost as the mesh is refined. The number of diffusive jumps per time unit of simulation time is proportional to D/h^2, where D is the diffusion constant and h is the length of a voxel.[20] This means that, even if a good accuracy can be obtained with the RDME for mesh sizes close to the critical mesh size discussed above, the simulations might be prohibitively expensive if the entire system is simulated at this scale.

The problem for practical modeling is that spatial models often have at least a few reactions that are diffusion limited and hence require a high spatial resolution, but on the same time species that are present in relatively large copy numbers (in the hundreds or thousands). This results in a situation where neither method performs well. A natural way to approach these multiscale systems is to blend mesoscopic and microscopic methods in one single simulation. Previous work on such hybrid methods has highlighted the large computational savings made possible by a multiresolution approach.[21] A challenge is to partition the system into its microscopic and mesoscopic part without prior knowledge about the system dynamics. In previous work, Hellander *et al.* demonstrated how analysis from Ref. 20 can be used for automatic system partitioning.[21]

In addition to speeding up simulations with multiscale reaction properties, mesoscopic-microscopic hybrid methods can be applied to split simulation accuracy in different parts of the domain[22–24] and to augment mesoscopic models for situations where the mesoscopic framework is not well defined, such as for interactions between molecules and surfaces, and for 2D-3D interactions.[22]

Hybrid methods can achieve good speedups, but a distinct disadvantage is the relatively large complexity in their implementation and overhead caused by switching between data structures optimal for the respective algorithms. In this paper, we present a new pure on-lattice multiscale and multilevel method for spatial stochastic simulations. Based on our previous analysis of the accuracy of the RDME on different spatial resolutions,[15,16] we design a hierarchical simulation algorithm that employs several meshes of different resolution in order to capture the fine scale dynamics of highly diffusion-limited reactions while avoiding the need to resolve the entire systems on that same high level. In a series of

numerical examples of increasing complexity, we demonstrate an accuracy comparable with pure GFRD simulations at a simulation cost up to three orders of magnitude below state-of-the-art GFRD implementations.

II. BACKGROUND

In Sec. III, we describe a method that allows reactions to take place on different mesh resolutions depending on the degree of diffusion control. This hierarchical algorithm for the RDME (hRDME) allows for high accuracy at a much reduced cost compared to a fully microscopically resolved system, for models with multiscale properties. In this section, we first describe the underlying mesoscopic model and then briefly review the microscale model. We consider the more fine-grained microscale as the correct model later when computing the error of the mesoscopic simulations.

A. Reaction-diffusion master equation

The reaction-diffusion master equation is the natural spatial extension of the popular well-mixed Markov process description of chemical kinetics.[25,26] This model formalism is widely used in systems biology and models the state $\mathbf{x}$ of the system as a vector consisting of the discrete number of molecules of each chemical species. Formally, chemical species X_i, $i = 1 \ldots N$ participate in M chemical reactions R_j, $j = 1 \ldots M$. For example, a bimolecular reaction where species X_1 react with X_2 to form X_3 can be written as

$$X_1 + X_2 \xrightarrow{k} X_3, \qquad (1)$$
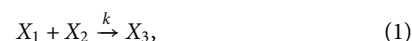
where k is the reaction rate parameter for the reaction. Using mass action kinetics, the propensity function for reaction (1) is a function of the rate constant and the copy number of the reactants X_1, X_2, $a(\mathbf{x}) = kx_1x_2$. In the Markov process formalism, the inverse of the propensity $1/a_r(\mathbf{x})$ gives the transition rate for changing states from $[x_1, x_2, x_3]$ to $[x_1 - 1, x_2 - 1, x_3 + 1]$.

The time evolution of the probability density of the system is governed by the forward Kolmogorov equation, or the chemical master equation (CME), but since this equation is infeasible to solve for systems with a large number of chemical species, kinetic Monte Carlo simulation using the direct stochastic simulation algorithm (SSA),[27] or one of its many optimized or approximate variants,[26] is normally used to analyze the system.

In the spatial stochastic case, the computational domain is partitioned into K voxels V_k using a mesh. Molecules move by diffusion, modeled as discrete jump events between adjacent voxels, according to a linear event

$$X_{ij} \xrightarrow{d_{ijk}} X_{ik}. \qquad (2)$$

The rate d_{ijk} depends on the diffusion constant of X_i and on the shape and size of the voxels.[28] Chemical reactions are modeled as in the well-mixed model, but now locally confined to individual voxels. Compared to simulation of well-mixed systems, the computational cost grows quickly with the size of the mesh. If h is a measure of the length scale of the voxel, the total number of diffusion events in a simulation scale like $1/h^2$. This stiffness problem causes RDME simulations to become highly computationally expensive if a high spatial resolution is needed.

B. Next-particle method

There are different methods for generating trajectories of the RDME. Widely used is the Next-Subvolume Method (NSM) in which the population count of each species is tracked inside the voxels.[4] While the specific choice of solver is not critical for the hRDME method, we here choose a different approach for practical reasons. It will be useful to know for how long each molecule has existed within the system, and therefore, we implement the hRDME with the Next-Particle Method (NPM)[21] as the underlying mesoscopic solver. In this section, we describe the method as implemented on a single mesh, and then, in Sec. III, we describe how it can be adapted to the case of multiple meshes.

The method is initialized by sampling a voxel for each molecule in the system. Usually, the initial distribution of molecules is uniform, in which case we sample a voxel from a uniform distribution, but this is not a necessary requirement. Just as in the case of the NSM, we will maintain an event queue sorted in descending order based on the time for each event (so that the next event in the queue is also the event that fires next). For each molecule in the system, we therefore sample tentative events and add them to the queue.

1. Initialization

1. For each molecule, add a tentative next diffusion event.
2. For all molecules participating in a unimolecular event, add the tentative next unimolecular event. Note that we only add one tentative event for each molecule, as later unimolecular events could never fire.
3. For all molecules participating in a bimolecular event, add the tentative next bimolecular event.
4. In addition, we add tentative events of the type $\emptyset \rightarrow S$. For each reaction of that type, we add one tentative next reaction.

2. Propagation

After the system has been initialized, the algorithm proceeds by executing the events in order.

1. If the next event is a diffusion event, move the molecule accordingly. If the molecule participates in any bimolecular reactions, remove those from the queue. Unimolecular events are not affected by diffusion events, so they are left on the queue. Finally, sample new tentative bimolecular events and a new diffusion event.
2. If the next event is a unimolecular event, remove all tentative events involving the reacting molecule. Initialize new molecules inside the same voxel as the reacting molecule and finally sample new tentative diffusion and reaction events for each new molecule.
3. If the next event is a bimolecular event, remove all tentative events involving either of the reacting molecules. Initialize products inside the same voxel and sample new tentative events for all the products.
4. If the next event is of the type $\emptyset \rightarrow S$, initialize a new molecule of species S into the system. Unless otherwise specified, its initial position will be uniform. Update tentative new reaction events depending on species S.
5. Repeat until the final time T.

C. Microscopic scale

On the mesoscopic scale, particles are restricted to nodes on a computational grid. In contrast, on the microscopic scale, particles diffuse freely in continuous space according to normal diffusion. On the mesoscopic scale, particles are point particles, while on the microscopic scale, reacting molecules are modeled by hard spheres. Here, reactions occur with some probability when the molecules collide. The reaction dynamics is governed by the probability density function solving the Collins-Kimball-Smoluchoski partial differential equation (PDE).[19,29,30]

Let r be the distance between two reactive molecules A and B, D is the sum of the diffusion constants, σ is the sum of the reaction radii, and k_r is the reaction rate. The probability for the distance r at time t, given that the distance was r_0 at t_0, is given by $p(r, t|r_0, t_0)$, solving the equation

$$\frac{\partial p}{\partial t} = D\Delta p(r, t|r_0, t_0) \tag{3}$$

with boundary condition

$$K\frac{\partial p}{\partial n}\bigg|_{r=\sigma} = k_r p(r, t|r_0, t_0), \tag{4}$$

where

$$K = \begin{cases} 4\pi\sigma^2 D & \text{(3D)}, \\ 2\pi\sigma D & \text{(2D)}. \end{cases} \tag{5}$$

There exist several popular implementations of solvers of this model. Prominent examples are Smoldyn,[31] MCell,[11] and eGFRD.[19] The former two, Smoldyn and MCell, take a similar approach in that they select a fixed time step and proceed by propagating the system one time step at a time. An alternative approach is implemented in eGFRD where the system is propagated in continuous time. This approach tends to be more efficient if very high accuracy is required and if the system is reasonably sparse, while Smoldyn and MCell can be significantly more efficient in other cases.

In this paper, we determine the accuracy of our simulations by comparing to the corresponding simulations on the microscopic scale. To ensure the highest possible accuracy on the microscopic scale, we have compared to the results obtained with the eGFRD algorithm, as well as another efficient implementation of a similar algorithm.[32,33]

D. Reaction rates

For reference, we provide the formula used to convert microscopic reaction rates for bimolecular reactions to the corresponding mesoscopic reaction rates. Here, we use the formula derived in Refs. 15 and 16.

Assume that the microscopic reaction rate is k_r, the sum of the reaction radii is σ and the sum of the diffusion constants is D. On a Cartesian mesh with voxels of width h, the mesoscopic reaction rates are given by

$$k_a^{\text{meso}} = \frac{k_a}{h^d}\left(1 + \frac{k_a}{D}G(h, \sigma)\right)^{-1}, \tag{6}$$

where

$$G(h, \sigma) = \begin{cases} \frac{1}{2\pi} \log\left(\pi^{-\frac{1}{2}} \frac{h}{\sigma}\right) - \frac{1}{4}\left(\frac{3}{2\pi} + C_2\right) & \text{(2D)}, \\ \frac{1}{4\pi\sigma} - \frac{C_3}{6h} & \text{(3D)}, \end{cases} \quad (7)$$

$$C_d \approx \begin{cases} 0.1951, & d = 2, \\ 1.5164, & d = 3, \end{cases} \quad (8)$$

and d is the dimension ($d = 2$ or $d = 3$).

III. METHOD

We are often interested in simulating systems displaying dynamics on widely different scales. Parts of the system require a high spatial resolution, while other parts can be simulated on a coarse-grained mesh to satisfactory accuracy.

Instead of simulating the whole system on the fine-grained level, we will here describe an approach to coupling several mesh resolutions. Some molecules will be simulated on a fine-grained mesh, while others can be simulated to high accuracy on a much more coarse-grained mesh. Molecules can also be initialized on a fine-grained mesh and, after diffusing for a sufficiently long time, be transferred to a more coarse-grained mesh.

A. A hierarchy of meshes

Here, for simplicity, we will consider Cartesian meshes only. The coarsest possible mesh is a single voxel. This corresponds to a fully well-mixed system. This mesh can then be successively refined by halving the voxel width, thus obtaining a sequence of meshes with 1, 2^3, 4^3, 8^3, …, number of voxels in 3D. By halving the width of the voxels in each step, each voxel will be fully contained within a voxel on a coarser mesh. This is not a necessary requirement for the method to work, but it does simplify the implementation and keeps the overhead of the method at a minimum.

With this particular structure of the hierarchy of meshes, in which each voxel on a finer mesh is fully contained within a voxel on a coarser mesh, it is fairly straightforward to map molecules between the different scales. The mapping is a pure preprocessing step, in which each voxel on each mesh is assigned a parent voxel in the mesh one level coarser and children voxels in the mesh one level finer. This assignment is particularly simple for Cartesian meshes, but would be possible to perform also in the case of an unstructured mesh (albeit much more computationally expensive).

B. Move molecules between meshes

The core idea of the algorithm is to transfer molecules between the different levels of the hierarchy depending on the dynamics of the system. For each species, we can determine the finest mesh resolution necessary to resolve all dynamics involving that species (see Sec. III C 5), and each molecule of that species will be initialized on that mesh resolution. Depending on how the simulation proceeds, the molecule can be moved to coarser levels in the hierarchy, and products resulting from reactions involving the molecule can be moved to finer levels in the hierarchy.

1. Move a molecule from a fine mesh to a coarse mesh

However, if the molecule survives for long enough, and thus diffuse enough, it can successively be moved to a more coarse-grained mesh resolution without losing too much accuracy. The time until we can move a molecule from a fine mesh to a coarser mesh is related to the diffusion constant D of the molecule and the width h of the voxels in the current mesh. In particular, the time t_{transfer} until we can transfer a molecule to a coarser level is given by the relation

$$h = \sqrt{\frac{6Dt_{\text{transfer}}}{C}} \quad (9)$$

for some constant C. In words, the molecule should, on average, diffuse a distance that is a multiple $\sqrt{C}$ of the voxel width on the current mesh, before we move it to a coarser mesh. This corresponds to the molecule getting "well-mixed" on the length scale of the voxels.

A voxel on the fine mesh is fully contained within a voxel on the coarse mesh, and for each molecule, we keep track of the time it has diffused since its last transfer. When it has diffused for at least t_{transfer}, it can be transferred again. When a molecule is transferred to a coarser mesh, it is simply placed in the voxel containing its current voxel. In Sec. IV A, we determine a reasonable value for C numerically.

2. Move a molecule from a coarse mesh to a fine mesh

Sometimes a molecule needs to be moved from a coarse mesh to a finer mesh in the hierarchy. For instance, when a molecule dissociates on a coarse mesh, the products might have to be initialized on a much finer mesh than that occupied by the reacting molecule. This is done by placing the molecule randomly inside one of the voxels contained within the voxel on the coarse mesh.

3. When can a molecule be moved?

A molecule is only transferred between meshes immediately after a diffusion event has fired, and before, we sample new tentative bimolecular reactions and a new tentative diffusion event.

The reason is that transfering a molecule to a new mesh is similar to executing a diffusion event; the molecule is placed inside a new voxel. This means that after transfering a molecule, we have to sample a new tentative bimolecular event and a new diffusion event. Transfering the molecule immediately following a diffusion event thus minimizes the overhead because we need to perform these operations either way. We also avoid introducing a bias by artificially discarding tentative reaction events following a molecule transfer.

C. Reactions

When simulating the RDME on a single mesh, bimolecular reactions may occur when molecules occupy the same voxel, and products of zeroth- and first-order reactions are simply placed in the voxel of the reacting molecule. In the case of the hRDME, it will not be as straightforward. In particular, we need to determine the reaction rate for two reactive molecules occupying overlapping voxels on different levels in the hierarchy of meshes. Also, molecules can

be initialized on a different level than that occupied by the reacting molecules, and in those cases, we need to determine which voxel the products should be placed inside.

1. Zeroth order reactions

Reactions of the type $\emptyset \rightarrow S$ are executed just as in the standard NSM algorithm. We sample a tentative reaction time and add the tentative event to the reaction queue. If the reaction fires, the new molecule is initialized on the mesh size required for the species S.

2. Diffusion events

A diffusion event is executed by first moving the molecule to one of the neighboring voxels with a uniform probability. Following a diffusion event, all tentative bimolecular events involving the molecule is removed from the queue. If we find a new tentative bimolecular event in the updated voxel, it is added to the queue. Unimolecular reactions are not affected by the diffusion event. A new tentative diffusion is also added to the queue.

3. Unimolecular reactions

Whenever a molecule is introduced into the system, we sample the next tentative unimolecular reaction involving this molecule. We will not have to update this event during the lifespan of the molecule, as unimolecular reactions are not affected by diffusion events.

When a unimolecular event fires, the reacting molecule is replaced by the product molecules. With each product molecule is associated a required mesh size. The products are initialized on their respective required mesh size. If this mesh is finer than the mesh occupied by the reacting molecule, we sample a voxel uniformly from the children of this voxel. If the mesh is coarser than that of the reacting molecule, we sample a voxel from the parents of the current voxel occupied by the reacting molecule.

For each new molecule introduced, we sample the corresponding tentative next events (diffusion, unimolecular, and bimolecular).

4. Bimolecular reactions

Let S_1 and S_2 be two species that react according to $S_1 + S_2 \xrightarrow{k_a} S_3$. If two molecules of species S_1 and S_2 occupy the same voxel on the same level in the hierarchy, they react just as in the NPM. However, here we frequently encounter the case where one of the molecules, M_1 of species S_1, occupies a voxel on a finer mesh than that occupied by molecule M_2 of species S_2. There is still a possibility that the molecules react if the voxel occupied by M_1 is contained within the voxel occupied by M_2. In this case, they will react as if both molecules occupied the voxel occupied by M_2 on the coarser mesh.

Assume that the molecules react with the rate $k_r^{\text{meso}}(s^{-1})$, still undetermined. The S_1 molecule diffuses with the rate $k_{\text{diff}}(s^{-1})$, and the molecule M_2 occupies the coarsest mesh in the hierarchy consisting of one voxel. This means that M_2 does not diffuse and that M_1 always occupies a voxel contained in the voxel of M_2.

Now, in each step, the probability that the molecules react, P_{react}, is given by

$$P_{\text{react}} = \frac{k_r^{\text{meso}}}{k_r^{\text{meso}} + k_{\text{diff}}}. \tag{10}$$

Each event has a waiting time, t_{event}, of

$$t_{\text{event}} = \frac{1}{k_r^{\text{meso}} + k_{\text{diff}}}. \tag{11}$$

Thus, the average time τ until the molecules react is the average number of events until a reaction fires, $(P_{\text{react}})^{-1}$, times the average time per event, t_{event}, so we get

$$\tau = (P_{\text{react}})^{-1} t_{\text{event}} = \frac{1}{k_r^{\text{meso}}}. \tag{12}$$

We now want to choose k_r^{meso} so that we obtain the correct mean binding time τ. However, for a large enough domain, the mean binding time is $(k_{\text{meso}})^{-1}$, where k_{meso} is the reaction rate for the voxel occupied by the molecule M_2. Thus, the reaction rate should be chosen as if both molecules occupied the coarser mesh, in order to reproduce the correct mean binding time of the molecules.

This argument holds in general. Consider the case where the M_2 molecule occupies a level in the hierarchy that is not the coarsest. Assume that the M_2 molecule diffuses with diffusion rate D_A and that the M_1 molecule is fixed inside a voxel. Now, there exists only one voxel that the M_2 molecule can occupy, that is also occupied by the M_1 molecule. This means that the molecules should react as if both molecules occupy the mesh of the M_2 molecule.

5. Choosing an initial mesh hierarchy

It has been shown that mesoscopic simulations of a reversible reaction become more accurate down to some mesh size $h^* \approx \frac{2}{3}\pi C_3 \sigma \approx 3.2\sigma$, where σ is the sum of the reaction radii of a reactive pair of molecules.[15] For mesh sizes below h^*, simulations actually get less and less accurate, so the optimal mesh size is h^*. It was also shown that for this mesh size, we will reproduce the correct average rebind time.

The relative error of the mean rebind time, $\tau_{\text{react}}^{\text{meso}}$, for two particles in the RDME using mesh size h is given by[21]

$$W(h) = \frac{|\tau_{\text{react}}^{\text{meso}} - \tau_{\text{react}}^{\text{micro}}|}{\tau_{\text{react}}^{\text{micro}}} = \frac{k_a}{D} G(h, \sigma), \tag{13}$$

where G is defined as in Eqs. (7) and (8).

This error was used to, given a fixed mesh with mesh size h and a chemical reaction system, partition a model into a mesoscopic and microscopic subset in a hybrid method.[21] Here, we can instead use it to, given a model, compute a largest h for which any given reaction can be handled to satisfy an error $W(h) < \epsilon$, where ϵ is a user supplied tolerance. This holds when[21]

$$k_a(1 + \epsilon)^{-1} < k_a^{\text{meso}} h^3, \tag{14}$$

where k_a is the microscopic reaction rate and k_a^{meso} is the mesoscopic reaction rate.

For each species, we can compute the coarsest mesh resolution satisfying (14). Whenever a molecule is created, it will be initialized

to this mesh size, and this will be the finest mesh size on which we will ever need to simulate this molecule.

IV. NUMERICAL EXPERIMENTS

In this section, we show that for some problems we obtain simulation results as accurate as with a microscale simulation, but with a three orders of magnitude speed-up compared to microscale simulations. Specifically, we consider problems where molecules can participate in reactions soon after a dissociation, requiring detailed spatial resolution in order to resolve the spatial correlation of the products following such a dissociation. In particular, in these examples, it will be necessary to resolve diffusion at varying length scales to simulate the systems accurately. For instance, to simulate rebinding events accurately, it is necessary to accurately simulate the diffusive process of a molecule starting in a voxel, diffusing out, and then returning to the same voxel after some time.

All timing data have been generated on a Macbook Pro 2017, 3.1 GHz CPU with 8 GB of RAM. The problems have been chosen such that (14) is satisfied to a sufficiently small ϵ only for the finest possible mesh size h^*. All results for the eGFRD algorithm have been generated with a state-of-the-art implementation,[12,19,34] available at https://github.com/gfrd/modern_egfrd.

A. Rebinding dynamics

In the first example, we show that we accurately reproduce the rebinding dynamics of a bimolecular reaction. We thus consider the rebind dynamics of the simple system

$$S_1 + S_2 \underset{k_2}{\overset{k_1}{\rightleftharpoons}} S_3. \tag{15}$$

Depending on the association rate k_1, the products S_1 and S_2 may have a high probability of rebinding fast. On the microscopic scale, the molecules are placed in contact following a dissociation event, potentially leading to many fast rebind events. On a coarse mesoscopic mesh, the products are placed in the same voxel, but we assume that they are immediately well-mixed inside that voxel. We thus lose spatial information and will see fewer fast rebinding events.

With the hRDME, we want to accurately reproduce the behavior of a simulation on the finest mesh size h^*. This means that the average rebind time will be correct, that is, the distribution of rebinding times should match the distribution of a mesoscopic simulation on the mesh size h^* (but not necessarily the rebinding-time distribution of a microscopic simulation on length scales smaller than h^*).

In Fig. 1, we show that for appropriately chosen method parameters, simulations with the hRDME on a sequence of seven meshes, $1, 2^3, 4^3, 8^3, 16^3, 32^3$, and 64^3 voxels, is able to reproduce the distribution of rebinding times obtained with a pure RDME simulation on a mesh of 64^3 voxels.

The reaction radius of all species is $\sigma = 0.002\,46$ (so that $h^* \approx 3.2 \cdot 2\sigma \approx 1/64$), the diffusion constant is $D = 1.0$, and the association rate $k_2 = 1.0$, with a domain volume of 1. Note that the dissociation rate k_1 is not important for the rebinding time distribution.
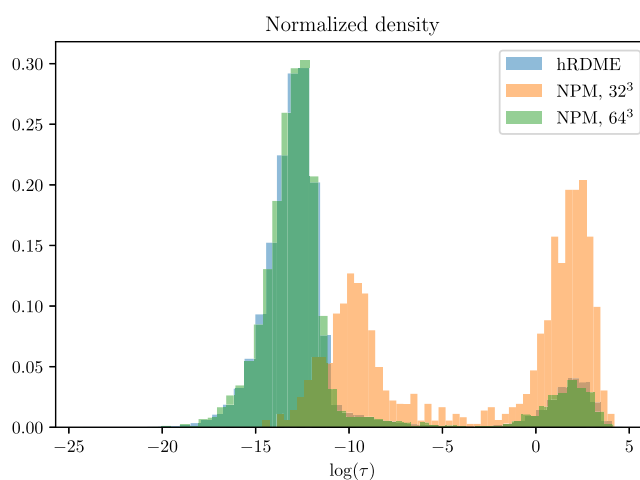

FIG. 1. The distribution obtained with the hRDME on a sequence of seven meshes overlaps the distribution obtained with the RDME on a mesh of 64^3 voxels, the finest mesh used for the hRDME simulation. The histograms are based on roughly 500 000 bimolecular reactions.

1. How to choose the constant C?

In (9), there is a constant C that controls how much the molecules should diffuse (on average) before they are moved between meshes. To reproduce the rebind distribution, we find that $C = 1$ seems sufficient. We show this in Fig. 2. However, as we have no method to determine the optimal value for C for the general case, we choose $C = 20$ for the following numerical examples. While we could likely choose a smaller C, and thus save even more computational time, we want to choose a C that is likely to work for almost any system.
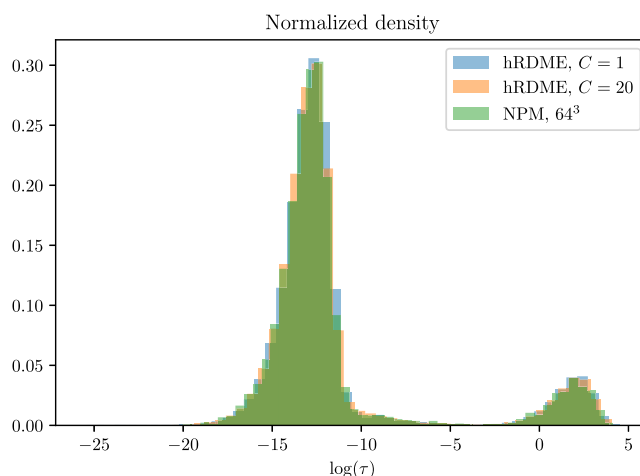

FIG. 2. Rebinding time distributions for the hRDME with $C = 1$ and $C = 20$, compared to a pure NPM simulation on a maximally resolved mesh of 64^3 voxels. The distributions overlap. While $C = 1$ could provide sufficient accuracy for many problems, we choose $C = 20$ to ensure a large enough C for the vast majority of problems.

B. Fast rebinding

Consider a simple system

$$S_1 \xrightarrow{k_1} S_{11} + S_{12} \xrightarrow{k_2} S_2. \tag{16}$$
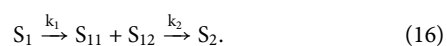
The same system, and extensions of it, has been studied in detail before.[15,21] Capturing the mean behavior of this system relies heavily on being able to capture the reaction dynamics of S_{11} and S_{12} to sufficient accuracy. To do this, the system has to be simulated on a sufficiently fine mesh. The required resolution can be determined from the criteria given by (13) for some sufficiently small ϵ. We showed that $\epsilon < 0.025$ is a reasonable choice for this system and similar systems.[21]

If the association reaction is fast, we will need a mesoscopic mesh of maximum resolution. This will make the simulation very expensive, often more expensive than a simulation with the eGFRD algorithm. However, by using the fact that most of the system can be simulated at a coarse level, we can speed up the simulation by several orders of magnitude.

For the simple system in (16), we note that both the molecules S_1 and S_2 can be safely simulated on the coarsest scale, so all molecules of these species will be initialized on the coarsest mesh. When an S_1 molecule dissociates, the products S_{11} and S_{12} are placed in the same voxel on the finest mesh. We thus resolve the possible rebind events to the highest possible accuracy, and then, if the molecules survive for some time, we can start moving them up in the hierarchy.
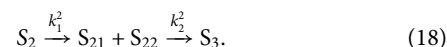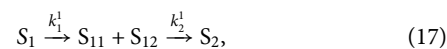
All molecules have a reaction radius $\sigma = 0.0025$ and diffuse with diffusion constant $D = 1.0$. We let $k_1 = 1.0$ and $k_2 = 1.0$. The coarsest mesh has 1 voxel, with 7 meshes total in the hierarchy, so that the finest mesh has $(2^6)^3 = 64^3 = 262\,144$ voxels. The total volume of the domain is $V = 1$, and we simulate the system for a total of 5 s and sample the time series in 100 equidistant points between 0 and 5. At time 0, we have 100 uniformly distributed S_1 molecules.

We compute the error E as the sum of the difference between the time series, where the eGFRD simulations are considered the correct solution.

The hRDME simulation is roughly 2000 times faster than eGFRD and 500 times faster than a pure mesoscopic simulation of maximum resolution. The accuracy is comparable to the accuracy of the eGFRD simulation. In Table I, we present the accuracy of pure NPM simulations on different mesh sizes compared to the hRDME and eGFRD, as well as the wall time per trajectory. The hRDME outperforms both eGFRD and highly resolved mesoscopic simulations, without losing too much accuracy.

A simple extension of the system (16) is to add another layer,

$$S_1 \xrightarrow{k_1^1} S_{11} + S_{12} \xrightarrow{k_2^1} S_2, \tag{17}$$

$$S_2 \xrightarrow{k_1^2} S_{21} + S_{22} \xrightarrow{k_2^2} S_3. \tag{18}$$

Similarly, products produced by a dissociation is placed in the same voxel on the finest mesh, while S_1, S_2, and S_3 can be safely simulated on the well-mixed scale. This means that we lose spatial information in between reactions, but this will not negatively affect the accuracy as long as we accurately capture fast rebinding events following dissociations. All parameters are as above, with all reaction rates equal to 1.0.

The speed-up compared to eGFRD is in this case approximately a factor of 1800, and 400 times faster than an NPM simulation on a maximally resolved mesh. In Fig. 3, we plot the time series of both of the systems above, simulated with the hRDME and eGFRD. There is no visible difference between the results. In Table I, we present the error (computed as above) and wall time per trajectory for simulations with the NPM on different mesh sizes, compared to the hRDME and eGFRD. In Fig. 4, we plot the probability density of the number of S_2 molecules after 1.0 s of the system (16). There is good agreement between the mean value and the standard deviation of the hRDME method and the eGFRD method.

C. Mitogen Activate Protein Kinase Cascade

Takahashi *et al.* have shown that a Mitogen Activate Protein Kinase Cascade (MAPK) system, for some parameter values, exhibits fine-grained dynamics that cannot be accurately resolved with a well-mixed model.[35] Hellander *et al.* were able to reproduce the behavior of the system with a highly resolved RDME simulation.[15] However, this required a maximally resolved mesh, in that case 64^3 voxels, making the simulation very slow and completely dominated by diffusion events. At this resolution, the RDME was slower than microscale simulations.

TABLE I. Speedup, as a multiple of the hRDME (so that a large number means a slower simulation, and a small number is faster), and relative error. We have tabulated the results of simulations with the NPM on a single mesh with varying mesh resolution (n^3 corresponds to a simulation with the NPM on a Cartesian mesh consisting of n^3 voxels), results of the hRDME, and finally, results from simulations with the eGFRD algorithm. The estimate of the error is based on 200 trajectories.

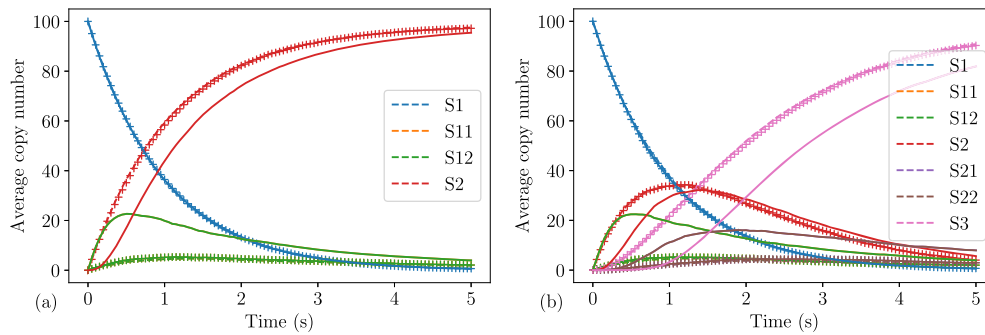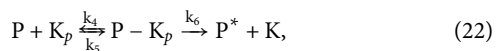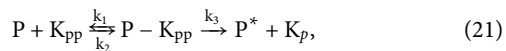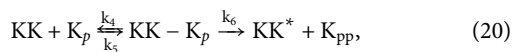| System		8^3	16^3	32^3	64^3	hRDME	eGFRD
Single (16)	Speedup	8.45	30.60	134.35	528.33	1.00	2006
	Error	0.1529	0.1292	0.0936	0.0031	0.0062	
Double (17)	Speedup	7.46	26.39	110.93	404.95	1.00	1800
	Error	0.2748	0.2419	0.1684	0.0079	0.0072	

FIG. 3. In (a), we plot the time series of the system (16) and in (b) the time series of the system (17). The eGFRD results are plotted with dashed lines and hRDME with +. For reference, we have plotted the well-mixed results in solid lines. As we can see, eGFRD results and hRDME results agree well, while there is a significant error in the well-mixed simulation results. The average is based on 200 trajectories, sampled at 101 points from 0 to 5.

The MAPK model is given by

$$KK + K \underset{k_2}{\overset{k_1}{\rightleftharpoons}} KK - K \overset{k_3}{\rightarrow} KK^* + K_p, \quad (19)$$

$$KK + K_p \underset{k_5}{\overset{k_4}{\rightleftharpoons}} KK - K_p \overset{k_6}{\rightarrow} KK^* + K_{pp}, \quad (20)$$

$$P + K_{pp} \underset{k_2}{\overset{k_1}{\rightleftharpoons}} P - K_{pp} \overset{k_3}{\rightarrow} P^* + K_p, \quad (21)$$

$$P + K_p \underset{k_5}{\overset{k_4}{\rightleftharpoons}} P - K_p \overset{k_6}{\rightarrow} P^* + K, \quad (22)$$

$$KK^* \overset{k_7}{\rightarrow} KK, \quad (23)$$

$$P^* \overset{k_7}{\rightarrow} P. \quad (24)$$

The volume of the domain is $V = 1.0$, all species diffuse with diffusion constant $D = 1.0$ and the reaction radius of all species is $\sigma = 0.002\,459\,9$. The reaction rates are

$$k_1 = 0.044\,834\,6, \quad (25)$$

$$k_2 = 1.35, \quad (26)$$

$$k_3 = 1.5, \quad (27)$$

$$k_4 = 0.092\,990\,2, \quad (28)$$

$$k_5 = 1.73, \quad (29)$$

$$k_6 = 15.0, \quad (30)$$
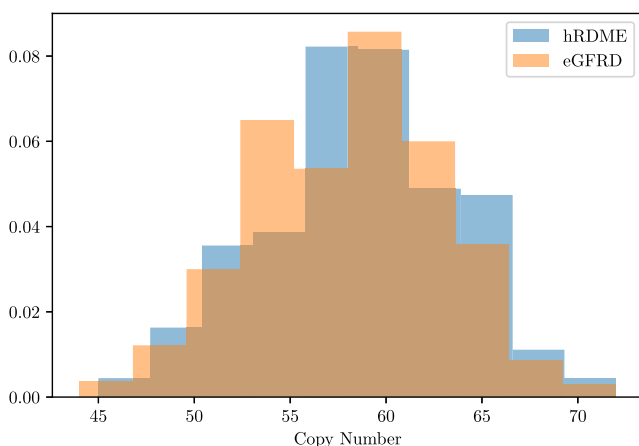
$$k_7 = 693\,147.18. \quad (31)$$

FIG. 4. Density of number of S_2 molecules after 1.0 s, estimated from 500 trajectories. The mean is 58.8 with the hRDME algorithm and 57.9 with eGFRD. The standard deviation is 4.84 and 4.87, respectively.
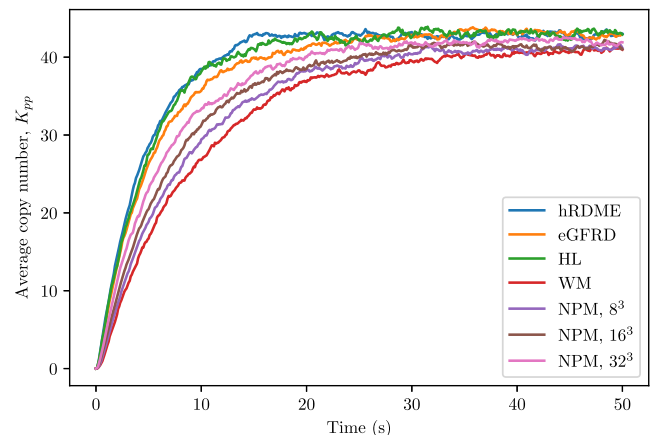

FIG. 5. The system was simulated for 50 s. The copy numbers are the average of 200 trajectories. We simulated the system with the hRDME algorithm, an implementation of the eGFRD algorithm, and with a microscale solver implemented by the authors[33] (based on the same modeling framework as eGFRD and denoted by HL in the plot). As we can see, all simulations match reasonably well, but there is a small difference compared to the eGFRD results, while the hRDME matches the other microscale implementation (HL) very well. We have also plotted results of pure mesoscopic simulations on different mesh sizes for reference, where WM is the result of a well-mixed simulation.

TABLE II. Speedup (defined as in Table I) for different mesh resolutions for the MAPK system, from a well-mixed simulation (WM) to 64^3 voxels and the hRDME algorithm, compared to eGFRD simulations. The speed-up of the hRDME is roughly a factor of 60 compared to eGFRD, with a wall time of 7.51 s per trajectory for hRDME. We have also simulated the system with $C = 1$ in the hRDME algorithm, with a total speed-up of 170 and no noticeable difference in accuracy compared to the simulations with $C = 20$.

	WM	8^3	16^3	32^3	64^3	hRDME	eGFRD
Speedup	0.05	0.35	1.36	5.36	94.15	1.00	59.03

We start the simulation with 120 copies of species K, and 30 each of species KK and species P, all initialized uniformly on the finest mesh.

Here, we show that it is possible to simulate this system with high accuracy, but with a great speed-up compared to GFRD simulations. For simplicity, we have made no assumptions about which species could be simulated on a coarse-grained level. It is possible that the simulations could be optimized even more by initializing some molecules on a mesh that is coarser than the finest mesh.

In Fig. 5, we plot the average time series of the species K_{pp}. The hRDME simulations match the microscale simulations well. We have also simulated the system with the NPM on different mesh resolutions, for reference. Timing results are presented in Table II.

V. DISCUSSION

We have shown that coupling mesoscopic simulations on different mesh sizes can save orders of magnitudes of computational time while being as accurate as microscale or highly resolved mesoscopic simulations. This methodology is also faster, and much simpler to implement, than a mesoscopic-microscopic hybrid scheme. However, there are still cases where a hierarchical mesoscopic simulation will not be sufficiently accurate. For instance, molecular crowding effects are not captured by the mesoscopic model, while they are captured with the microscopic hard-sphere model. The accuracy is also still limited by h^*;[15,16] this lower bound on the mesh size is inherent to the mesoscopic model and not due to the methodology presented herein. This implies that the hRDME cannot always accurately reproduce a microscale simulation, in particular, in cases where a mesh size finer than that given by h^* is required.

Here, we have considered structured Cartesian meshes only. While the methodology, in principle, could be extended to unstructured meshes, this is technically more difficult, and the complex shape of the voxels would incur a larger overhead. This, in turn, means that it is more difficult to handle complex geometries than it is when we have pure microscopic or mesoscopic simulations, or hybrid methods.

ACKNOWLEDGMENTS

REFERENCES

[1] M. Sturrock, A. Hellander, A. Matzavinos, and M. A. J. Chaplain, "Spatial stochastic modelling of the Hes1 gene regulatory network: Intrinsic noise can explain heterogeneity in embryonic stem cell differentiation," J. R. Soc. Interface **10**, 20120988 (2013).

[2] M. Sturrock, A. Hellander, S. Aldakheel, L. Petzold, and M. A. J. Chaplain, "The role of dimerisation and nuclear transport in the Hes1 gene regulatory network," Bull. Math. Biol. **76**, 766–798 (2013).

[3] J. S. van Zon, M. J. Morelli, S. Tănase-Nicola, and P. R. ten Wolde, "Diffusion of transcription factors can drastically enhance the noise in gene expression," Biophys. J. **91**, 4350–4367 (2006).

[4] J. Elf and M. Ehrenberg, "Spontaneous separation of bi-stable biochemical systems into spatial domains of opposite phases," Syst. Biol. **1**, 230–236 (2004).

[5] D. Fange and J. Elf, "Noise induced Min phenotypes in E. coli," PLoS Comput. Biol. **2**, e80 (2006).

[6] M. J. Lawson, B. Drawert, M. Khammash, L. Petzold, and T.-M. Yi, "Spatial stochastic dynamics enable robust cell polarization," PLoS Comput. Biol. **9**, e1003139 (2013).

[7] B. Drawert, M. Trogdon, S. Z. Toor, L. Petzold, and A. Hellander, "MOLNs: A cloud platform for interactive, reproducible and scalable spatial stochastic computational experiments in systems biology using PyURDME," SIAM J. Sci. Comput. **38**(3), C179–C202 (2016).

[8] B. Drawert, S. Engblom, and A. Hellander, "URDME: A modular framework for stochastic simulation of reaction-transport processes in complex geometries," BMC Syst. Biol. **6**, 76 (2012).

[9] J. Hattne, D. Fange, and J. Elf, "Stochastic reaction-diffusion simulation with MesoRD," Bioinformatics **21**, 2923–2924 (2005).

[10] S. S. Andrews, N. J. Addy, R. Brent, and A. P. Arkin, "Detailed simulations of cell biology with Smoldyn 2.1," PLoS Comput. Biol. **6**, e1000705 (2010).

[11] R. A. Kerr, T. M. Bartol, B. Kaminsky, M. Dittrich, J.-C. J. Chang, S. B. Baden, T. J. Sejnowski, and J. R. Stiles, "Fast Monte Carlo simulation methods for biological reaction-diffusion systems in solution and on surfaces," SIAM J. Sci. Comput. **30**, 3126 (2008).

[12] T. Opplestrup, V. V. Bulatov, G. H. Gilmer, M. H. Kalos, and B. Sadigh, "First-passage Monte Carlo algorithm: Diffusion without all the hops," Phys. Rev. Lett. **97**, 230602 (2006).

[13] J. Schaff, C. C. Fink, B. Slepchenko, J. H. Carson, and L. M. Loew, "A general computational framework for modeling cellular structure and function," Biophys. J. **73**, 1135–1146 (1997).

[14] B. Drawert, A. Hellander, B. Bales, D. Banerjee, G. Bellesia, B. J. Daigle, Jr., G. Douglas, M. Gu, A. Gupta, S. Hellander, C. Horuk, D. Nath, A. Takkar, S. Wu, P. Lötstedt, C. Krintz, and L. R. Petzold, "Stochastic simulation service: Bridging the gap between the computational expert and the biologist," PLoS Comput. Biol. **12**, e1005220 (2016).

[15] S. Hellander, A. Hellander, and L. Petzold, "Reaction rates for mesoscopic reaction-diffusion kinetics," Phys. Rev. E **91**, 023312 (2015).

[16] S. Hellander, A. Hellander, and L. Petzold, "Reaction-diffusion master equation in the microscopic limit," Phys. Rev. E **85**, 042901 (2012).

[17] D. Fange, O. G. Berg, P. Sjoberg, and J. Elf, "Stochastic reaction-diffusion kinetics in the microscopic limit," Proc. Natl. Acad. Sci. U. S. A. **107**, 19820–19825 (2010).

[18] A. Mahmutovic, D. Fange, O. G. Berg, and J. Elf, "Lost in presumption: Stochastic reactions in spatial models," Nat. Methods **9**, 1163–1166 (2012).

[19] J. S. van Zon and P. R. ten Wolde, "Green's-function reaction dynamics: A particle-based approach for simulating biochemical networks in time and space," J. Chem. Phys. **123**, 234910 (2005).

[20] A. Hellander, M. J. Lawson, B. Drawert, and L. Petzold, "Local error estimates for adaptive simulation of the reaction–diffusion master equation via operator splitting," J. Comput. Phys. **266**, 89–100 (2014).

[21] S. Hellander, A. Hellander, and L. Petzold, "Mesoscopic-microscopic spatial stochastic simulation with automatic system partitioning," J. Chem. Phys. **147**, 234101 (2017).

[22] A. Hellander, S. Hellander, and P. Lötstedt, "Coupled mesoscopic and microscopic simulation of stochastic reaction-diffusion processes in mixed dimensions," Multiscale Model. Simul. **10**, 585–611 (2012).

[23] M. B. Flegg, S. Hellander, and R. Erban, "Convergence of methods for coupling of microscopic and mesoscopic reaction-diffusion simulations," J. Comput. Phys. **289**, 1–17 (2015).

[24] M. Flegg, J. Chapman, and R. Erban, "The two-regime method for optimizing stochastic reaction-diffusion simulations," J. R. Soc. Interface **9**, 859–868 (2012).

[25] N. G. Van Kampen, *Stochastic Processes in Physics and Chemistry* (Elsevier, 1992), Vol. 1.

[26] D. T. Gillespie, A. Hellander, and L. R. Petzold, "Perspective: Stochastic algorithms for chemical kinetics," J. Chem. Phys. **138**, 170901 (2013).

[27] D. T. Gillespie, "A general method for numerically simulating the stochastic time evolution of coupled chemical reacting systems," J. Comput. Phys. **22**, 403–434 (1976).

[28] S. Engblom, L. Ferm, A. Hellander, and P. Lötstedt, "Simulation of stochastic reaction-diffusion processes on unstructured meshes," SIAM J. Sci. Comput. **31**, 1774–1797 (2009).

[29] M. v. Smoluchowski, "Versuch einer mathematischen Theorie der Koagulationskinetik kolloider Lösungen," Z. Phys. Chem. **92**, 129–168 (1917).

[30] F. C. Collins and G. E. Kimball, "Diffusion-controlled reaction rates," J. Colloid Sci. **4**, 425–437 (1949).

[31] S. S. Andrews and D. Bray, "Stochastic simulation of chemical reactions with spatial resolution and single molecule detail," Phys. Biol. **1**, 137–151 (2004).

[32] S. Hellander and P. Lötstedt, "Flexible single molecule simulation of reaction–diffusion processes," J. Comput. Phys. **230**, 3948–3965 (2011).

[33] S. Hellander, "Single molecule simulations in complex geometries with embedded dynamic one-dimensional structures," J. Chem. Phys. **139**, 014103 (2013).

[34] T. R. Sokolowski, J. Paijmans, L. Bossen, T. Miedema, M. Wehrens, N. B. Becker, K. Kaizu, K. Takahashi, M. Dogterom, and P. R. ten Wolde, "eGFRD in all dimensions," J. Comput. Phys. **150**, 054108 (2019).

[35] K. Takahashi, S. Tănase-Nicola, and P. R. ten Wolde, "Spatio-temporal correlations can drastically change the response of a MAPK pathway," Proc. Natl. Acad. Sci. U. S. A. **107**, 2473–2478 (2010).