SIAM J. SCI. COMPUT.
Vol. 38, No. 1, pp. A55–A83

ANALYSIS AND DESIGN OF JUMP COEFFICIENTS IN DISCRETE STOCHASTIC DIFFUSION MODELS*

LINA MEINECKE†, STEFAN ENGBLOM†, ANDREAS HELLANDER†, AND PER LÖTSTEDT†

Abstract. In computational systems biology, the mesoscopic model of reaction-diffusion kinetics is described by a continuous time, discrete space Markov process. To simulate diffusion stochastically, the jump coefficients are obtained by a discretization of the diffusion equation. Using unstructured meshes to represent complicated geometries may lead to negative coefficients when using piecewise linear finite elements. Several methods have been proposed to modify the coefficients to enforce the nonnegativity needed in the stochastic setting. In this paper, we present a method to quantify the error introduced by that change. We interpret the modified discretization matrix as the exact finite element discretization of a perturbed equation. The forward error, the error between the analytical solutions to the original and the perturbed equations, is bounded by the backward error, the error between the diffusion of the two equations. We present a backward analysis algorithm to compute the diffusion coefficient from a given discretization matrix. The analysis suggests a new way of deriving nonnegative jump coefficients that minimizes the backward error. The theory is tested in numerical experiments indicating that the new method is superior and also minimizes the forward error.

Key words. stochastic simulation, diffusion, unstructured mesh, finite element method

AMS subject classifications. 65C40, 65C05, 65M60, 60H35, 92C05

DOI. 10.1137/15M101110X

1. Introduction.

The molecular pathways that regulate cellular function are inherently spatial. Cells have a high level of subcellular organization, such as a confined nucleus in eukaryotes or membrane bound reaction complexes. Macromolecules are transported by passive diffusion or active transport, driven by molecular motors, between different areas in the cell in order to arrive at the correct location to perform their function. For example, many gene regulatory pathways rely on a cytoplasmic component, where a signal is propagated from the cell membrane to the nucleus, and a nuclear component, where transcription factors bind to DNA to regulate the expression of genes.

On a *macroscopic* modeling level, the diffusion equation—a partial differential equation (PDE)—is used to describe the time evolution of the concentration of a population of molecules undergoing diffusion. This is a valid model if molecules are abundant. But in cellular regulatory networks, key proteins such as transcription factors are present only in low copy numbers, and the deterministic PDE model becomes inaccurate. Experiments [8, 26, 32, 34, 35, 38, 45] and theory [14, 33] have shown the importance of accounting for intrinsic noise when modeling cellular control systems. Consequently, we need spatial stochastic simulation methods, and diffusion in particular is described by a random walk. We can distinguish between two levels of accuracy.

*Submitted to the journal's Methods and Algorithms for Scientific Computing section March 5, 2015; accepted for publication (in revised form) October 20, 2015; published electronically January 6, 2016. This work was supported by Swedish Research Council grant 621-2011-3148, the UPMARC Linnaeus center of Excellence, the Swedish strategic research programme eSSENCE, and NIH grant for StochSS 1R01EB014877-01.

http://www.siam.org/journals/sisc/38-1/M101110.html

†Division of Scientific Computing, Department of Information Technology, Uppsala University, SE-75105 Uppsala, Sweden (lina.meinecke@it.uu.se, stefane@it.uu.se, andreas.hellander@it.uu.se, perl@it.uu.se).

On the *mesoscopic* level we use a discrete Brownian motion to model the jump process of the molecules. The domain is partitioned into compartments or voxels. The state of the system is the number of molecules of each species in each voxel. Molecules can jump between neighboring voxels and react if they are in the same voxel. The probability density function (PDF) for the probability of being in a state at a certain time satisfies a master equation. If bimolecular reactions are included, in general there is no analytical solution for the PDF, and a numerical solution is difficult to compute due to the high dimensionality of the state space. Instead, the stochastic simulation algorithm (SSA) can be used to generate trajectories of the system. It was first developed by Gillespie [16, 17] for reactions independent of space. Its efficiency has been improved in [5, 15], and it is extended to space dependency with a Cartesian partitioning of the space in [7, 20, 22]. A more accurate description is the space-continuous *microscopic* level, where individual molecules are followed along their Brownian trajectories. Methods and software for this approach are found in [1, 6, 23, 41, 50].

In this work, we focus on diffusion at the mesoscopic level. The probability per unit of time for a molecule to jump from its voxel to a neighbor is obtained by a discretization of the Laplace operator in the diffusion equation on the same mesh. A mathematically equivalent interpretation is that this probability is obtained from a discretization of the Fokker–Planck equation for Brownian motion. The resulting matrix is the generator of a Markov process, and all the off-diagonal entries, which represent transition rates, need to be nonnegative. On the macroscopic level of deterministic PDEs, requiring nonnegative jump coefficients enforces the discrete maximum principle [46, 48]. To represent the complicated geometries present in cells (e.g., mitochondria or convoluted membranes), we work with unstructured meshes, meaning triangular or tetrahedral meshes in two and three dimensions. Many interesting cellular processes happen on the membranes. Using a vertex centered discretization allows us to couple diffusion in the bulk to diffusion on the surface of the domain in a straightforward way. If a molecule reaches a boundary node, we can use the surface mesh for its two-dimensional (2D) diffusion on the membrane.

Piecewise linear finite elements on unstructured meshes are used in [9] to obtain the jump propensities. Software exists for discretizing PDEs with the finite element method (FEM) on a given mesh; see, e.g., [30]. In two dimensions, mesh generators are usually able to provide high quality meshes [10], but in three dimensions the mesh quality decreases and negative off-diagonal elements often appear in the FEM discretization matrix [3, 24, 27]. This matrix can then no longer be interpreted as the generator matrix to a Markov process and thus provide transition rates. For our application in stochastic simulations in systems biology, we need to modify this discretization matrix to guarantee nonnegative jump coefficients. In [9] the negative entries are set to zero and the diagonal element is recalculated so that the row sum equals zero. This changes the diffusion speed and leads to errors in, for example, the time a signal needs to propagate from the nucleus to the cell membrane. To address this, in previous work [31] we developed a method that preserves mean first passage times. Another approach to obtain nonnegative jump coefficients is to use the finite volume method (FVM). But, as we will see, despite positive coefficients the vertex centered FVM scheme does not approximate diffusion more accurately than a filtered FEM discretization for typical meshes. These methods make the stochastic simulation of diffusion mesh dependent, but this also holds for the accuracy of the numerical solution of the PDE. FEM and FVM coefficients have been modified in

[4, 13, 42] to be nonnegative, but they depend on the PDE solution, which makes them unsuitable for stochastic simulation.

In this paper we analyze the error introduced by modifying the discretization matrix to enforce nonnegative jump coefficients. Since the concentration of the species simulated by the SSA converges towards the solution of the diffusion equation [28, 29], we quantify the error in this deterministic limit. We use backward analysis to find the diffusion equation solved by the new discretization matrix. We study two error estimates: the *backward error,* describing the difference in the diffusion in the equations, and the *forward error,* describing the error in the solutions to the equations. The analysis suggests a new method to obtain a nonnegative discretization by minimizing the backward error.

In section 2 we describe the mesoscopic model and how the jump coefficients are obtained for unstructured triangular and tetrahedral meshes. In section 3, we develop theory to bound the forward error by the backward error. An algorithm is provided in section 4 for calculating the backward error, and then in section 5 we show how new error minimizing jump coefficients can be computed. In the experiments in section 6, we analyze the errors numerically, test our new method, observe that it also minimizes the forward error in agreement with the estimates in section 3, and discuss possibilities for a practical implementation. Final conclusions are drawn in section 7.

Vectors and matrices are written in boldface. A vector $\mathbf{u}$ has the components u_i, and the elements of a matrix $\mathbf{A}$ are A_{ij}. For vectors and matrices, $\|\mathbf{u}\|_p$ denotes the vector norm in ℓ_p and $\|\mathbf{A}\|_p$ denotes its subordinate matrix norm, and $\|\mathbf{u}\|_{\mathbf{A}}^2 = \mathbf{u}^T\mathbf{A}\mathbf{u}$ for a positive definite $\mathbf{A}$. The derivative of a variable u with respect to time t is written u_t. If $\mathbf{u}(\mathbf{x})$, $\mathbf{x} \in \Omega$, varies in space, then $\|\mathbf{u}\|_{L^p}^p = \int_\Omega \|\mathbf{u}\|_p^p \, d\Omega$.

2. Mesoscopic model of diffusion. To model diffusion in discrete space, the domain of interest Ω is partitioned into nonoverlapping voxels $\mathcal{V}_k$, $k = 1, \ldots, N$. Each voxel $\mathcal{V}_j$ has a node or vertex with the coordinates $\mathbf{x}_j$ in the interior; see Figure 1. If $\mathcal{V}_i$ and $\mathcal{V}_j$ are neighbors, the vertices $\mathbf{x}_i$ and $\mathbf{x}_j$ are connected by the edge $\mathbf{e}_{ij}$. Molecules in a voxel $\mathcal{V}_j$ can diffuse by jumps to a neighboring voxel $\mathcal{V}_i$ along the edge e_{ij}.

The state of the system is the discrete number of molecules of each species in each voxel. Let y_j be the number of molecules of chemical species Y in $\mathcal{V}_j$. The jump rate

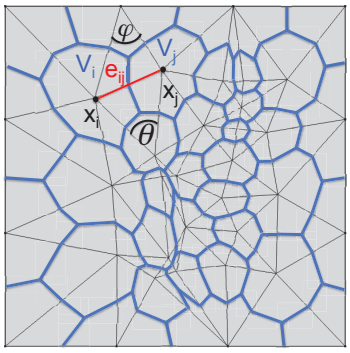


FIG. 1. *The primal triangular mesh (thin lines), defining the edges e_{ij}, and the dual mesh (thick lines), defining the voxels $\mathcal{V}_i$.*

λ_{ji} from $\mathcal{V}_j$ to a neighboring $\mathcal{V}_i$ needs to satisfy the condition

$$\lambda_{ji} \geq 0. \tag{2.1}$$

The total jump rate out of voxel $\mathcal{V}_j$ is $\lambda_j = \sum_{i,i\neq j} \lambda_{ji}$. The next time for a jump from $\mathcal{V}_j$ is exponentially distributed with the intensity $\lambda_j y_j$. Voxel $\mathcal{V}_i$ is chosen as the destination with a probability proportional to λ_{ji}. After a jump, the number of molecules is updated and the time for a new jump is determined. This is the SSA of Gillespie [16] for simulating mesoscopic diffusion of molecules between the voxels.

Algorithm 1 Stochastic simulation algorithm for diffusion [16].

1: Initialize $y_k, k = 1, \ldots, N$, in the N voxels at $t = 0$.
2: Sample the exponentially distributed time t_k with rate $\lambda_k y_k$ to the first diffusion event in all N voxels.
3: Let t_j be the minimum of all t_k. If $t_j > T$, then stop; otherwise continue.
4: For the jump from $\mathcal{V}_j$, sample a jump to $\mathcal{V}_i$ with probability λ_{ji}/λ_j.
5: Update $t := t_j$, y_i, and y_j. Sample Δt_i with the rate $\lambda_i y_i$ and Δt_j with the rate $\lambda_j y_j$ and recompute $t_i = t + \Delta t_i$ and $t_j = t + \Delta t_j$. Go to 3.

We will now show how to derive the propensities λ_{ij} from a discretization of the diffusion equation. Let $\mathbf{y}$ be a vector with entries y_i, describing the number of Y molecules in voxel $\mathcal{V}_i$. The probability distribution $p(\mathbf{y}, t)$ for the distribution of the molecules is the solution to the diffusion master equation

$$p_t(\mathbf{y}, t) = \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_{ij}(\mathbf{y} - \boldsymbol{\mu}_{ij}) p(\mathbf{y} - \boldsymbol{\mu}_{ij}, t) - \lambda_{ij}(\mathbf{y}) p(\mathbf{y}, t), \tag{2.2}$$

where $\mu_{ij,i} = -1$, $\mu_{ij,j} = 1$, and $\mu_{ij,k} = 0$ for $k \neq i, j$. Calculating the expected value $\overline{y}_i$ of the number of molecules in each voxel i leads to a system of ordinary differential equations (ODEs) for the mean concentration $u_i = \overline{y}_i/|\mathcal{V}_i|$ in each voxel $\mathcal{V}_i$,

$$u_{it} = \sum_{j=1}^{N} \frac{|\mathcal{V}_j|}{|\mathcal{V}_i|} \lambda_{ji} u_j - u_i \sum_{j=1}^{N} \lambda_{ij}; \tag{2.3}$$

see [9]. This can be interpreted as a discretization of the diffusion equation

$$\begin{aligned} u_t = \gamma \Delta u = \nabla \cdot (\boldsymbol{\gamma} \nabla u), &\qquad \mathbf{x} \in \Omega,\ t \geq 0, \\ \mathbf{n} \cdot \nabla u = 0, &\qquad \mathbf{x} \in \partial\Omega,\ t \geq 0, \\ u = u_0, &\qquad \mathbf{x} \in \Omega,\ t = 0, \end{aligned} \tag{2.4}$$

with the diffusion coefficient γ and $\boldsymbol{\gamma} = \gamma \mathbf{I}$. Thus, the jump rates λ_j and λ_{ji} can be computed using discretizations of the diffusion equation on the triangular or tetrahedral mesh. The space derivatives in (2.4) are approximated in the voxels by $\mathbf{D}$ to obtain equations for the unknowns u_i,

$$\mathbf{u}_t = \mathbf{D}\mathbf{u}. \tag{2.5}$$

A discretization with FEM using piecewise linear Lagrangian basis and test functions yields a mass matrix $\mathbf{M}$ and a stiffness matrix $\mathbf{S}$. The diagonal $\mathbf{A}$ is obtained after

mass lumping of $\mathbf{M}$. The diagonal elements are $A_{jj} = |\mathcal{V}_j|$. Then the system matrix in (2.5) is

$$\mathbf{D} = \mathbf{A}^{-1}\mathbf{S}. \tag{2.6}$$

Let h be a measure of the mesh size. The solution of (2.5) converges to the solution of (2.4) when $h \to 0$, and the difference between them is $\mathcal{O}(h^2)$. If the off-diagonal elements D_{ij} in $\mathbf{D}$ are nonnegative, then these are taken as the jump coefficients λ_{ji} in the SSA in Algorithm 1 scaled by the volumes of the voxels $|\mathcal{V}_i|$ and $|\mathcal{V}_j|$,

$$\lambda_{ji} = D_{ij}\frac{|\mathcal{V}_i|}{|\mathcal{V}_j|} = \frac{S_{ij}}{|\mathcal{V}_j|}; \tag{2.7}$$

see [9]. The concentrations $y_j/|\mathcal{V}_j|$ computed by the SSA converge in the limit of large numbers of molecules to the concentrations in (2.5) by [28, 29].

In two dimensions, the entry S_{ij} corresponding to edge e_{ij} is

$$S_{ij} = \sin(\varphi + \theta)/2\sin(\varphi)\sin(\theta), \tag{2.8}$$

where φ and θ are the two angles opposing e_{ij} [49]; see Figure 1. If $\varphi + \theta > \pi$, then $S_{ij} < 0$, and we can no longer use it to define a jump propensity. A similar condition exists in three dimensions [49]. Mesh generators in two dimensions are usually able to construct meshes leading to positive S_{ij} [10], but in three dimensions, negative off-diagonal entries often occur [24]. The extra requirement in systems biology to have nonnegative off-diagonal elements in the stiffness matrix is a sufficient but unnecessary condition to fulfill the discrete maximum principle when solving the PDE (2.4) numerically [46, 48].

We now present three different methods of modifying the stiffness matrix $\mathbf{S}$, or the discretization matrix $\mathbf{D}$ containing off-diagonal negative coefficients, so that we can interpret them as the generator matrix of the Markov process simulated by the SSA. The discretization matrix $\mathbf{D}$ is modified to $\tilde{\mathbf{D}}$ in [9] such that, if $D_{ij} < 0$, then $\tilde{D}_{ij} = 0$ and $\tilde{D}_{ii} = -\sum_{j=1}^{n_i}\tilde{D}_{ij}$, where n_i is the number of edges leaving vertex i. This method of calculating the jump coefficients by eliminating the negative contributions is denoted here by nnFEM (nonnegative FEM). Convergence of the solution to the equation with the diffusion operator $\gamma\Delta$ is lost, but nonnegative jump coefficients are defined. Solving the system of equations

$$\tilde{\mathbf{u}}_{ht} = \mathbf{A}^{-1}\tilde{\mathbf{S}}\mathbf{u}_h = \tilde{\mathbf{D}}\mathbf{u}_h, \tag{2.9}$$

however, can be viewed as a discrete approximation to a perturbed diffusion equation

$$\tilde{u}_t = \nabla\cdot(\tilde{\boldsymbol{\gamma}}\nabla\tilde{u}). \tag{2.10}$$

The diffusion matrix $\tilde{\boldsymbol{\gamma}}$ belongs to $\mathbb{R}^{2\times 2}$ in two dimensions and $\mathbb{R}^{3\times 3}$ in three dimensions, is symmetric, and should be positive definite for all $\mathbf{x}$. If $\tilde{\boldsymbol{\gamma}}$ is only positive semidefinite, it has at least one eigenvalue equal to zero, which means that there is no diffusion along the direction of the corresponding eigenvector. It is unrealistic to expect this to happen inside living cells, and we do not consider this case, although the following analysis can be generalized to the positive semidefinite case.

Another option is to choose a straightforward FVM. If the boundary $\partial\mathcal{V}_j$ of a voxel $\mathcal{V}_j$ consists of n_j straight segments (two dimensions) or flat faces (three dimensions)

$\partial\mathcal{V}_{ji}, i = 1, \ldots, n_j$, of length or area $|\partial\mathcal{V}_{ji}|$ with normal $\mathbf{n}_{ji}$ of unit length, then

$$\int_{\mathcal{V}_j} \nabla \cdot (\boldsymbol{\gamma}\nabla u)\, dv = \int_{\partial\mathcal{V}_j} \mathbf{n} \cdot \boldsymbol{\gamma}\nabla u\, ds \approx \sum_{i=1}^{n_j} \mathbf{n}_{ji} \cdot \boldsymbol{\gamma}\, \mathbf{e}_{ji}(u_i - u_j)\frac{|\partial\mathcal{V}_{ji}|}{\|\mathbf{e}_{ji}\|_2^2}, \tag{2.11}$$

and the stiffness matrix in (2.9) is $\tilde{S}_{ji} = \mathbf{n}_{ji} \cdot \boldsymbol{\gamma}\, \mathbf{e}_{ji}|\partial\mathcal{V}_{ji}|/\|\mathbf{e}_{ji}\|_2^2$. The elements in $\tilde{\mathbf{S}}$ derived from (2.11) are always nonnegative, and hence the λ_{ji} in (2.7) defined by the FVM are nonnegative. The N components of $\mathbf{u}$ represent the average value of u in the voxels. However, the solution of (2.5) may not converge to the solutions of (2.4) when the mesh size h is reduced [44], since the approximation in (2.11) is consistent with $\gamma\Delta u$ only if the mesh is of Voronoi type [12]. But we can again interpret $\tilde{D}_{ij} = \tilde{S}_{ij}/|\mathcal{V}_i|$ as a consistent FEM discretization of the perturbed equation (2.10). The scheme in (2.11) is a vertex centered FVM. A cell centered FVM is used in [21] to define the jump coefficients.

In [31], the jump coefficients are chosen to be close to the FEM coefficients in (2.6). If D_{ij} is nonnegative, then λ_{ji} is as in (2.7). If $D_{ij} < 0$, then the λ_{ji} coefficients for voxel j are determined such that the mean first exit time ε_j from a vertex j in the mesh to the boundary $\partial\Omega$ is a solution of the system of linear equations

$$\tilde{\mathbf{D}}\boldsymbol{\varepsilon} = -\mathbf{e}, \quad \mathbf{e}^T = (1, 1, \ldots, 1), \tag{2.12}$$

where $\tilde{D}_{ij} = \lambda_{ji}|\mathcal{V}_j|/|\mathcal{V}_i|$. The mean first exit time is the expected time it takes for a molecule initially at a position inside Ω to reach $\partial\Omega$. With these coefficients in Algorithm 1, the average of the simulated first exit times from vertices in the mesh agree very well in [31] with those computed numerically with a FEM discretization of (2.12). This method of computing the jump coefficients is based on the global first exit time of the molecules and is denoted by GFET.

3. Analysis. Previously, we presented three methods that can be regarded as modifications of the FEM discretization matrix $\mathbf{D}$ into a matrix $\tilde{\mathbf{D}}$ with nonnegative jump coefficients. In this section we view $\tilde{\mathbf{D}}$ as a FEM discretization of a certain perturbed PDE (3.2) below. To quantify the error introduced by the change in the diffusion matrix, we therefore aim at bounding the difference between the solutions to the PDEs,

$$u_t = \gamma\Delta u, \tag{3.1}$$

$$\tilde{u}_t = \nabla \cdot (\tilde{\boldsymbol{\gamma}}\nabla\tilde{u}), \tag{3.2}$$

for $\mathbf{x} \in \Omega$, with homogeneous Neumann boundary conditions $\partial u/\partial n = \partial\tilde{u}/\partial n = 0$ for $\mathbf{x} \in \partial\Omega$, and initial data $u_0 = \tilde{u}_0$ at $t = 0$. Here $\tilde{\boldsymbol{\gamma}}(\mathbf{x})$ is a symmetric, *uniformly* positive definite matrix. The mean first exit time used to define the GFET algorithm fulfills Poisson's equation [36]

$$-1 = \gamma\Delta\varepsilon, \tag{3.3}$$

with the corresponding perturbed equation

$$-1 = \nabla \cdot (\tilde{\boldsymbol{\gamma}}\nabla\tilde{\varepsilon}) \tag{3.4}$$

and homogeneous Dirichlet boundary condition.

Let $H^1(\Omega)$ be the Hilbert space of all functions $u \in L^2(\Omega)$ for which the first weak derivative exists and lies in $L^2(\Omega)$. The corresponding weak problems for (3.1) and (3.2) are *find* $u, \tilde{u} \in H^1(\Omega)$ *such that* $\forall\, v \in H^1(\Omega)$,

$$(v, u_t) = -(\nabla v, \gamma \nabla u), \tag{3.5}$$

$$(v, \tilde{u}_t) = -(\nabla v, \tilde{\boldsymbol{\gamma}} \nabla \tilde{u}). \tag{3.6}$$

For finite element solutions in a finite-dimensional subspace $H_h^1(\Omega) \subset H^1(\Omega)$, we have the following set of equations (see also (2.5) and (2.9) above):

$$\mathbf{M}\mathbf{u}_t = \mathbf{S}\mathbf{u}, \tag{3.7}$$

$$\mathbf{M}\tilde{\mathbf{u}}_t = \tilde{\mathbf{S}}\tilde{\mathbf{u}}. \tag{3.8}$$

We shall require the following basic a priori estimates.

LEMMA 3.1. *For some* $C > 0$,

$$\|\nabla u\|_{L^2} \le \|\nabla u_0\|_{L^2}, \tag{3.9}$$

$$\|\nabla \tilde{u}\|_{L^2} \le C\|\nabla u_0\|_{L^2}, \tag{3.10}$$

where (3.10) *assumes that* $\tilde{\boldsymbol{\gamma}}$ *is uniformly positive definite such that* $g\|\mathbf{y}\|_2^2 \le \mathbf{y}^T\tilde{\gamma}(\mathbf{x})\mathbf{y} \le G\|\mathbf{y}\|_2^2$, *for some positive constants* (g, G), $\forall \mathbf{x} \in \Omega \subset \mathbb{R}^d$, $\forall \mathbf{y} \in \mathbb{R}^d$, *where* d *is the dimension.*

Proof. The case (3.9) of a scalar γ follows straightforwardly, so we focus on (3.10). Letting $v = \tilde{u}_t$ in (3.6), we arrive at

$$\|\tilde{u}_t\|_{L^2}^2 = -(\nabla \tilde{u}_t, \tilde{\boldsymbol{\gamma}} \nabla \tilde{u}) = -\frac{1}{2}\frac{d}{dt}(\nabla \tilde{u}, \tilde{\boldsymbol{\gamma}} \nabla \tilde{u}),$$

since $\tilde{\boldsymbol{\gamma}}$ is symmetric. Integrating, we get

$$(\nabla \tilde{u}, \tilde{\boldsymbol{\gamma}} \nabla \tilde{u}) \le (\nabla \tilde{u}_0, \tilde{\boldsymbol{\gamma}} \nabla \tilde{u}_0).$$

Invoking the definiteness of $\tilde{\boldsymbol{\gamma}}$, we arrive at

$$g\|\nabla \tilde{u}\|_{L^2}^2 \le G\|\nabla \tilde{u}_0\|_{L^2}^2. \qquad \square$$

We now bound the error in the two solutions u and $\tilde{u}$.

THEOREM 3.2. *For some constant* $C > 0$,

$$\|\tilde{u} - u\|_{L^2}^2 \le Ct\|\boldsymbol{\gamma} - \tilde{\boldsymbol{\gamma}}\|_\infty \|\nabla u_0\|_{L^2}^2, \tag{3.11}$$

where the norm of the difference of the diffusion rates is defined by

$$\|\boldsymbol{\gamma} - \tilde{\boldsymbol{\gamma}}\|_\infty := \max_{x \in \Omega} \|\boldsymbol{\gamma} - \tilde{\boldsymbol{\gamma}}\|_2. \tag{3.12}$$

Proof. Subtracting (3.5) from (3.6) and using $v = \tilde{u} - u$, we get

$$\begin{aligned}
\frac{1}{2}\frac{d}{dt}\|v\|_{L^2}^2 &= -\gamma\|\nabla v\|_{L^2}^2 + (\nabla v, (\boldsymbol{\gamma} - \tilde{\boldsymbol{\gamma}})\nabla \tilde{u}) \le \int_\Omega |\nabla v^T(\boldsymbol{\gamma} - \tilde{\boldsymbol{\gamma}})\nabla \tilde{u}| d\Omega \\
&\le \int_\Omega \|\boldsymbol{\gamma} - \tilde{\boldsymbol{\gamma}}\|_2 \|\nabla v\|_2 \|\nabla \tilde{u}\|_2 d\Omega \le \max_{x \in \Omega} \|\boldsymbol{\gamma} - \tilde{\boldsymbol{\gamma}}\|_2 \int_\Omega \|\nabla v\|_2 \|\nabla \tilde{u}\|_2 d\Omega \\
&\le \|\boldsymbol{\gamma} - \tilde{\boldsymbol{\gamma}}\|_\infty \|\nabla v\|_{L^2} \|\nabla \tilde{u}\|_{L^2} \le \|\boldsymbol{\gamma} - \tilde{\boldsymbol{\gamma}}\|_\infty (\|\nabla u\|_{L^2} + \|\nabla \tilde{u}\|_{L^2}) \|\nabla \tilde{u}\|_{L^2} \\
&\le C\|\boldsymbol{\gamma} - \tilde{\boldsymbol{\gamma}}\|_\infty \|\nabla u_0\|_{L^2}^2
\end{aligned}$$

using Lemma 3.1. The estimate (3.11) follows by integration of the inequality. ∎

This shows that the forward error $\|u-\tilde{u}\|_{L^2}$ is bounded. Using the maximum norm of the difference between the two diffusion constants as in (3.11) is, however, pessimistic, and instead we now use the mean value of $\|\boldsymbol{\gamma}-\tilde{\boldsymbol{\gamma}}(\mathbf{x})\|_2$ over Ω to bound the error in the solutions.

PROPOSITION 3.3. *For some constant $C>0$,*

$$\frac{d}{dt}\|\tilde{u}-u\|_{L^2}^2 \le C\|\boldsymbol{\gamma}-\tilde{\boldsymbol{\gamma}}\|_*(\|\nabla u\|_{L^4}+\|\nabla\tilde{u}\|_{L^4})\|\nabla\tilde{u}\|_{L^4}, \tag{3.13}$$

where

$$\|\boldsymbol{\gamma}-\tilde{\boldsymbol{\gamma}}\|_*^2 := \frac{1}{|\Omega|}\int_\Omega \|\boldsymbol{\gamma}-\tilde{\boldsymbol{\gamma}}\|_2^2\, d\Omega. \tag{3.14}$$

Proof. As previously, subtracting (3.5) from (3.6) and using $v=\tilde{u}-u$, we get

$$\frac{1}{2}\frac{d}{dt}\|v\|_{L^2}^2 \le \int_\Omega \|\boldsymbol{\gamma}-\tilde{\boldsymbol{\gamma}}\|_2\|\nabla v\|_2\|\nabla\tilde{u}\|_2 d\Omega \le \left(\int_\Omega \|\boldsymbol{\gamma}-\tilde{\boldsymbol{\gamma}}\|_2^2 d\Omega\right)^{\frac{1}{2}}\left(\int_\Omega \|\nabla v\|_2^2\|\nabla\tilde{u}\|_2^2 d\Omega\right)^{\frac{1}{2}}$$
$$\le \|\boldsymbol{\gamma}-\tilde{\boldsymbol{\gamma}}\|_*|\Omega|^{\frac{1}{2}}\left(d\int_\Omega \|\nabla v\|_4^2\|\nabla\tilde{u}\|_4^2 d\Omega\right)^{\frac{1}{2}} \le \|\boldsymbol{\gamma}-\tilde{\boldsymbol{\gamma}}\|_*|\Omega|^{\frac{1}{2}}d^{\frac{1}{2}}\|\nabla v\|_{L^4}\|\nabla\tilde{u}\|_{L^4}. \quad ∎$$

For $t\ge\delta>0$, the diffusion equations (3.1) and (3.2) smooth out irregularities in the initial data, so we can assume that $\|\nabla u\|_{L^4}$ and $\|\nabla\tilde{u}\|_{L^4}$ are bounded such that for some $C>0$,

$$\|\nabla u\|_{L^4}\le C \quad \text{and} \quad \|\nabla\tilde{u}\|_{L^4}\le C, \tag{3.15}$$

and hence by integration of (3.13),

$$\|\tilde{u}-u\|_{L^2}^2 \le Ct\|\boldsymbol{\gamma}-\tilde{\boldsymbol{\gamma}}\|_*. \tag{3.16}$$

In summary, Theorem 3.2 shows that the forward error can be bounded in terms of the difference $\|\boldsymbol{\gamma}-\tilde{\boldsymbol{\gamma}}\|_\infty$ and some factors that are independent of $\tilde{\boldsymbol{\gamma}}$ (assuming that $\tilde{\boldsymbol{\gamma}}$ is uniformly positive definite). Inequality (3.16) shows a sharper bound in terms of $\|\boldsymbol{\gamma}-\tilde{\boldsymbol{\gamma}}\|_*$ at the cost of factors that may depend on $\tilde{\boldsymbol{\gamma}}$. We take (3.16) as the basis for our further analysis, thus assuming essentially that C in (3.15) depends only mildly on $\tilde{\boldsymbol{\gamma}}$.

The following proposition proves that u and $\tilde{u}$ have identical steady states, which shows that the t-dependent estimates in Theorem 3.2 and (3.16) are pessimistic and give a relevant bound only for small t.

PROPOSITION 3.4. *For $t\to\infty$ the steady state solutions of* (3.1) *and* (3.2) *fulfill*

$$u_\infty = \tilde{u}_\infty = \|u_0\|_{L_1}/|\Omega|. \tag{3.17}$$

Proof. Using $v=\tilde{u}_\infty$ in (3.6), we have at the steady state that

$$0 = (\nabla\tilde{u}_\infty, \tilde{\boldsymbol{\gamma}}\nabla\tilde{u}_\infty).$$

By the positive definiteness of $\tilde{\boldsymbol{\gamma}}$ this means $\nabla\tilde{u}_\infty=0$, and hence $\tilde{u}_\infty$ is constant. A similar argument for u implies the same property, and, moreover, since u is a density, we can safely assume $u_0=\tilde{u}_0\ge 0$. Setting $v=1$ in (3.6), we conclude that

$$\tilde{u}_\infty|\Omega| = \int_\Omega \tilde{u}_\infty d\Omega = (\tilde{u}_\infty, 1) = (\tilde{u}_0, 1) = \|\tilde{u}_0\|_{L^1},$$

which also holds analogously for u_∞. ∎

Using $u_0 = \tilde{u}_0$, we have from Proposition 3.4 that, since $\|u - \tilde{u}\|_{L^2}$ is continuous in time, there exists a $t^\star \in (0, \infty)$, where the error reaches its maximum $\|u(t^\star) - \tilde{u}(t^\star)\|_{L^2} \geq \|u(t) - \tilde{u}(t)\|_{L^2} \, \forall t$.

We obtain a similar result for the error in Poisson's equations.

THEOREM 3.5. *Assume $\partial\Omega \in C^\infty$ and $\tilde{\boldsymbol{\gamma}}(x) \in C^\infty$; then for the weak solutions ε and $\tilde{\varepsilon}$ of the weak problems corresponding to* (3.3) *and* (3.4),

$$\|\varepsilon - \tilde{\varepsilon}\|_{L^2}^2 \leq C\|\boldsymbol{\gamma} - \tilde{\boldsymbol{\gamma}}\|_* \tag{3.18}$$

for some constant $C > 0$.

Proof. Choosing $v = \varepsilon - \tilde{\varepsilon}$ and subtracting the weak formulations for (3.3) and (3.4), we obtain, with $\|\mathbf{v}\|_{\boldsymbol{\gamma}}^2 = (\mathbf{v}, \boldsymbol{\gamma}\mathbf{v})$,

$$0 = \|\nabla v\|_{\tilde{\boldsymbol{\gamma}}}^2 + (\nabla v, (\boldsymbol{\gamma} - \tilde{\boldsymbol{\gamma}})\nabla\varepsilon);$$

using the same arguments as in the proof of Proposition 3.3 gives

$$\|\nabla v\|_{\tilde{\boldsymbol{\gamma}}}^2 \leq C\|\boldsymbol{\gamma} - \tilde{\boldsymbol{\gamma}}\|_* \|\nabla v\|_{L^4} \|\nabla\varepsilon\|_{L^4}.$$

By the Poincaré–Friedrich inequality and the positive definiteness of $\tilde{\boldsymbol{\gamma}}$,

$$\|v\|_{L^2} \leq Cg^{-1}\|\boldsymbol{\gamma} - \tilde{\boldsymbol{\gamma}}\|_* \|\nabla v\|_{L^4} \|\nabla\varepsilon\|_{L^4}.$$

By [11, Chap. 6.3, Thm. 6] and the assumptions, we obtain $\varepsilon \in C^\infty(\bar{\Omega})$ and $\tilde{\varepsilon} \in C^\infty(\bar{\Omega})$. This bounds both $\|\nabla\varepsilon\|_\infty$ and $\|\nabla\tilde{\varepsilon}\|_\infty$, and hence $\|\nabla\varepsilon\|_{L^4} \leq C$ and $\|\nabla\tilde{\varepsilon}\|_{L^4} \leq C$. ∎

We conclude that we can effectively bound both the forward error $\|u - \tilde{u}\|_{L^2}$ and the error in the mean first exit time $\|\varepsilon - \tilde{\varepsilon}\|_{L^2}$ by the difference $\|\boldsymbol{\gamma} - \tilde{\boldsymbol{\gamma}}\|_*$. In the following section we present an algorithm for calculating this quantity for a given discretization matrix $\tilde{\mathbf{D}}$.

4. Backward analysis. In section 2, we presented the FVM and the modified FEM to compute the stiffness matrix $\tilde{\mathbf{S}}$ leading to nonnegative jump coefficients in (2.7). This matrix can be interpreted as the FEM matrix of a standard, convergent discretization of the perturbed equation (3.2). The general diffusion matrix $\tilde{\boldsymbol{\gamma}}(x)$ is symmetric and positive definite and may have nonzero off-diagonal elements. The difference between $\boldsymbol{\gamma}$ and $\tilde{\boldsymbol{\gamma}}$ should be as small as possible. This is a measure of how close the jump coefficients are to modeling stochastic diffusion, which converges to isotropic diffusion with a constant γ.

4.1. The FEM discretization. Interpreting $\tilde{\mathbf{S}}$ as the standard FEM stiffness matrix to the perturbed equation (3.2) implies that

$$\tilde{S}_{ij} = -(\nabla\psi_i, \tilde{\boldsymbol{\gamma}}(\mathbf{x})\nabla\psi_j) \tag{4.1}$$

$\forall$ edges e_{ij}. The sparsity pattern of $\mathbf{S}$ and $\tilde{\mathbf{S}}$ is the same and is determined by the connectivity of the mesh. Here ψ_i and ψ_j are the hat functions of linear Lagrangian finite elements with $\psi_i(\mathbf{x}_i) = 1$ and $\psi_i(\mathbf{x}_j) = 0$ when $i \neq j$. Since the right-hand side of (4.1) is a symmetric expression in i and j, the perturbed stiffness matrix $\tilde{\mathbf{S}}$ has to be symmetric. The FVM and nnFEM generate symmetric stiffness matrices. To symmetrize $\tilde{\mathbf{S}}$ resulting from GFET, we use its symmetric part $(\tilde{\mathbf{S}} + \tilde{\mathbf{S}}^T)/2$ as $\tilde{\mathbf{S}}$ in

the following. The boundary $\partial\Omega$ of the domain Ω is assumed to be polygonal, and Ω is discretized such that

$$\Omega = \bigcup_{T_k \in \mathcal{T}} T_k, \tag{4.2}$$

where $\mathcal{T}$ is the set of all nonoverlapping elements T_k. These elements are triangles in two dimensions and tetrahedra in three dimensions in the primal mesh on Ω defined by the edges e_{ij}; see Figure 1. The dual mesh on Ω defines the voxels $\mathcal{V}_i$ in section 2. With $\mathcal{T}_{ij} \subset \mathcal{T}$ being the set of all triangles in two dimensions or tetrahedra in three dimensions containing edge e_{ij}, we can write (4.1) as

$$\begin{aligned} \tilde{S}_{ij} &= -\sum_{T_k \in \mathcal{T}_{ij}} \int_{T_k} \nabla\psi_i^T \tilde{\boldsymbol{\gamma}}(\mathbf{x}) \nabla\psi_j d\mathbf{x} = -\sum_{T_k \in \mathcal{T}_{ij}} \nabla\psi_i^T\big|_{T_k} \int_{T_k} \tilde{\boldsymbol{\gamma}} d\mathbf{x} \nabla\psi_j\big|_{T_k} \\ &= -\sum_{T_k \in \mathcal{T}_{ij}} \nabla\psi_i^T\big|_{T_k} \tilde{\boldsymbol{\gamma}}_k \nabla\psi_j\big|_{T_k} |T_k|, \end{aligned} \tag{4.3}$$

since the gradients are constant in T_k. It is only the average $\tilde{\boldsymbol{\gamma}}_k$ of $\tilde{\boldsymbol{\gamma}}(x)$ on each element T_k that contributes to $\tilde{S}_{ij}$. Thus, we calculate $\tilde{\boldsymbol{\gamma}}_k$ of the following type in two and three dimensions, respectively:

$$\tilde{\boldsymbol{\gamma}}_k^{2D} = \begin{pmatrix} \tilde{\gamma}_{k1} & \tilde{\gamma}_{k3} \\ \tilde{\gamma}_{k3} & \tilde{\gamma}_{k2} \end{pmatrix}, \quad \tilde{\boldsymbol{\gamma}}_k^{3D} = \begin{pmatrix} \tilde{\gamma}_{k1} & \tilde{\gamma}_{k4} & \tilde{\gamma}_{k5} \\ \tilde{\gamma}_{k4} & \tilde{\gamma}_{k2} & \tilde{\gamma}_{k6} \\ \tilde{\gamma}_{k5} & \tilde{\gamma}_{k6} & \tilde{\gamma}_{k3} \end{pmatrix}. \tag{4.4}$$

With

$$\nabla\psi_i^{2D} = \begin{pmatrix} \nabla\psi_{i1} \\ \nabla\psi_{i2} \end{pmatrix}, \quad \nabla\psi_i^{3D} = \begin{pmatrix} \nabla\psi_{i1} \\ \nabla\psi_{i2} \\ \nabla\psi_{i3} \end{pmatrix}, \tag{4.5}$$

and the coefficients C_{ijkl} and L as in Table 1, for each edge e_{ij}, (4.1) becomes

$$\tilde{S}_{ij} = \sum_{T_k \in \mathcal{T}_{ij}} \sum_{l=1}^{L} C_{ijkl} \tilde{\gamma}_{kl}. \tag{4.6}$$

TABLE 1
Coefficients in (4.6).

	Two dimensions	Three dimensions
L	3	6
C_{ijk1}	$-\nabla\psi_{i1}\nabla\psi_{j1}$	$-\nabla\psi_{i1}\nabla\psi_{j1}$
C_{ijk2}	$-\nabla\psi_{i2}\nabla\psi_{j2}$	$-\nabla\psi_{i2}\nabla\psi_{j2}$
C_{ijk3}	$-(\nabla\psi_{i1}\nabla\psi_{j2} + \nabla\psi_{i2}\nabla\psi_{j1})$	$-\nabla\psi_{i3}\nabla\psi_{j3}$
C_{ijk4}		$-(\nabla\psi_{i1}\nabla\psi_{j2} + \nabla\psi_{i2}\nabla\psi_{j1})$
C_{ijk5}		$-(\nabla\psi_{i1}\nabla\psi_{j3} + \nabla\psi_{i3}\nabla\psi_{j1})$
C_{ijk6}		$-(\nabla\psi_{i2}\nabla\psi_{j3} + \nabla\psi_{i3}\nabla\psi_{j2})$

In two dimensions, the integrand in (4.1) is nonzero on two triangles and on at least three tetrahedra in three dimensions for the edges in the interior of Ω. One can show using induction that a 2D mesh with N vertices and E_B edges at the boundary

has $T = 2N - 2 - E_B$ triangles and $E = 3N - 3 - E_B$ edges. Taking into account that there are three unknowns per triangle in (4.4) and one equation (4.6) per edge, we have to solve an underdetermined system for any triangulation containing more than one triangle with $3N - 3 - 2E_B$ remaining degrees of freedom.

In three dimensions, the system of linear equations defined by (4.6) is also underdetermined if the mesh consists of more than one tetrahedron. Each edge in the mesh is an edge of at least one tetrahedron, but there may be only one tetrahedron associated with the edge on the boundary. Then the number of unknowns is six in (4.4) and the number of linear constraints of the form (4.6) is six. For each additional tetrahedron sharing the same edge, there are six new unknowns and three new constraints. The total number of unknowns for each edge is $6T$, where T is the number of tetrahedra with a common edge and the number of linear constraints is $3T + 3$. Locally, the diffusion matrix $\tilde{\boldsymbol{\gamma}}$ is underdetermined with $3T - 3$ degrees of freedom.

Consequently, the diffusion $\tilde{\boldsymbol{\gamma}}$ satisfying (4.6) is not unique, but for all possible $\tilde{\boldsymbol{\gamma}}$ the error analysis in section 3 holds. We obtain the sharpest bounds on $\|u - \tilde{u}\|_{L^2}$ and $\|\varepsilon - \tilde{\varepsilon}\|_{L^2}$ by finding $\tilde{\boldsymbol{\gamma}}$ satisfying (4.6) and minimizing the difference $\|\boldsymbol{\gamma} - \tilde{\boldsymbol{\gamma}}\|_*$ in the equations. An alternative would be to replace $\|\cdot\|_2$ in (3.14) by the Frobenius norm $\|\cdot\|_F$. Since

$$\|\mathbf{A}\|_2 \le \|\mathbf{A}\|_F \le \sqrt{d}\|\mathbf{A}\|_2 \tag{4.7}$$

for a matrix $\mathbf{A}$ [18], the bound in section 3 is sharper if the minimization is made in the $\|\cdot\|_2$ norm. In the following, we propose a global and a local optimization procedure to find these minimizers $\tilde{\boldsymbol{\gamma}}$.

4.2. Global optimization. The diffusion matrix $\tilde{\boldsymbol{\gamma}}$ closest to the original diffusion $\boldsymbol{\gamma}$ with constant coefficient γ is found by minimizing the distance between $\tilde{\boldsymbol{\gamma}}(\mathbf{x})$ and $\boldsymbol{\gamma}$ under the constraints in (4.6). The stiffness matrix $\tilde{\mathbf{S}}$ is given by (2.7) and one of the methods in section 2. As only the average $\tilde{\boldsymbol{\gamma}}_k$ of $\tilde{\boldsymbol{\gamma}}(\mathbf{x})$ appears in the FEM approximation on each triangle T_k (see (4.3)), the norm of the difference in diffusion in (3.14) reduces to the weighted sum of the differences $\|\tilde{\boldsymbol{\gamma}}_k - \boldsymbol{\gamma}\|_2^2$ as a measure of the distance resulting in the following optimization problem:

$$\min_{\tilde{\boldsymbol{\gamma}}_k} \sum_{T_k \in \mathcal{T}} |T_k| \|\tilde{\boldsymbol{\gamma}}_k - \boldsymbol{\gamma}\|_2^2, \tag{4.8}$$

$$\sum_{T_k \in \mathcal{T}_{ij}} \sum_{l=1}^{L} C_{ijkl} \tilde{\gamma}_{kl} = \tilde{S}_{ij} \quad \forall e_{ij}. \tag{4.9}$$

This is a nonlinear programming problem with $3T$ variables in two dimensions, $6T$ variables in three dimensions, and E linear constraints.

The difference $\|\tilde{\boldsymbol{\gamma}}_k - \boldsymbol{\gamma}\|_2^2$ is a convex function in the unknowns in $\tilde{\boldsymbol{\gamma}}_k$. Hence, the objective function in (4.8) is a convex function too. Since also the constraint set in (4.9) is convex, the local solution to (4.8) and (4.9) is the unique global optimum. If $S_{ij} \ge 0 \,\forall\, i, j$ from the FEM discretization with diffusion constant γ, then $\tilde{S}_{ij} = S_{ij} \,\forall\, i, j$, and the solution to (4.8) is $\tilde{\boldsymbol{\gamma}}_k = \boldsymbol{\gamma} \,\forall\, T_k$.

The mean value matrix $\tilde{\boldsymbol{\gamma}}_k$ defines two (three) main axes in two (three) dimensions on T_k. Let the columns of $\mathbf{V}$ be the eigenvectors $\mathbf{v}_j$ of $\tilde{\boldsymbol{\gamma}}_k$ with eigenvalues λ_j. After

a coordinate transformation from $\mathbf{x}$ to $\mathbf{y}$ with $\mathbf{x} = \mathbf{V}\mathbf{y}$, the diffusion term is

$$\nabla_{\mathbf{x}} \cdot (\tilde{\boldsymbol{\gamma}} \nabla_{\mathbf{x}} u) = \sum_j \lambda_j \frac{\partial^2 u}{\partial v_j^2}. \tag{4.10}$$

The eigenvectors define the main axes of the diffusion, and the diffusion speed along those axes is given by the eigenvalues of $\tilde{\boldsymbol{\gamma}}$. Since

$$\|\tilde{\boldsymbol{\gamma}}_k - \boldsymbol{\gamma}\|_2 = \max_j |\lambda_j - \gamma|, \tag{4.11}$$

the ℓ_2 norm in (4.8) measures the maximum deviation in speed of the diffusion in $\tilde{\boldsymbol{\gamma}}$ compared to γ weighted by the size of T_k. In the Frobenius norm,

$$\|\tilde{\boldsymbol{\gamma}}_k - \boldsymbol{\gamma}\|_F = \left(\sum_{j=1}^{d} (\lambda_j - \gamma)^2 \right)^{1/2}, \tag{4.12}$$

and the norm is equal to the ℓ_2 norm of the difference in diffusion speed in all directions. The objective function in (4.8) is continuous in $\tilde{\boldsymbol{\gamma}}$ but not continuously differentiable everywhere.

4.3. Local optimization. The optimization problem in the previous section may be computationally expensive but is simplified if we approach the solution of (4.9) by local optimization. Let $\mathcal{E}_{ij}$ be defined by

$$\mathcal{E}_{ij} = \{e_{mn} : \ e_{mn} \text{ is an edge of any } T_k \in \mathcal{T}_{ij}\}. \tag{4.13}$$

The adjacent $\tilde{\boldsymbol{\gamma}}_k$ in T_k in $\mathcal{T}_{ij}$ for each edge e_{ij} is optimized, while keeping $\tilde{S}_{ij}$ constant on the other edges in $\mathcal{E}_{ij}$. Update $\tilde{\boldsymbol{\gamma}}_k$ with the most recently computed diffusion matrix. Then iterate over all edges once. Still, the underdetermined system (4.9) will be satisfied, but with a different $\tilde{\boldsymbol{\gamma}}^L$ compared to $\tilde{\boldsymbol{\gamma}}^G$ solving (4.8). The algorithm is as follows.

Algorithm 2 Local optimization I.

1: $\tilde{\boldsymbol{\gamma}}_k = \boldsymbol{\gamma}\ \forall T_k \in \mathcal{T}$
2: **for all** e_{ij} **do**
3: Solve

$$\begin{array}{ll} \min_{\tilde{\boldsymbol{\gamma}}_k^{new}} & \sum_{T_k \in \mathcal{T}_{ij}} |T_k| \|\tilde{\boldsymbol{\gamma}}_k^{new} - \boldsymbol{\gamma}\|_2^2 \\ & \sum_{T_k \in \mathcal{T}_{ij}} \sum_{l=1}^{L} C_{ijkl} \tilde{\gamma}_{kl}^{new} \quad = \tilde{S}_{ij}, \\ & \sum_{l=1}^{L} C_{mnkl} \tilde{\gamma}_{kl}^{new} \quad = \sum_{l=1}^{L} C_{mnkl} \tilde{\gamma}_{kl} \\ & \qquad \forall e_{mn} \in \mathcal{E}_{ij} \setminus e_{ij},\ \forall T_k \in \mathcal{T}_{ij} \end{array}$$

4: $\tilde{\boldsymbol{\gamma}}_k = \tilde{\boldsymbol{\gamma}}_k^{new}$, $T_k \in \mathcal{T}_{ij}$
5: **end for**

The diffusion $\tilde{\boldsymbol{\gamma}}_k$ changes successively only on the elements adjacent to e_{ij} (two triangles in two dimensions and at least three tetrahedra in three dimensions in the interior) in each iterative step. At each inner edge in two dimensions, there are six variables and five constraints. As remarked in section 4.1 above, the number of

variables in three dimensions in Algorithm 2 is $6T$ and the number of constraints is $3T + 3$, where T is the number of tetrahedra sharing the common edge e_{ij}. For a boundary edge, the number of unknowns equals the number of constraints in two dimensions, and one has to solve only the linear system in (4.6).

Also, here we have that if $\tilde{S}_{ij} = S_{ij} \geq 0 \, \forall \, i, j$, then the solution is $\tilde{\boldsymbol{\gamma}}_k = \boldsymbol{\gamma}$. The order in which the edges are traversed matters for the result of the algorithm, but when all edges have been visited, (4.9) is satisfied. In the numerical experiments in section 6, the order is random but other choices are possible.

The global $\tilde{\boldsymbol{\gamma}}_k^G$ from (4.8) and the local $\tilde{\boldsymbol{\gamma}}_k^L$ from Algorithm 2 fulfill

$$\eta_2^G = \sqrt{\frac{1}{|\Omega|} \sum_{T_k \in \mathcal{T}} |T_k| \|\tilde{\boldsymbol{\gamma}}_k^G - \boldsymbol{\gamma}\|_2^2} \leq \sqrt{\frac{1}{|\Omega|} \sum_{T_k \in \mathcal{T}} |T_k| \|\tilde{\boldsymbol{\gamma}}_k^L - \boldsymbol{\gamma}\|_2^2} = \eta_2^L, \tag{4.14}$$

since $\tilde{\boldsymbol{\gamma}}_k^G$ is the global minimum solution.

Neither the global nor the local procedure for determining $\tilde{\boldsymbol{\gamma}}$ guarantees its positive definiteness when the solution is computed with an optimization algorithm for a nonlinear objective function with linear constraints. Extra nonlinear constraints can be added to enforce positive definiteness. That leads to slow algorithms or sometimes very large backward errors $\|\boldsymbol{\gamma} - \tilde{\boldsymbol{\gamma}}_k\|_2$ in the numerical experiments in section 6. An alternative would be to apply a computationally more expensive semidefinite programming algorithm [2, 47] to the problem. In section 6, we first compute $\tilde{\boldsymbol{\gamma}}$ without constraints for positive definiteness and then check the solution for positive definiteness. The nonlinear programming algorithm finds positive definite $\tilde{\boldsymbol{\gamma}}_k$ for all elements in most cases.

The diffusion $\tilde{\boldsymbol{\gamma}}$ is computed for a given mesh of finite mesh size h. What happens with $\tilde{\boldsymbol{\gamma}}$ when the mesh is refined depends on the mesh generator. If all S_{ij} become nonnegative as $h \to 0$, then $\tilde{\boldsymbol{\gamma}} \to \boldsymbol{\gamma}$. Otherwise, there will be a difference $\|\boldsymbol{\gamma} - \tilde{\boldsymbol{\gamma}}\|_*$ of $\mathcal{O}(1)$ as h vanishes.

The backward analysis is extended in the next section to the design of the stiffness matrix $\tilde{\mathbf{S}}$ such that the backward error $\|\boldsymbol{\gamma} - \tilde{\boldsymbol{\gamma}}\|_*$ is minimized.

5. Design. In the previous section, we described how to analyze existing methods for creating positive jump coefficients by backward analysis. In this section we determine a new discretization using FEM by minimizing the backward error. We devise nonnegative jump coefficients such that the perturbed diffusion $\tilde{\boldsymbol{\gamma}}$ is as close as possible to the original diffusion with a constant γ. The connectivity of the network of edges is the same as in section 4.1 but $\tilde{S}_{ij}$ is free to vary. Molecules in the stochastic setting are allowed to jump only to the neighboring voxels, but the rate is a free variable to be optimized such that the distribution of molecules converges to the diffusion equation (3.2) in the limit of large molecules numbers.

5.1. Global optimization. The diffusion $\tilde{\boldsymbol{\gamma}}_k$ in each triangle or tetrahedron is determined such that

$$\min_{\tilde{\boldsymbol{\gamma}}_k} \sum_{T_k \in \mathcal{T}} |T_k| \|\tilde{\boldsymbol{\gamma}}_k - \boldsymbol{\gamma}\|_2^2, \tag{5.1}$$

$$\sum_{T_k \in \mathcal{T}_{ij}} \sum_{l=1}^{L} C_{ijkl} \tilde{\gamma}_{kl} \; \geq 0 \quad \forall e_{ij}. \tag{5.2}$$

The equality constraints in (4.9) are replaced by the inequalities in (5.2). The new jump coefficients λ_{ji} are computed by the optimal $\tilde{\boldsymbol{\gamma}}$ and $\tilde{S}_{ij}$,

$$\tilde{S}_{ij} = \sum_{T_k \in \mathcal{T}_{ij}} \sum_{l=1}^{L} C_{ijkl} \tilde{\gamma}_{kl}, \quad i \neq j, \tag{5.3}$$

inserted into (2.7). The stiffness matrix is thus obtained from the FEM discretization of the diffusion term in (2.10) with linear Lagrangian elements and diffusion matrix $\tilde{\boldsymbol{\gamma}}_k$ on T_k.

5.2. Local optimization. The local optimization algorithm in section 4.3 to analyze given jump coefficients is modified in the same way to generate new coefficients instead. For each edge, the adjacent diffusion matrices are computed such that they are close to $\boldsymbol{\gamma}$ and the nonnegativity constraint is satisfied for the edges. Instead of keeping the contribution to the other edges constant, we let it vary constrained by nonnegativity. If e_{mn} is an edge in $\mathcal{E}_{ij} \setminus e_{ij}$, then we allow $\tilde{\boldsymbol{\gamma}}$ to be such that

$$\sum_{T_k \in \mathcal{T}_{mn}} \sum_{l=1}^{L} C_{mnkl} \tilde{\gamma}_{kl} = \tilde{S}_{mn} \geq 0. \tag{5.4}$$

In each local optimization step, $\tilde{\boldsymbol{\gamma}}_k$ in the elements T_k adjacent to edge e_{ij} are modified while keeping $\tilde{S}_{mn}$ in other edges in $\mathcal{E}_{ij}$ nonnegative. Splitting the sum in (5.4) into two parts, we have

$$\sum_{T_k \in \mathcal{T}_{ij} \cap \mathcal{T}_{mn}} \sum_{l=1}^{L} C_{mnkl} \tilde{\gamma}_{kl} \geq - \sum_{T_k \in \mathcal{T}_{mn} \setminus (\mathcal{T}_{ij} \cap \mathcal{T}_{mn})} \sum_{l=1}^{L} C_{mnkl} \tilde{\gamma}_{kl} \tag{5.5}$$

$$= \sum_{T_k \in \mathcal{T}_{ij} \cap \mathcal{T}_{mn}} \sum_{l=1}^{L} C_{mnkl} \tilde{\gamma}_{kl} - \tilde{S}_{mn}. \tag{5.6}$$

The diffusion matrix on the left-hand side of (5.5) is updated given the diffusion matrix in the right-hand side in (5.6). This is repeated successively for all edges in the following algorithm.

Algorithm 3 Local optimization II.

1: $\tilde{\boldsymbol{\gamma}}_k = \boldsymbol{\gamma}\ \forall T_k \in \mathcal{T}$
2: **for all** e_{ij} **do**
3: $\quad \tilde{S}_{ij} = \sum_{T_k \in \mathcal{T}_{ij}} \sum_{l=1}^{L} C_{ijkl} \tilde{\gamma}_{kl}$
4: **end for**
5: **for all** e_{ij} **do**
6: $\quad$ Solve

$$\begin{aligned} \min_{\tilde{\boldsymbol{\gamma}}_k^{new}} \quad & \textstyle\sum_{T_k \in \mathcal{T}_{ij}} |T_k| \|\tilde{\boldsymbol{\gamma}}_k^{new} - \boldsymbol{\gamma}\|_2^2 \\ & \textstyle\sum_{T_k \in \mathcal{T}_{ij}} \sum_{l=1}^{L} C_{ijkl} \tilde{\gamma}_{kl}^{new} && \geq 0, \\ & \textstyle\sum_{T_k \in \mathcal{T}_{ij} \cap \mathcal{T}_{mn}} \sum_{l=1}^{L} C_{mnkl} \tilde{\gamma}_{kl}^{new} && \geq \textstyle\sum_{T_k \in \mathcal{T}_{ij} \cap \mathcal{T}_{mn}} \sum_{l=1}^{L} C_{mnkl} \tilde{\gamma}_{kl} - \tilde{S}_{mn} \\ & && \forall e_{mn} \in \mathcal{E}_{ij} \setminus e_{ij},\ \forall T_k \in \mathcal{T}_{ij} \end{aligned}$$

7: $\quad \tilde{\boldsymbol{\gamma}}_k = \tilde{\boldsymbol{\gamma}}_k^{new},\ T_k \in \mathcal{T}_{ij}$
8: $\quad \tilde{S}_{ij} = \sum_{T_k \in \mathcal{T}_{ij}} \sum_{l=1}^{L} C_{ijkl} \tilde{\gamma}_{kl}$
9: **end for**

The number of optimization problems to be solved in Algorithm 3 is the number of edges E which is bounded by a constant times the number of vertices N in a mesh. The size of each optimization problem is independent of E and N. Hence, the computational work is proportional to N in the local optimization and of the same computational complexity as the matrix assembly of $\mathbf{S}$. The edges are traversed in a random order in the experiments in section 6. When all edges have been visited once, $\tilde{S}_{ij}$ satisfies (5.3) and $\tilde{S}_{ij} \geq 0$, and the new λ_{ji} is computed using (2.7).

5.3. Practical implementation. The local minimization problem contains only the adjacent triangles or tetrahedra and is hence faster to compute, but $\eta_2^L > \eta_2^G$; see (4.14). Instead of running the local Algorithms 2 and 3 only once, we can repeat them iteratively with the results $\tilde{\boldsymbol{\gamma}}_k$ of the previous iteration as the initial guess for the next minimization. Then η_2^L will approach η_2^G.

A possible way to speed up the computation is to replace the ℓ_2 norm of the error by the Frobenius norm,

$$\eta_F = \sqrt{\frac{1}{|\Omega|} \sum_{T_k \in \mathcal{T}} |T_k| \|\tilde{\boldsymbol{\gamma}}_k - \boldsymbol{\gamma}\|_F^2}. \tag{5.7}$$

The global nonlinear minimization problem (4.8) then simplifies to the quadratic programming problem

$$\min_{\bar{\gamma}} \ \bar{\boldsymbol{\gamma}}^T \mathbf{H} \bar{\boldsymbol{\gamma}} - 2\mathbf{f}^T \bar{\boldsymbol{\gamma}}, \tag{5.8}$$

$$\sum_{T_k \in \mathcal{T}_{ij}} \sum_{l=1}^{L} C_{ijkl} \tilde{\gamma}_{kl} \geq 0 \quad \forall e_{ij}, \tag{5.9}$$

where $\mathbf{H}$ is a diagonal matrix with positive elements on the diagonal and $\bar{\boldsymbol{\gamma}}$ is a vector with $\tilde{\gamma}_{kl}$ as components. By the relation between the ℓ_2 and Frobenius norms (4.7), the resulting η_F yields only an upper bound on the global minimum η_2^G. In the local optimizations in Algorithms 2 and 3, $\|\cdot\|_2^2$ is then substituted by $\|\cdot\|_F^2$.

We can further reduce the computational complexity by rewriting the high-dimensional minimization problem (5.8) as the smaller dual problem,

$$\min_{\boldsymbol{\mu} \geq 0} \boldsymbol{\mu}^T \tilde{\mathbf{H}} \boldsymbol{\mu} + 2\tilde{\mathbf{f}}^T \boldsymbol{\mu}, \tag{5.10}$$

where $\boldsymbol{\mu} \geq 0$ is equivalent to $\mu_i \geq 0 \, \forall i$, and

$$\tilde{\mathbf{H}} = \mathbf{C}\mathbf{H}^{-1}\mathbf{C}^T, \qquad \tilde{\mathbf{f}} = \mathbf{C}\mathbf{H}^{-1}\mathbf{f}, \qquad \boldsymbol{\gamma} = -\mathbf{H}^{-1}(\mathbf{C}^T\boldsymbol{\mu} - \mathbf{f}). \tag{5.11}$$

In (5.11), $\mathbf{C}$ is such that (5.9) is replaced by $\mathbf{C}\bar{\boldsymbol{\gamma}} \geq 0$. The primal problem of dimension $3T$ is hence reduced by approximately a factor of two to the dual problem of dimension E in two dimensions. In three dimensions the dual problem is more than a factor of 4.5 smaller than the primal problem in the numerical experiments in section 6. The interior point algorithm is well suited for the quadratic programming problems (5.8) and (5.10); see, e.g., [2, 25].

5.4. Alternatives to determine a nonnegative $\tilde{\mathbf{S}}$. Another two possibilities are investigated to calculate an $\tilde{\mathbf{S}}$ with only nonnegative off-diagonal entries. From the set of discrete equations (2.5) it appears that a smaller difference between the discretization matrices $\mathbf{D}$ and $\tilde{\mathbf{D}}$ leads to a smaller error in the solution $\|\mathbf{u} - \tilde{\mathbf{u}}\|_{L^2}$. That

suggests finding an $\tilde{\mathbf{S}}$ with the same sparsity pattern as $\mathbf{S}$ but with only nonnegative entries such that $\|\mathbf{D} - \tilde{\mathbf{D}}\|_2$ is minimized.

A second alternative to guarantee nonnegative jump coefficients is adding artificial viscosity to the system. The same viscosity is added patchwise in all elements with a common vertex. If edge e_{ij} corresponds to a negative entry, then enough viscosity to eliminate the negative entry S_{ij} is added to all edges originating from $\mathbf{x}_i$ and $\mathbf{x}_j$ as in the graph Laplacian. The symmetry of the original matrix $\mathbf{S}$ is preserved by adding $|S_{ij}|/2$ to the nodes around $\mathbf{x}_i$ and $\mathbf{x}_j$ in the following way:

$$\tilde{S}_{ik} = S_{ik} + |S_{ij}|/2, \qquad \tilde{S}_{ki} = S_{ki} + |S_{ij}|/2 \quad \forall \mathbf{x}_k \text{ connected to } \mathbf{x}_i \text{ by } e_{ki},$$
$$\tilde{S}_{jk} = S_{jk} + |S_{ij}|/2, \qquad \tilde{S}_{kj} = S_{kj} + |S_{ij}|/2 \quad \forall \mathbf{x}_k \text{ connected to } \mathbf{x}_j \text{ by } e_{kj}.$$

This is a generalization of the nnFEM approach, where a sufficient amount of viscosity is added only to the negative edge. This type of artificial viscosity is introduced in [19] to prove that the maximum principle is satisfied for a conservation law.

6. Numerical experiments. In this section, we determine numerically the local η^L and global η^G backward errors in (4.14) for the different methods generating nonnegative coefficients as described in sections 2 and 5 with a diffusion coefficient $\gamma = 1$. By the analysis in section 3, the backward error bounds the forward error of the mean values in the spatial distribution of the copy numbers of the molecules $\|u - \tilde{u}\|_{L^2}$ in (3.13) and the exit times $\|\varepsilon - \tilde{\varepsilon}\|_{L^2}$ in (3.18). All computations are done in MATLAB using its optimization routines. The meshes are generated by COMSOL Multiphysics, and the FEM matrices are assembled by the same software.

6.1. Diffusion in two dimensions. The square $[-0.5, 0.5] \times [-0.5, 0.5]$ is discretized into 227 nodes; see Figure 2. As mentioned in section 2, mesh generators usually produce good quality meshes in two dimensions, and the mesh in Figure 2 is intentionally perturbed to obtain 47 edges with negative jump coefficients (marked as thick lines in Figure 2).

The requirement to obtain a nonnegative discretization poses other constraints on the mesh than what is necessary for a FEM solution of high accuracy. Examples of quality measures $\mathcal{Q}$ related to errors in the finite element solution of Poisson's equation are found in [43]. In two dimensions, let h_1, h_2, h_3 be the lengths of the edges of a triangle of area A with the angle φ_3 opposing the edge of maximum length h_3. A bound on the error in the gradient between f and the approximating f_h in the triangle is in [43],

$$\|\nabla f - \nabla f_h\|_\infty \le c_f \frac{3h_1h_2h_3}{2A} = c_f \frac{3h_3}{2\sin(\varphi_3)} = c_f \frac{1}{\mathcal{Q}}, \tag{6.1}$$

where c_f is a bound on the second derivatives of f. The measure $\mathcal{Q}$ is positive and should be as large as possible. Suppose that two triangles with the same edge lengths have the edge e_{ij} of length h_3 in common. Then S_{ij} in (2.8) is negative when $\varphi_3 > \pi/2$, while the estimate in (6.1) is as small as possible when φ_3 is in the neighborhood of $\pi/2$. On the other hand, the accuracy is poor if h_3 is large and all angles are less than $\pi/2$. Then $\mathcal{Q}$ is small but the jump coefficients are positive. A large h_3 will of course also affect the spatial resolution of the stochastic simulations, but the diffusion propensities are well defined.

6.1.1. Backward analysis. We compare the FVM, the symmetrized GFET method, the nnFEM, and the method minimizing the backward error (MBE). These

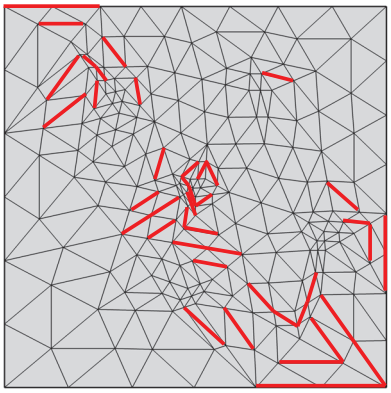


FIG. 2. *The mesh in two dimensions. Negative edges are shown as thick lines.*

methods all produce discretization matrices approximating the Laplacian with only nonnegative off-diagonal entries. The experiments are carried out for the mesh in Figure 2. In Figure 3, the local backward error $e = \|\tilde{\boldsymbol{\gamma}}_k - \boldsymbol{\gamma}\|_2$ is plotted for all triangles $T_k \in \mathcal{T}$, calculated by the local and global minimizations in Algorithms 2 and 3 and in (4.8), (4.9) and (5.1), (5.2).

In Table 2 we show the error

$$\eta_2 = \sqrt{\sum_{T_k \in \mathcal{T}} |T_k| \|\tilde{\boldsymbol{\gamma}}_k - \boldsymbol{\gamma}\|_2^2 / |\Omega|}, \tag{6.2}$$

where η_2 is either η_2^L or η_2^G according to (4.14). For the global MBE, the matrices $\tilde{\boldsymbol{\gamma}}_k$ are first calculated by nnFEM and then used as an initial guess for the MBE optimization. The minima are computed by the MATLAB `fmincon` with the active-set algorithm.

The local calculation of the backward error is more pessimistic than the global one in the table as expected from (4.14), but the ranking of the different methods is the same for both η_2^L and η_2^G. The FVM naturally leading to nonnegative jump coefficients causes the largest backward error when used on a poor mesh. A partial explanation of the FVM results may be that the jump coefficients are generated by a different principle than FEM. The fluxes over the element boundaries are approximated in FVM, and the method is forced into the framework of FEM. On four triangles, a discretization with FVM even leads to a negative definite diffusion matrix $\tilde{\boldsymbol{\gamma}}_k$ when calculated locally without the positive definiteness constraint. The GFET and nnFEM perform comparably for the mesh in Figure 2. Computing $\tilde{\mathbf{D}}$ for GFET is slightly more expensive than setting the negative off-diagonal entries to 0 in nnFEM. However, contrary to the nnFEM, the GFET preserves the exit time property of the original diffusion; see [31]. The minimization constrained by inequalities to obtain the discretization matrices with MBE improves the introduced backward error substantially. The faster local and slower global minimization algorithms yield similar results for MBE on the mesh in this example.

6.1.2. Forward error. The relative error in discrete solution $\|u_h - \tilde{u}_h\|_{L^2} / \|u_h\|_{L^2}$ is computed to verify our analysis in section 3. Here $\|u_h\|_{L^2}^2 = \mathbf{u}_h^T \mathbf{M} \mathbf{u}_h$, with $\mathbf{u}_h$ being

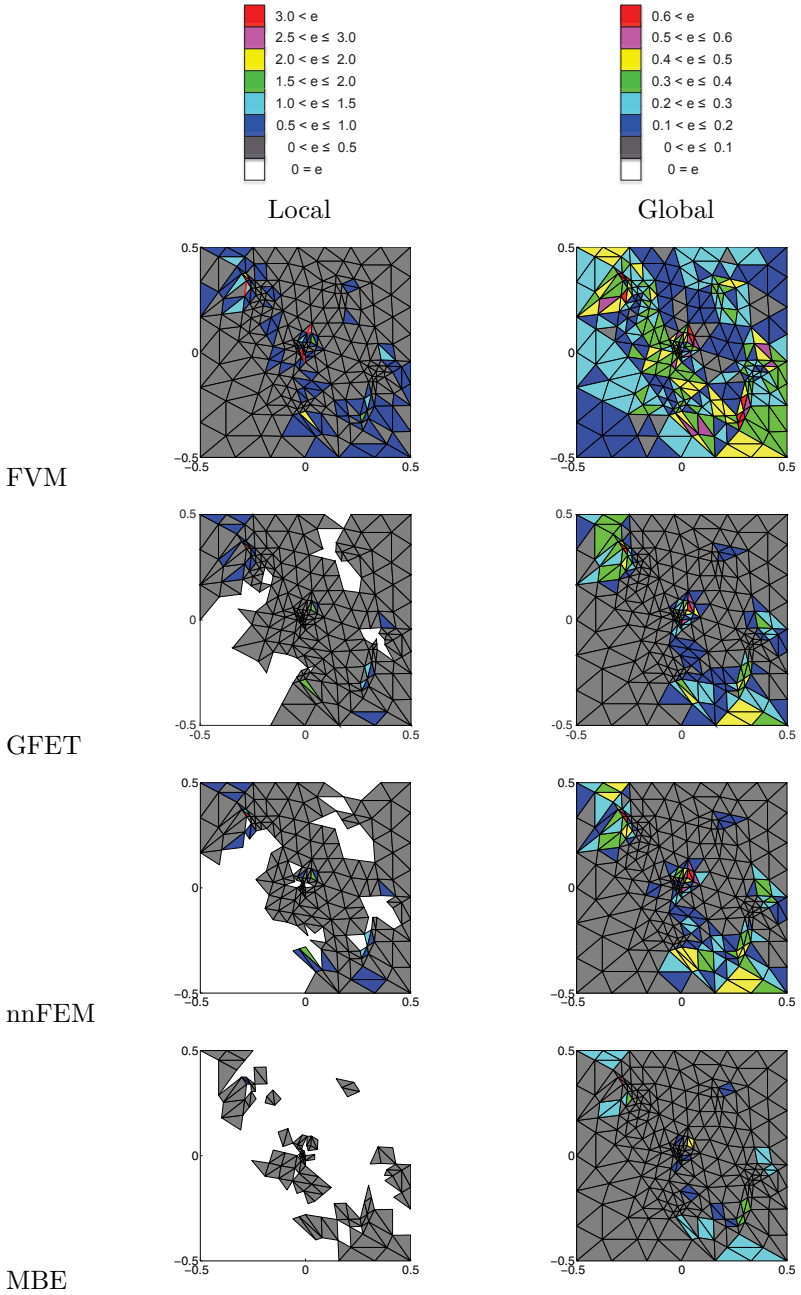


FIG. 3. *The backward error calculated by the local and global minimizations in sections* 4 *and* 5. *The local error is* $e = \|\tilde{\boldsymbol{\gamma}}_k - \boldsymbol{\gamma}\|_2$ *on each triangle* $T_k \in \mathcal{T}$.

the vector of the solution in each node and $\mathbf{M}$ the lumped mass matrix. The relative error is plotted between the discrete solution u_h with the original discretization matrix $\mathbf{D}$ (with negative off-diagonal entries) and the perturbed discrete solution $\tilde{u}_h$ resulting from one of the algorithms generating nonnegative off-diagonal entries in $\tilde{\mathbf{D}}$. The system of ODEs in (3.8) is solved by the MATLAB `ode15s`. Figure 4(a) shows

TABLE 2
The global backward errors in (6.2) *computed locally* η_2^L *and globally* η_2^G.

	η_2^L	η_2^G
FVM	0.4211	0.2729
nnFEM	0.2057	0.1524
GFET	0.1963	0.1356
MBE	0.0693	0.0690

the initial condition

$$u(\mathbf{x}, 0) = \tanh(20x_1)\tanh(20x_2) + 1 \tag{6.3}$$

on the mesh in Figure 2.

The forward error $\|u_h(\mathbf{x}, t) - \tilde{u}_h(\mathbf{x}, t)\|_{L^2}/\|u_h\|_{L^2}$ in space at $t = 0.01$ is depicted in Figures 4(b)–(f).

The forward error behaves in the way predicted by the backward error in Figure 3, Table 2, and the stability estimates in section 3. This is also confirmed in Figure 5, where the forward error in time is displayed in a log-lin scale such that the error for short times becomes visible. The unique steady state (3.17) is reached for large t, and the bound $\|u_h - \tilde{u}_h\|_2 \leq kt$ with some $k > 0$ for the error derived in section 3 is sharp for small t only.

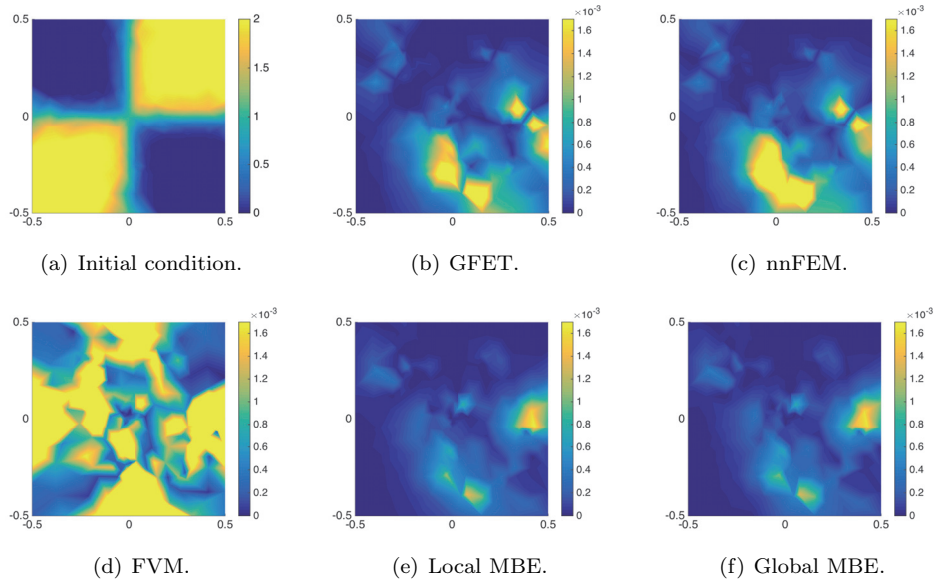
Comparing the results in Figure 5 and Table 2, we see that the order between the methods is the same using the minimization procedure in section 4 and the solutions of (2.5) and (2.9). The performance of the different methods in the forward error is correctly predicted by the performance in the backward error, as expected from section 3. The MBE is the best method and FVM is the worst method, but the forward error is quite small in all methods, with a peak for FVM of less than three percent.

6.1.3. Error in eigenvalues. In the original equation (2.4), the diffusion is isotropic, the quotient between the eigenvalues in (4.10) of γ in two dimensions is $\lambda_1/\lambda_2 = 1$, and the eigenvectors point in the coordinate directions. The quotient $q = \lambda_{\min}/\lambda_{\max}$ is g/G in Lemma 3.1, and for the FVM and the MBE, q is shown in Figure 6.

Avoiding the negative off-diagonal elements leads to a local anisotropy in the diffusion $\tilde{\boldsymbol{\gamma}}$ in Figure 6. The eigenvectors corresponding to the larger eigenvalues are not aligned but point in what appears to be random directions. The effect of the change of the diffusion from $\boldsymbol{\gamma}$ to $\tilde{\boldsymbol{\gamma}}$ is randomized over Ω. A global anisotropy is not found here, in contrast to what we have in the special regular rhombus mesh in [31]. All voxels are tilted there in the same direction, increasing the diffusion speed in this direction in almost all mesh triangles. A random change of the major axis of diffusion is expected in general in a mesh created by any mesh generator.

6.1.4. Alternative methods. The two alternatives suggested in section 5.4—minimizing the difference between $\mathbf{D}$ and $\tilde{\mathbf{D}}$ in the ℓ_2 norm and adding viscosity—are compared to the previous methods.

At a first glance at (3.7) and (3.8) it may seem like minimizing the difference between the matrices $\mathbf{D}$ and $\tilde{\mathbf{D}}$ in ℓ_2 would result in the smallest forward error, but this is not the case. Indeed, Figure 7 shows the forward error for this approach together with the best (MBE) and the worst (FVM) methods. Comparing the matrix deviations in Table 3 with the results in Figure 7 reveals that there is only a weak



(a) Initial condition. (b) GFET. (c) nnFEM.

(d) FVM. (e) Local MBE. (f) Global MBE.

FIG. 4. (a) *Initial condition.* (b)–(f) *Forward error* $\|u_h(\mathbf{x}_i) - \tilde{u}_h(\mathbf{x}_i)\|_{L^2}/\|u_h\|_{L^2}$ *for different approximations at each node at* $t = 0.01$.

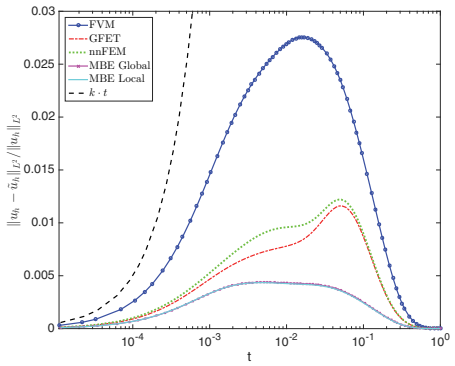
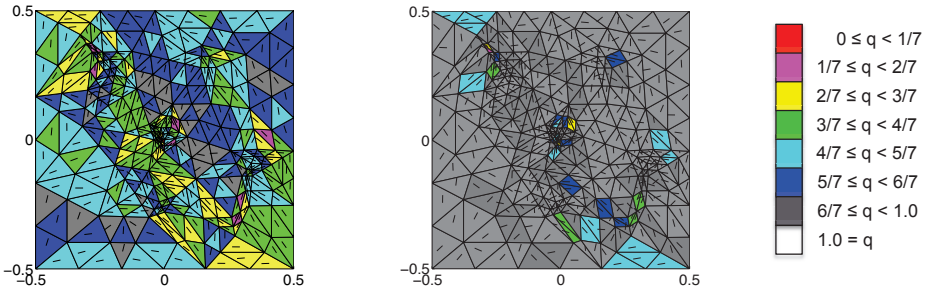


FIG. 5. *Forward relative error* $\|u_h - \tilde{u}_h\|_{L^2}/\|u_h\|_{L^2}$ *for* $0 \le t \le 1$.

correlation between a small forward error and the closeness of the matrices in the ℓ_2 norm.

The ℓ_2 norm of the difference in the discretization matrices in Table 3 does not reflect the behavior of the forward and backward errors, and the $\tilde{\mathbf{D}}$ optimized in the ℓ_2 norm does not reproduce the correct steady state in Figure 7. There is no unique equation and no unique $\tilde{\boldsymbol{\gamma}}(x)$ corresponding to a discretization matrix $\tilde{\mathbf{D}}$ in section 4. Hence, it is more meaningful to quantify and minimize the error in the solutions u_h and $\tilde{u}_h$ than to compare the discretization matrices representing many different analytical equations. In section 3 we showed that the $\tilde{\boldsymbol{\gamma}}$ closest to $\boldsymbol{\gamma}$ can be used to bound the error in the solution.

We see that adding viscosity to all nodes in the patch leads to adding an unnecessarily large amount of viscosity, resulting in a larger error than even the FVM in this case.



(a) (b)

FIG. 6. *Quotient between the two eigenvalues* $\lambda_{min}/\lambda_{max}$ *of* $\tilde{\boldsymbol{\gamma}}_i$ *and the direction of the eigenvector corresponding to the larger eigenvalue.* (a) *Global FVM.* (b) *Global MBE.*

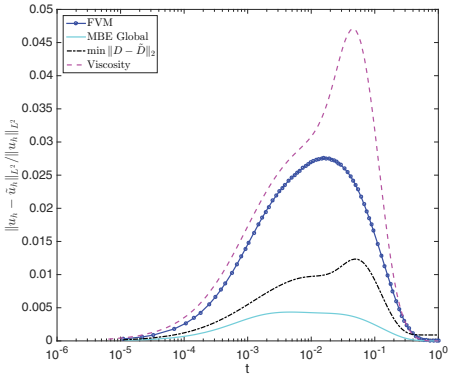


FIG. 7. *Forward relative error* $\|u_h - \tilde{u}_h\|_{L^2}/\|u_h\|_{L^2}$ *for* $0 \le t \le 1$, *for* ℓ_2*-optimization* $\eta_2^G = 0.1643$, *and for added viscosity* $\eta_2^G = 0.6265$.

TABLE 3
Relative difference $\|\mathbf{D} - \tilde{\mathbf{D}}\|_2/\|\mathbf{D}\|_2$ *between the discretization matrices.*

	$\|\mathbf{D} - \tilde{\mathbf{D}}\|_2/\|\mathbf{D}\|_2$
Viscosity	0.7856
FVM	0.4520
Local MBE	0.1967
Global MBE	0.1487
nnFEM	0.1059
GFET	0.0846
ℓ_2 optimal	0.0696

6.1.5. Practical implementation. The local backward error is too pessimistic in Table 2, but since only small optimization problems have to be solved involving the local triangles or tetrahedra to an edge, it is faster to compute. Furthermore, the backward error can be reduced by repeatedly applying Algorithm 2. The last solution in one iteration is the initial guess in the next iteration, and the edges are traversed in a different order in each iteration.

The effect of repeating Algorithm 2 is to spread the local error in each triangle over

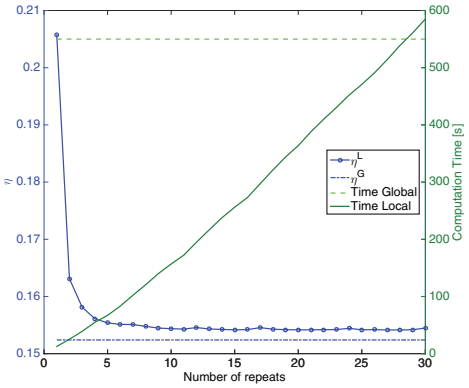


FIG. 8. *The error η_2^L for different numbers of iterations of Algorithm 2 and nnFEM, compared to the error η_2^G and the computing time for η_2^L and η_2^G.*

TABLE 4
The computation time in seconds for the global MBE and the resulting global error in the ℓ_2 norm when minimizing the ℓ_2 norm and the Frobenius norm in the primal (5.8) problem.

Minimization	Time	$\sqrt{\sum_{T_k\in\mathcal{T}} \lvert T_k\rvert \lVert\boldsymbol{\gamma}-\tilde{\boldsymbol{\gamma}}_k\rVert_2^2/\lvert\Omega\rvert}$
$\lVert\cdot\rVert_2$	3620	0.0690
$\lVert\cdot\rVert_F$	0.9737	0.0804

the domain. The η_2^L error is reduced towards the global minimum when altering the order of the edges in each iteration, but it does not seem to converge to η_2^G. Repeating Algorithm 2 about five times in Figure 8 is sufficient to achieve an improved backward error at a small increase in computation time.

Minimizing the ℓ_2 norm becomes prohibitively slow, especially in the most expensive algorithm of computing the MBE globally. Therefore, to arrive at a practical implementation we switch to the Frobenius norm (5.7). Then the minimization problem (5.1) has a quadratic objective function in (5.8) and is solved by the MATLAB `quadprog` with the interior-point-convex algorithm. The approximate computing times in seconds and the computed backward error for the global MBE are displayed in Table 4. Since $\lVert\mathbf{A}\rVert_2 \le \lVert\mathbf{A}\rVert_F$, the minimization in the Frobenius norm does not reach the global minimum in the ℓ_2 norm, but this is compensated for by a substantial speedup.

In Figure 9, we compare the forward errors for the MBE when minimizing in the Frobenius and ℓ_2 norms. The error resulting from minimization in the Frobenius norm is not much larger, while a reduction of the computing time of more than 3000 is achieved in Table 4.

6.2. Diffusion in three dimensions. Our methods are tested on a more realistic mesh such as those encountered in systems biology simulations. A sphere with radius 1 is discretized into two tetrahedral meshes with 602 and 1660 nodes. In both meshes, about 17 percent of the edges have a negative jump propensity with the standard FEM discretization. Mesh generators other than that in COMSOL Multiphysics were tested in [24] with similar results.

In the following experiments, the global backward error and $\tilde{\mathbf{D}}$ in MBE are

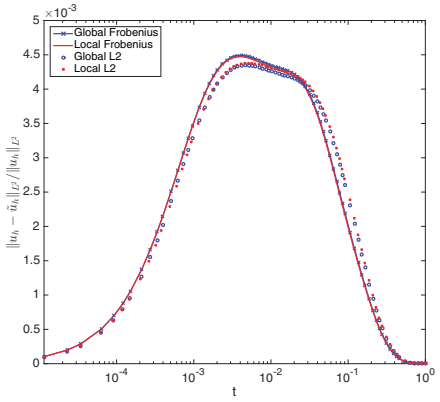


FIG. 9. *Forward error of the MBE with minimization in Frobenius and ℓ_2 norms. The errors for the global and local minimizations in the Frobenius norm are indistinguishable.*

TABLE 5

Locally and globally computed errors $\sqrt{\sum_{T_k \in \mathcal{T}} |T_k| \|\tilde{\boldsymbol{\gamma}}_k - \boldsymbol{\gamma}\|_2^2 / |\Omega|}$ *on the coarse (602 nodes) and fine (1660 nodes) meshes.*

	602		1660	
	Local	Global	Local	Global
FVM	0.6415	0.4284	0.6077	0.4227
nnFEM	0.3720	0.2898	0.3683	0.2818
GFET	0.3604	0.2618	0.3570	0.2632
MBE	0.1680	0.1593	0.1676	0.1698

computed by minimizing in the Frobenius norm. The number of unknown variables in the global optimization problem for the largest mesh is 51096 in the primal problem (5.8) and 10599 in the dual problem (5.10) in Table 5. The local optimization problems typically have 24 variables in the primal problem and 13 in the dual problem. In Figure 10 and Table 5, we see that the methods examined in two dimensions behave similarly to those in three dimensions. The FVM leads to a nonpositive definite diffusion $\tilde{\boldsymbol{\gamma}}_k$ in five tetrahedra on the coarse mesh and 12 on the fine mesh when calculated locally with Algorithm 2, where $\|\cdot\|_2$ is replaced by $\|\cdot\|_F$. There is a slight difference between the local and global MBEs but the error is in general small for all methods. The ranking of the methods is the same as in two dimensions in Table 2 for small t. Since the percentage of negative edges is the same when refining the mesh, the nnFEM and GFET methods do not improve on a finer mesh. How the backward error of the methods behaves when the mesh is refined depends not only on the mesh size but also on the shape of the elements. For small t, the methods perform in forward error on each mesh as predicted by the respective backward error in Table 5.

6.3. Mean first hitting time. Molecules in biological cells undergo not only diffusion but also reactions. In order to measure the error in a way relevant for reaction-diffusion kinetics, we construct a problem that mimics the mean first binding time for two molecules A and B diffusing in a spherical domain in a diffusion limited case. The assumption is that the molecules react instantaneously when they are in the same voxel. We calculate the mean time it takes for molecule A diffusing in Ω with reflecting boundary conditions at $\partial\Omega$ to reach a certain node i at $\mathbf{x}_i$, where its

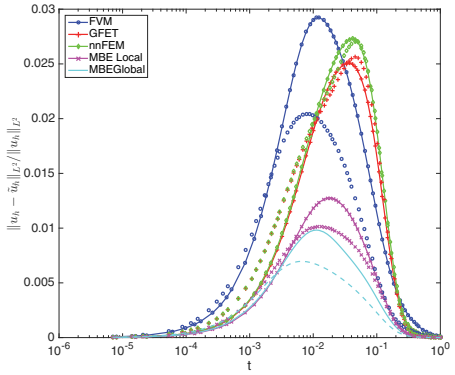


FIG. 10. *Forward error* $\|u_h - \tilde{u}_h\|_{L^2}/\|u_h\|_{L^2}$ *for the different methods on a mesh with* 602 *nodes (solid lines) and with* 1660 *nodes (dashed lines with the same markers) discretizing the unit sphere.*

reaction partner B is located. The molecule A is removed when it reaches the node at $\mathbf{x}_i$ by introducing a sink at this point. This setup models a reaction complex B situated at node $\mathbf{x}_i$, transforming our molecule of interest A.

Let $p_i(\mathbf{x}, t)$ be the probability distribution function of finding A in $\mathbf{x}$ at time t when B is at $\mathbf{x}_i$, and let δ be the Dirac measure. Then p_i satisfies

$$p_{it}(\mathbf{x}, t) = \gamma \Delta p_i(\mathbf{x}, t) - k\delta(\mathbf{x} - \mathbf{x}_i) p_i(\mathbf{x}, t), \tag{6.4}$$

with a Neumann boundary condition at $\partial\Omega$ and a constant $k > 0$. The mean value of the hitting time τ_i for A to find B is determined for all possible starting positions of A in the mesh. The initial condition is a uniform distribution of A, $p_i(\mathbf{x}, 0) = 1/|\Omega|$. The domain Ω is the sphere of radius 1 discretized by the same meshes as in the previous section. A discrete approximation of (6.4) is

$$\mathbf{p}_{it} = (\gamma \mathbf{D} - \mathbf{K}_i)\mathbf{p}_i, \tag{6.5}$$

where $\mathbf{p}_i^T = (p_{i1}, p_{i2}, \ldots, p_{iN})$ and $\mathbf{K}_i$ is the zero matrix except for $K_{ii} = 10^9$.

The survival probability $S_i(t)$ and the PDF $\pi_i(\tau)$ for τ_i are defined by

$$S_i(t) = \int_\Omega p_i(\mathbf{x}, t)\, d\mathbf{x} = P(\tau_i \geq t), \qquad \pi_i(t) = -S_{it}. \tag{6.6}$$

The expected value of the hitting time τ_i can then be calculated by

$$\begin{aligned} \mathbb{E}[\tau_i] &= \int_0^\infty \tau \pi_i(\tau)\, d\tau = -\int_0^\infty \tau S_{it}\, d\tau = \int_0^\infty \int_\Omega p_i(\mathbf{x}, \tau)\, d\mathbf{x} d\tau \\ &\approx \int_0^\infty \sum_{k=1}^N |\mathcal{V}_k| p_{ik}\, d\tau. \end{aligned} \tag{6.7}$$

In Table 6, we compare $\mathbb{E}[\tau_i]$ on the coarse mesh for the original discretization matrix $\mathbf{D}$ and the modified discretizations $\tilde{\mathbf{D}}$ described in sections 2 and 5. A sink is placed at one node i in the mesh. Since many interesting reactions in cells occur in reaction complexes bound to the membrane or in the nucleus, we especially investigate

TABLE 6

Averages of the expected first hitting time $\mathbb{E}[\tau_i]$ defined in (6.8) for different methods are shown in columns 2–5 on the mesh with 602 nodes.

	E_{All}	E_{Bnd}	E_{Cdet}	E_{Cstoch}
FEM	8.0413	11.6860	4.9211	N/A
FVM	8.8255	12.7055	5.9304	5.9733
GFET	7.7308	11.4102	3.6250	3.6424
nnFEM	7.4794	10.8482	4.4648	4.4707
MBE	8.2944	12.0863	5.3001	5.3323

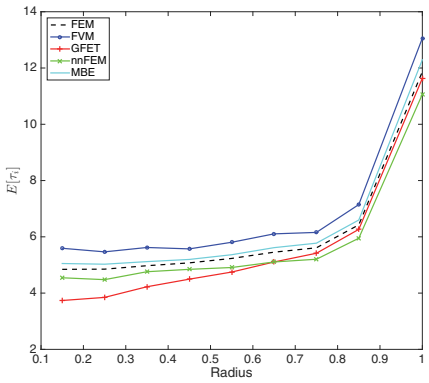


FIG. 11. *Expected exit times as a function of the radial position.*

the time it takes for A to find B either at a boundary node or at the node closest to the center. The average of $\mathbb{E}[\tau_i]$ over i is first computed when the sink and B are at any node and at any node on the boundary. Then the sink is at the node closest to the center. Finally, $\mathbb{E}[\tau_i]$ is computed with stochastic simulation employing Algorithm 1. The time for A to reach B at the center is recorded for each trajectory, and the average is taken over 10^5 realizations. The initial position of A is sampled from a uniform distribution. With N being the total number of nodes, N_B the number of boundary nodes, M the number of trajectories, and τ_c^m the hitting time of trajectory m, the quantities in the table are

$$
\begin{aligned}
&E_{\text{All}} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}[\tau_i], \quad E_{\text{Bnd}} = \frac{1}{N_B}\sum_{\mathbf{x}_i\in\partial\Omega}\mathbb{E}[\tau_i], \quad E_{\text{Cdet}} = \mathbb{E}[\tau_c],\\
&E_{\text{Cstoch}} = \frac{1}{M}\sum_{m=1}^{M}\tau_c^m, \quad E_{\text{Std}}^2 = \frac{1}{N_B-1}\sum_{\mathbf{x}_i\in\partial\Omega}(\mathbb{E}[\tau_i]-E_{\text{Bnd}})^2.
\end{aligned}
\tag{6.8}
$$

The results with standard FEM are found in the top row in the table as reference values. The FEM values are second order accurate and converge to the analytical values of the original diffusion equation when the mesh size is reduced. Since the FEM stiffness matrix has negative off-diagonal elements, stochastic simulation with its jump coefficients is impossible. In Figure 11, we illustrate the results in Table 6 by plotting the expected time to reach a node $\mathbf{x}_i$ as a function of its radial position. We average the expected exit times at all nodes in shells of the sphere of width 0.1.

The MBE is the superior method both in the average over all nodes and in reaching the center when compared to the FEM in Table 6 and Figure 11. The GFET, which

was designed in [31] to be accurate for the global first exit time—meaning the first hitting time of a boundary node—performs best for the boundary nodes. However, the GFET is not able to compute the time to reach the center of the cell very accurately. This shortcoming was discussed in [31]. The FVM is too slow, with a longer time to reach the sinks than FEM, and GFET and nnFEM are a little too fast, corresponding to a global diffusion coefficient larger than γ. This tendency was noted also in [31]. The results of the stochastic simulations in column 5 for the central node are close to the deterministic values of (6.7) in column 4 as expected for a large M.

The standard deviation of the mean time it takes to reach a boundary node measures the variation between different nodes at the boundary and should be small (ideally 0) for a good mesh and an accurate discretization. In Table 7, we compare the expected time to reach a boundary node and its standard deviation (6.8) for the different methods on the two meshes.

TABLE 7

Expected time to reach a boundary node and its standard deviation on a mesh with 602 *nodes (left) and a fine mesh with* 1660 *nodes (right).*

	602		1660	
	E_{Bnd}	$E_{\mathrm{Std}}/E_{\mathrm{Bnd}}$	E_{Bnd}	$E_{\mathrm{Std}}/E_{\mathrm{Bnd}}$
FEM	11.6860	0.0876	15.8015	0.0808
FVM	12.7055	0.1557	16.7264	0.1386
GFET	11.4102	0.1170	15.4179	0.1060
nnFEM	10.8482	0.1171	14.6743	0.1060
MBE	12.0863	0.0908	16.2679	0.0853

The standard deviation of MBE is close to that of FEM, demonstrating that the anisotropy in the diffusion introduced by MBE has a small impact compared to the accuracy effects of the discretization and the mesh. The small difference in $E_{\mathrm{Std}}/E_{\mathrm{Bnd}}$ between MBE and FEM is most likely explained by the random directions of maximum and minimum diffusion as in Figure 6. The relative standard deviation is reduced slightly when the mesh is refined but E_{Bnd} has not yet converged. The E_{Bnd} closest to the FEM value is obtained by GFET. When simulating a signal being transmitted inside the cell, it is advantageous to use GFET if the important reactions occur on the membrane. On the other hand, if the signal is traveling inside the cytoplasm and reacting there, then the MBE results in the most accurate transmission time in Table 6.

7. Conclusion. For the discrete stochastic simulation of diffusion in systems biology, we need jump propensities for the molecules in the discrete space model. These propensities are chosen as the off-diagonal elements of the discretization matrix obtained by a numerical approximation of the Laplacian. The jump coefficients have to be nonnegative. For unstructured meshes, nonnegative off-diagonal elements cannot be guaranteed with a discretization matrix assembled by a standard finite element method (FEM), but there exist different approaches to change this discretization matrix to fulfill the nonnegativity condition. As a result of this change, a diffusion equation with an altered diffusion is approximated.

We first present a method to analyze these existing methods, producing nonnegative jump propensities on an unstructured mesh of poor quality. The difference between the solution to the original and the perturbed diffusion equations is bounded by the difference in the diffusion coefficients. Then the perturbed diffusion is retrieved by backward analysis. This leads us to the derivation of a new algorithm creating a

discretization matrix based on FEM, minimizing the backward error.

We show in numerical experiments that the finite volume method (FVM) to compute a nonnegative discretization incurs high forward and backward errors on our meshes. Our previously proposed methods of eliminating the negative coefficients in the finite element method (nnFEM) and satisfying the global first exit time constraint (GFET) perform comparably. The new method to generate jump coefficients on a given mesh proposed in this paper results in a considerably smaller error on both an artificial mesh in two dimensions and a realistic mesh in three dimensions.

The average of the first hitting time obtained by stochastic simulations with nonnegative jump coefficients is close to the solution of a deterministic equation with a modified diffusion as expected. The accuracy of this average compared to the exact analytical values depends not only on the number of trajectories in the Monte Carlo simulation but also on the mesh size and the mesh quality. In general, with the MBE the backward and forward errors are small and the mean hitting time to any node is well approximated. The errors in the stochastic diffusion simulation are of the same order as the errors in biological measurements [37, 39]. Furthermore, there is a variation in the diffusion constant across the cell in measurements in [40] comparable to the variation of the space-dependent $\tilde{\gamma}$ of the perturbed diffusion equation (3.2) determined by our algorithms.

Since the off-diagonal elements are nonnegative, the FEM discretization of the equation with the modified diffusion satisfies the sufficient conditions for the discrete maximum principle for the FEM solution. The discrete maximum principle being satisfied for the original diffusion on any mesh seems to be possible only with a nonlinear scheme as in [4]. Then the stiffness matrix $\mathbf{S}$ is reassembled in every time step. Our scheme is linear with a constant $\mathbf{S}$ but for a modified diffusion.

Acknowledgments. We have had fruitful discussions with Murtazo Nazarov and Stefano Serra-Capizzano concerning parts of this work.

REFERENCES

[1] 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 (2010), e1000705.

[2] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004.

[3] J. Brandts, S. Korotov, M. Křížek, and J. Šolc, *On nonobtuse simplicial partitions*, SIAM Rev., 51 (2009), pp. 317–335.

[4] E. Burman and A. Ern, *Stabilized Galerkin approximation of convection-diffusion-reaction equations: Discrete maximum principle and convergence*, Math. Comp., 74 (2005), pp. 1637–1652.

[5] Y. Cao, D. T. Gillespie, and L. R. Petzold, *The slow-scale stochastic simulation algorithm*, J. Chem. Phys., 122 (2005), 014116.

[6] A. Donev, V. V. Bulatov, T. Oppelstrup, G. H. Gilmer, B. Sadigh, and M. H. Kalos, *A first-passage kinetic Monte Carlo algorithm for complex diffusion-reaction systems*, J. Comput. Phys., 229 (2010), pp. 3214–3236.

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

[8] M. B. Elowitz, A. J. Levine, E. D. Siggia, and P. S. Swain, *Stochastic gene expression in a single cell*, Science, 297 (2002), pp. 1183–1186.

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

[10] H. Erten and A. Üngör, *Quality triangulations with locally optimal Steiner points*, SIAM J. Sci. Comput., 31 (2009), pp. 2103–2130.

[11] L. C. Evans, *Partial Differential Equations*, Grad. Stud. Math. 19, AMS, Providence, RI, 1998.

[12] R. Eymard, T. Gallouët, and R. Herbin, *A cell-centred finite-volume approximation for anisotropic diffusion operators on unstructured meshes in any dimension*, IMA J. Numer. Anal., 26 (2006), pp. 326–353.
[13] Z. Gao and J. Wu, *A second-order positivity-preserving finite volume scheme for diffusion equations on general meshes*, SIAM J. Sci. Comput., 37 (2015), pp. A420–A438.
[14] C. W. Gardiner, K. J. McNeil, D. F. Walls, and I. S. Matheson, *Correlations in stochastic theories of chemical reactions*, J. Stat. Phys., 14 (1976), pp. 307–331.
[15] M. A. Gibson and J. Bruck, *Efficient exact stochastic simulation of chemical systems with many species and many channels*, J. Phys. Chem., 104 (2000), pp. 1876–1889.
[16] D. T. Gillespie, *A general method for numerically simulating the stochastic time evolution of coupled chemical reactions*, J. Comput. Phys., 22 (1976), pp. 403–434.
[17] D. T. Gillespie, A. Hellander, and L. R. Petzold, *Perspective: Stochastic algorithms for chemical kinetics*, J. Chem. Phys., 138 (2013), 170901.
[18] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, 1989.
[19] J. L. Guermond and M. Nazarov, *A maximum-principle preserving C^0 finite element method for scalar conservation equations*, Comput. Methods Appl. Mech. Engrg., 272 (2014), pp. 198–213.
[20] J. Hattne, D. Fange, and J. Elf, *Stochastic reaction-diffusion simulation with MesoRD*, Bioinformatics, 21 (2005), pp. 2923–2924.
[21] I. Hepburn, W. Chen, S. Wils, and E. De Schutter, *STEPS: Efficient simulation of stochastic reaction-diffusion models in realistic morphologies*, BMC Syst. Biol., 6 (2012), 36.
[22] S. A. Isaacson and C. S. Peskin, *Incorporating diffusion in complex geometries into stochastic chemical kinetics simulations*, SIAM J. Sci. Comput., 28 (2006), pp. 47–74.
[23] 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 (2008), pp. 3126–3149.
[24] E. Kieri, *Accuracy Aspects of the Reaction-Diffusion Master Equation on Unstructured Meshes*, MSc thesis, Tech. report UPTEC F 11014, Uppsala University, Uppsala, Sweden, 2011.
[25] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, *An interior-point method for large-scale ℓ_1-regularized least squares*, IEEE J. Selected Topics Signal Process., 1 (2007), pp. 606–617.
[26] D. J. Kiviet, P. Nghe, N. Walker, S. Boulineau, V. Sunderlikova, and S. J. Tans, *Stochasticity of metabolism and growth at the single-cell level*, Nature, 514 (2014), pp. 376–379.
[27] S. Korotov, M. Křížek, and P. Neittanmäki, *Weakened acute type condition for tetrahedral triangulations and the discrete maximum principle*, Math. Comp., 70 (2000), pp. 107–119.
[28] T. G. Kurtz, *Solutions of ordinary differential equations as limits of pure jump Markov processes*, J. Appl. Probab., 7 (1970), pp. 49–58.
[29] T. G. Kurtz, *Limit theorems for sequences of jump Markov processes approximating ordinary differential processes*, J. Appl. Probab., 8 (1971), pp. 344–356.
[30] A. Logg, K.-A. Mardal, and G. Wells, *Automated Solution of Differential Equations by the Finite Element Method*, Springer, Berlin, 2012.
[31] P. Lötstedt and L. Meinecke, *Simulation of stochastic diffusion via first exit times*, J. Comput. Phys., 300 (2015), pp. 862–886.
[32] H. H. McAdams and A. Arkin, *Stochastic mechanisms in gene expression*, Proc. Natl. Acad. Sci. USA, 94 (1997), pp. 814–819.
[33] D. A. McQuarrie, *Stochastic approach to chemical kinetics*, J. Appl. Probab., 4 (1967), pp. 413–478.
[34] R. Metzler, *The future is noisy: The role of spatial fluctuations in genetic switching*, Phys. Rev. Lett., 87 (2001), 068103.
[35] B. Munsky, G. Neuert, and A. van Oudenaarden, *Using gene expression noise to understand gene regulation*, Science, 336 (2012), pp. 183–187.
[36] B. Øksendal, *Stochastic Differential Equations*, 6th ed., Springer, Berlin, 2003.
[37] F. Persson, M. Lindén, C. Unoson, and J. Elf, *Extracting intracellular diffusive states and transition rates from single molecule tracking data*, Nature Methods, 10 (2013), pp. 265–269.
[38] A. Raj and A. van Oudenaarden, *Nature, nurture, or chance: Stochastic gene expression and its consequences*, Cell, 135 (2008), pp. 216–226.
[39] C. Di Rienzo, V. Piazza, E. Gratton, F. Beltram, and F. Cardarelli, *Probing short-range protein Brownian motion in the cytoplasm of living cells*, Nature Commun., 5 (2014), 5891.

[40] L. Lanzano S. Ranjit, and E. Gratton, *Mapping diffusion in a living cell via the phasor approach*, Biophys. J., 107 (2014), pp. 2775–2785.
[41] J. Schöneberg, A. Ullrich, and F. Noé, *Simulation tools for particle-based reaction-diffusion dynamics in continuous space*, BMC Biophys., 7 (2014), 11.
[42] Z. Sheng and G. Yuan, *An improved monotone finite volume scheme for diffusion equation on polygonal meshes*, J. Comput. Phys., 231 (2012), pp. 3739–3754.
[43] J. R. Shewchuk, *What is a good linear element? Interpolation, conditioning, and quality measures*, in Proceedings of the 11th International Meshing Roundtable (Ithaca, NY), Springer-Verlag, New York, 2002, pp. 115–126.
[44] M. Svärd, J. Gong, and J. Nordström, *An accuracy evaluation of unstructured node-centred finite volume methods*, Appl. Numer. Math., 58 (2008), pp. 1142–1158.
[45] P. S. Swain, M. B. Elowitz, and E. D. Siggia, *Intrinsic and extrinsic contributions to stochasticity in gene expression*, Proc. Natl. Acad. Sci. USA, 99 (2002), pp. 12795–12800.
[46] V. Thomée, *Galerkin Finite Element Methods for Parabolic Problems*, Springer, Berlin, 1997.
[47] L. Vandenberghe and S. Boyd, *Semidefinite programming*, SIAM Rev., 38 (1996), pp. 49–95.
[48] R. S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1962.
[49] J. Xu and L. Zikatanov, *A monotone finite element scheme for convection-diffusion equations*, Math. Comp., 68 (1999), pp. 1429–1446.
[50] 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 (2005), 234910.