Spatial stochastic simulation of cellular reaction networks

A comparison of discretizations of the Laplace operator for mesoscopic diffusion

Daniel Elfverson
Emil Kieri

Project in Computational Science: Report

June 2010
Abstract

The software URDME simulates stochastic chemical systems by sampling trajectories that are statistically consistent with a reaction-diffusion master equation. The probabilities for diffusive jumps are derived from a finite element discretization of the Laplace operator. If there exist obtuse dihedral angles in the mesh, jump coefficients may turn out negative, making a probabilistic interpretation troublesome. These negative coefficients are zeroed out, leading to wrong diffusion rates and an $O(1)$ error in the solution. We have implemented a node-centered edge-based finite volume discretization in the place of the finite volume method, and compared the convergence properties of the two methods. This finite volume method was found to be of accuracy similar to the finite element method. These discretization errors seem to have a limited effect on the solution quality. It is our belief that other sources of error are dominant.
## Contents

1 Introduction 3

2 The reaction-diffusion master equation 4

3 Diffusion coefficients 6
   3.1 Finite element derivation 7
   3.2 Finite volume derivation 8

4 Time integration 9

5 Experiments 9
   5.1 Macroscopic diffusion in a cube 9
   5.2 Macroscopic diffusion in a sphere 11
   5.3 Convergence of stochastic sampling 13
   5.4 Truncation of the stiffness matrix 14
   5.5 Comparison of mesh generators 16

6 The MinD/E system of E. coli 18

7 Conclusions 25

A The COMSOL fem struct 28
1 Introduction

The traditional approach to simulating chemical reaction networks is the reaction rate equations (RRE), a system of ordinary differential equations (ODEs) for the concentrations of the molecular species involved. The RRE rely on two assumptions: high copy numbers of the species involved and spatial homogeneity. The first is needed in order to treat the concentrations as continuous quantities, the second since there is no spatial dependence in the model. Inside a living cell, neither of these assumptions hold in general. Some molecular species involved in cellular reaction networks appear in very low copy numbers. This gives rise to stochastic noise which cannot be captured by deterministic models such as the RRE. In [25], Vilar et al. give an example of a cellular reaction network where stochastic noise results in a radically different behaviour of the system, compared to integration of the RRE.

Better accuracy can be achieved by using a stochastic model. Let the state of the system be the copy numbers of the molecular species involved. Let the chemical reactions be instantaneous events that change the state, and assume that the time until the next reaction has an exponential probability distribution function (PDF). This PDF should only depend on the current state, not on the history, i.e. the model should have the Markov property. The system is then modeled as a Markov process. As a consequence, the time evolution of the probability for a system to be in each of the states in the state-space is given by a master equation [14], which is a system of ODEs. For this kind of well-stirred chemical systems, the equation is called the chemical master equation (CME). The dimensionality of the CME is the number of states, and it is easy to see that this easily becomes a very high number. If there is no formal bound on the copy number of one of the species the CME is infinite-dimensional. Direct solution of the CME is usually out of reach.

The Gillespie method, or the stochastic simulation algorithm (SSA) [10], is a Monte Carlo method for calculating trajectories of the system that are statistically consistent with the CME. An efficient implementation of SSA is the next reaction method (NRM) due to Gibson and Bruck [9]. The stochastic noise in cellular reaction networks is captured by SSA, but one violated assumption remains. Also the CME assumes the system to be spatially homogeneous. In many reaction networks of living cells this is not the case. Diffusion may be slow compared to reaction times, and some reactions may be localized, e.g. to membranes or organelles [20].

To capture spatial differences, the domain is divided into subvolumes which are small enough to be considered spatially homogeneous. The state of the system is the copy numbers of the involved molecular species in each of the subvolumes. Chemical reactions may occur between molecules in the same subvolume. Diffusion is modeled by molecules jumping to neighbouring subvolumes. The model still has the Markov property, and the probability distribution of the time until the next reaction or diffusion event is exponential. The model of the system is then a continuous-time discrete-space Markov process (CTMC), and its time evolution is governed by the reaction-diffusion master equation (RDME) [19].

An efficient method for sampling trajectories consistent with the RDME is the next subvolume method (NSM) due to Elf and Ehrenberg [4]. The software URDME [3] is an implementation of NSM on unstructured meshes. The method
used by URDME to calculate the diffusion jump probabilities is presented in [5]. There, the diffusion probabilities are calculated by a finite element discretization of the Laplace operator, using piecewise linear basis functions. A problem with that is that some probabilities may come out negative if there exist obtuse dihedral angles in the mesh. In this report, we investigate if a finite volume discretization can be more appropriate. If the Laplace operator is discretized with a node-centered finite volume method (FVM) all probabilities will be non-negative, but the FVM has other convergence issues, see e.g. [24].

2 The reaction-diffusion master equation

The computational domain \( \Omega \subset \mathbb{R}^d \), \( d = 2, 3 \) is divided into non-overlapping computational cells \( C_j \), \( j = 1, \ldots, K \). The cells are polygons in 2D and polyhedra in 3D. In \( \Omega \), a chemical system of \( N \) molecular species is modeled. The molecules diffuse inside \( \Omega \), and may engage in \( R \) different chemical reactions. Let \( X_{ij} \) denote a molecule of species \( i \) in \( C_j \). The state of the system is the copy number of each of the species in each of the cells. The state is represented by the \( N \times K \)-matrix \( x \), where \( x_{ij} \) is the copy number of species \( i \) in \( C_j \). Molecules in the same cell may interact in chemical reactions, and molecules may diffuse to neighbouring cells. Two cells are considered neighbours if they share an edge in 2D or a face in 3D. The system is stochastic and assumed to have the Markov property [14, Ch. IV]. The time until the next reaction or diffusion event is assumed to have an exponential PDF. This makes the time evolution of the system a CTMC. The PDF \( p(x, t) \) of the system, interpreted as the probability of the system being in state \( x \) at time \( t \), is then governed by a master equation [14, Ch. V], in this case the RDME.

A reaction or diffusion event is an instantaneous transition from one state to another. A reaction is defined by a stochiometry vector \( n \) and a propensity function \( w \). The stochiometry vector is the nominal difference in state, and the value of the propensity function is the probability per unit time that the reaction will occur. In this manner, the reaction \( r \) in cell \( j \) can be modeled by

\[
x_{\cdot j} \xrightarrow{w_{rj}(x_{\cdot j})} x_{\cdot j} + n_{r, r}.
\]

As an example, the stochiometry vector \( n_{r, r} \) of the reaction in \( C_j \)

\[
X_{ij} + X_{kj} \rightarrow X_{lj}
\]

has \( n_{lr} = 1, n_{ir} = n_{kr} = -1 \) and all other elements equal to 0. Diffusion can at this mesoscopic level be modeled in the same way as chemical reactions. A molecule of species \( i \) diffusing from \( C_j \) to \( C_k \) is described by the reaction

\[
X_{ij} \xrightarrow{v_{jk}^{(i)}(x_{ij})} X_{ik}.
\]

The propensity is proportional to the copy number of species \( i \) in \( C_j \), \( v_{jk}^{(i)}(x_{ij}) = q_{jk}^{(i)} x_{ij} \), where \( q_{jk}^{(i)} \) is constant, but dependent on the shape of the cells \( C_j \) and \( C_k \), and on the properties of the species \( i \). Naturally, \( q_{jk}^{(i)} \) is non-zero only if \( C_j \) and \( C_k \) are neighbours. The stochiometry vector for this event is 1 in element
\[ \frac{\partial p(x,t)}{\partial t} = \mathcal{M}p(x,t) + \mathcal{D}p(x,t), \] (4)

where $\mathcal{M}$ governs the reactions and $\mathcal{D}$ the diffusion. Let $m^{(i)}_{jk}$ be the stochiometry vector for the diffusion event (3). The master operators are then defined by

\[
\mathcal{M}p(x,t) = \sum_{r=1}^{R} \sum_{j=1}^{K} w_{rj}(x \cdot j - n \cdot r)p(x,1,\ldots,x_{j} - n_{r},\ldots,x_{K},t) - \sum_{r=1}^{R} \sum_{j=1}^{K} w_{rj}(x \cdot j)p(x,t), \tag{5}
\]

\[
\mathcal{D}p(x,t) = \sum_{i=1}^{N} \sum_{j=1}^{K} \sum_{k=1}^{K} q^{(i)}_{jk}(x \cdot i - m^{(i)}_{jk})p(x_{i},\ldots,x_{i} - m^{(i)}_{jk},\ldots,x_{N},t) - \sum_{i=1}^{N} \sum_{j=1}^{K} \sum_{k=1}^{K} q^{(i)}_{jk} x_{ik} p(x,t). \tag{6}
\]

The first sums sum over the events that will cause the system to enter state $x$, and the second sums sum over the events that will cause the system to leave state $x$. The master equation (4) can be written on $W$-matrix form as

\[
\frac{\partial p(x,t)}{\partial t} = Wp(x,t), \tag{7}
\]

where the matrix $W$ has two important properties [14, Ch.V.2]:

\[
W_{jk} \geq 0, \quad j \neq k, \tag{8}
\]

\[
\sum_{j} W_{jk} = 0, \quad \forall k. \tag{9}
\]

Equation (8) ensures that the transition propensities are non-negative, and (9) ensures that no probability mass leaves the system. Direct solution of (4) is obviously intractable. Even storing a numerical solution on a digital medium may be out of reach for realistic problems. As an example, consider a small problem with two species where the copy number of each species may never exceed 100, and a mesh with 100 cells. That results in a master equation in
one million dimensions and time. We are forced to sample trajectories that are statistically consistent with the master equation, and draw as many conclusions as we can from them. Sometimes a single trajectory contains a lot of information about the behaviour of the system, sometimes it is better to compute many trajectories and extract their statistical properties.

3 Diffusion coefficients

The jump coefficients $q_{jk}$ in the RDME are calculated by discretizing the Laplace operator. On the macroscopic level, a system with one species and no reactions is described by the diffusion equation,

$$\dot{\phi} = \gamma \Delta \phi, \quad r \in \Omega, t > 0, \quad (10)$$

$$n \cdot \nabla \phi = 0, \quad r \in \partial \Omega, t > 0,$$

where $\phi(r,t)$ is the local concentration of the species and $n$ the outward unit normal of $\Omega$. A primal mesh of triangles in 2D and tetrahedrons in 3D is introduced. The nodes of the primal mesh are denoted by $r_j$. A dual mesh is constructed by connecting the midpoints of the edges with the midpoints of the triangles as in Figure 1. The dual cell with centroid $r_j$ is denoted by $C_j$. In 3D, the midpoints of the tetrahedra, their faces and edges are connected in an analogous manner. Equation (10) is then discretized in space on the mesh to

$$\dot{\phi} = \gamma D \phi. \quad (11)$$

The mesoscopic diffusion jump probabilities are picked directly from the matrix $D$, $q_{jk} = \gamma D_{jk}, j \neq k, q_{jj} = -\sum_{k \neq j} q_{jk}$, and are formally put in the $W$-matrix. $\gamma D^T$ is supposed to have the properties (8) and (9) of the $W$-matrix, but note that $\gamma D^T$ is not the $W$-matrix itself. $D$ is an $NK \times NK$-matrix, while $W$ reflects the dimensionality of the RDME which is much higher.

![Figure 1: A part of an unstructured mesh. The solid lines denote the primal mesh, and the dashed lines the dual mesh. $r_j$ is a node, and $n_k$ is the unit normal of the dual cell $C_j$ towards $C_k$](image)

On the microscopic level the jump coefficient $q_{jk}$ is interpreted as the expectation value of the first exit time of a Brownian particle in $C_j$ to $C_k$. This interpretation restricts our freedom in choosing a discretization of the Laplace operator. It rules out high-order discretization schemes since $q_{jk}$ cannot depend on the shape of any other subvolumes than $C_j$ and $C_k$. 

6
3.1 Finite element derivation

One approach to discretizing (10) is the finite element method (FEM). This was originally done in [5], and that derivation is followed here. A space \( V_h \) of continuous functions which are linear on each of the triangles or tetrahedra is introduced. Basis functions \( \varphi_i \) are constructed such that \( V_h = \text{span}\{\varphi_i\}_{i=1}^K \), \( \varphi_i(x_j) = \delta_{ij} \). The FEM approximation of (10) is then

For each \( t > 0 \), find \( \varphi_i \in V_h \) such that

\[
\int_{\Omega} \dot{\varphi}_i \varphi_j \, dx = -\gamma \int_{\partial\Omega} \nabla \varphi_i \cdot \nabla \varphi_j \, dx, \quad i = 1, \ldots, K.
\]  

(12)

This can be written on matrix form as

\[
M \dot{\phi} = \gamma S \phi,
\]  

(13)

where \( \phi \) is the nodal values of \( \phi \), \( M \) the mass matrix and \( S \) the stiffness matrix. For the details of a FEM discretization, consult e.g. [17]. Introduce the lumped mass matrix \( A \), \( A_{jk} = 0 \), \( j \neq k \), \( A_{jj} = \sum_{k=1}^N M_{jk} \). Then

\[
\dot{\phi} = \gamma D \phi, \quad D = A^{-1} S \]  

(14)

is a second-order approximation of (13), and \( A_{jj} = |C_j| \). It can easily be concluded that \( \gamma D^T \) fulfills the property (9):

\[
\sum_{k=1}^K \gamma D_{jk} = \gamma A^{-1}_{jj} \sum_{k=1}^K S_{jk}
\]

\[
= -\gamma A^{-1}_{jj} \sum_{k=1}^K \int_{\Omega} \nabla \varphi_j \cdot \nabla \varphi_k \, dx
\]

\[
= -\gamma A^{-1}_{jj} \int_{\Omega} \nabla \varphi_j \cdot \nabla \left( \sum_{k=1}^K \varphi_k \right) \, dx
\]

\[
= -\gamma A^{-1}_{jj} \int_{\Omega} \nabla \varphi_j \cdot \nabla (1) \, dx = 0.
\]

Unfortunately \( \gamma D^T \) does not in general fulfill property (8). If the angles between two edges in a triangle or two faces in a tetrahedron is no bigger than \( \frac{\pi}{2} \) everywhere in the primal mesh, the property is fulfilled. But if there are obtuse angles, off-diagonal elements in the stiffness matrix may become negative. This is illustrated in Figure 2. On the triangle \( \nabla \varphi_i \cdot \nabla \varphi_j > 0 \), so the triangle will give a negative contribution to \( S_{ij} \). If all angles are acute this cannot happen. Non-negativity of the off-diagonal elements is necessary, otherwise the probabilistic interpretation breaks down. To achieve this, any negative off-diagonal element in the stiffness matrix is zeroed out, and the diagonal elements are corrected so that the rows still sum to zero. This truncated stiffness matrix is denoted by \( \tilde{S} \). Generally in a simulation with URDME, the mesh is generated with COMSOL Multiphysics [1]. In a typical 3D mesh from COMSOL around 20 % of the non-zero off-diagonal elements of the stiffness matrix turn out negative.
3.2 Finite volume derivation

Another approach is to use a finite volume method (FVM). An advantage with FVM is that the probabilities for a molecule to move into another cell will always be non-negative. But it has the disadvantage that it is not consistent with the Laplace operator on a general unstructured mesh [24].

To obtain a FVM discretization we start by integrating over the subvolumes \( C_j \),

\[
\int_{C_j} \dot{\phi} dx = \gamma \int_{C_j} \Delta \phi dx. \tag{15}
\]

Then by using the Gauss’ divergence theorem on the right hand side we obtain for each \( j \)

\[
\frac{\partial}{\partial t} \int_{C_j} \phi dx = \gamma \int_{\partial C_j} \mathbf{n} \cdot \nabla \phi dS. \tag{16}
\]

Replacing the integrand in \( \int_{C_j} \phi dx \) with its mean value on the subvolume and summing over the faces of \( C_j \) gives us

\[
|C_j| \dot{\phi}_j = \gamma \sum_{k \in N_j} \int_{\partial C^k_j} |\mathbf{n}_k \cdot \nabla \phi dS, \tag{17}
\]

where \( N_j \) is the set of indices for cells neighbouring \( C_j \), \( \partial C^k_j \) is the common boundary between \( C_j \) and \( C_k \), and \( \mathbf{n}_k \) the unit normal of that boundary, pointing into \( C_k \).

Here, \( \mathbf{n}_k \cdot \nabla \phi \) is approximated by the nodal values divided by the distance between the nodes,

\[
|C_j| \dot{\phi}_j \approx \gamma \sum_{k \in N_j} |\partial C^k_j| \frac{\phi_k - \phi_j}{\| \mathbf{r}_k - \mathbf{r}_j \|}, \tag{18}
\]

\[
\dot{\phi}_j = \gamma \sum_{k \in N_j} \frac{|\partial C^k_j|}{|C_j|} \frac{\phi_k - \phi_j}{\| \mathbf{r}_k - \mathbf{r}_j \|}. \tag{19}
\]
This can be written in matrix notation as

\[ \dot{\phi} = \gamma D \phi, \quad D = A^{-1} \tilde{S}, \quad (20) \]

where \( A \) is the lumped mass matrix, \( A_{jk} = 0 \) if \( j \neq k \) and \( A_{jj} = |C_j| \), and \( \tilde{S} \) is the stiffness matrix,

\[ \tilde{S}_{jk} = \frac{|C_j|}{\|r_k - r_j\|}, \quad j \neq k \] and

\[ \tilde{S}_{jj} = -\sum_{k \in \mathcal{N}_j} \frac{|C_j|}{\|r_k - r_j\|}, \quad D \] will then have the desired properties

\[ D_{jk} \geq 0, \quad j \neq k \] and \( \sum_{K} D_{jk} = 0, \quad j = 1, \ldots, K \) so we can directly do the probabilistic interpretation \( q_{jk} = \gamma D_{jk} \).

4 Time integration

Stochastic simulation of the chemical system is made in URDME, which uses an implementation of NSM on unstructured tetrahedral meshes. For each computational cell \( C_i \), the total rates of diffusion events \( s_i \) and reaction events \( r_i \) are calculated. These depend on the copy numbers of the chemical species in \( C_i \). The time until the next event in \( C_i \) is exponentially distributed with expectation value \( \frac{1}{s_i + r_i} \). A sample \( \tau_i \) is drawn from these distributions for each of the cells, and the first event is taken to happen in the cell \( C_j \) having \( \tau_j = \min_i \tau_i \). A new random number, uniformly distributed between 0 and 1, is generated in order to determine which event is to occur. The reaction or diffusion event \( \lambda \) with rate \( f_\lambda \) occurs with probability \( \frac{f_\lambda s_j + r_j}{s_j + r_j} \).

The event changes the copy numbers of species in \( C_j \), and if a diffusion event also in a neighbouring cell \( C_k \). This changes the rates for some reaction and diffusion events in \( C_j \) and \( C_k \), which have to be updated. A new \( \tau_j \) is drawn. This is not necessary for \( \tau_k \), it is enough to rescale it according to the change in its expectation value. There is now a new smallest \( \tau_i \), and the process is repeated. For the details of NSM, see [4], and for details of its parent algorithm, NRM, see [9].

This process is an exact stochastic sampling of the corresponding master equation. It is though important to note that the master equation itself does not give an exact description of the system. It does, as all other models, suffer from a modeling error, and the deduction of diffusion coefficients from any discretization of the Laplace operator will be subject to a truncation error. Currently inconsistent discretizations are used, refining the mesh beyond a certain resolution will not reduce the truncation error. The model itself also sets a limit to the mesh resolution, it breaks down if the subvolumes become too small. Reactions can only occur between molecules in the same subvolume, in the limit \( h \to 0 \) there will not be any reactions [13]. It uncertain which limit is reached first.

5 Experiments

5.1 Macroscopic diffusion in a cube

In this section, we measure the error in the solution caused by errors in the diffusion jump coefficients. In [24] the FVM discretization of the Laplace operator is proved to be inconsistent on unstructured meshes. In the FEM discretization we have to zero out negative off-diagonal elements in the stiffness matrix, which also makes the discretization inconsistent where the mesh is sub-optimal.
We study the impact of the inconsistencies by using these discretizations of the diffusion equation on an unstructured tetrahedral mesh generated by COMSOL. We integrate it deterministically on the macroscopic level to avoid the stochastic noise and slow convergence of Monte Carlo methods. Our model problem is

\[
\dot{\phi} = \gamma \Delta \phi, \quad \text{r} \in \Omega, t > 0,
\]

\[
n \cdot \nabla \phi = 0, \quad \text{r} \in \partial \Omega, t > 0,
\]

\[
\phi = 1 + \cos(2\pi x) \cos(2\pi y) \cos(2\pi z), \quad \text{r} \in \Omega, t = 0,
\]

where \( \Omega = [0,1]^3 \), \( \gamma = 10^{-3} \text{ m}^2/\text{s} \). It has the analytical solution

\[
\phi(\text{r}, t) = 1 + \cos(2\pi x) \cos(2\pi y) \cos(2\pi z)e^{-12\pi^2\gamma t}.
\]

It is integrated to \( t = 1 \) with the Crank-Nicolson method [2] with a time step \( k = 0.01 \). The temporal error is then small, using an even smaller time step has negligible impact on the solution. Equation (21) is discretized in space to

\[
\dot{\phi} = \gamma D \phi.
\]

Figure 3: \( \ell_2 \)-error for the different discretizations of the diffusion equation in a cube, as function of the number of nodes in the mesh. Note how FVM and truncated FEM level out for fine grids. The dashed line shows the slope for second order convergence.
In the truncated FEM discretization \( D = A^{-1} \tilde{S} \), and in the FVM discretization \( D = A^{-1} \hat{S} \). For reference we also do simulation with the consistent mass-lumped FEM discretization, where \( D = A^{-1} S \), and with a finite difference method on a Cartesian mesh. Errors are measured in \( \ell_2 \) and \( \ell_\infty \), with norms defined as

\[
\|\phi\|_2^2 = \sum_j \phi_j^2 |C_j|, \quad \|\phi\|_\infty = \max_j |\phi_j|.
\]

(24)

The finite difference method in time used was Crank-Nicolson also for the Cartesian mesh, with the normal second-order discretization of the Laplace operator. The homogeneous Neumann boundary condition was implemented using ghost points outside the domain. The ghost points were assumed to have the same values as their neighbours on the boundary. The mesh had a fixed step size, equal in all directions.

The convergence rates in \( \ell_2 \) are illustrated in Figure 3, where we can see how all methods adhere to second order convergence for coarse grids. For fine grids the convergence levels out for both the truncated FEM and FVM. This is expected since they are not consistent with the equation. It is also worth noting that the error on the Cartesian mesh is about one order of magnitude smaller than for a FEM approximation with the same number of degrees of freedom. Errors in \( \ell_2 \) and \( \ell_\infty \) for the simulations with the truncated FEM and FVM discretizations are listed in Table 1. Note that around 20% of the non-zero off-diagonal entries in the stiffness matrix came out negative, and were zeroed out. The errors for the reference simulations with consistent mass-lumped FEM are listed in Table 2, as are those for finite differences on a Cartesian mesh. All stated errors are absolute. They are, however, very close to the relative error, since the continuous \( L_2 \)-norm of the analytical solution is

\[
\|\phi\|_{L_2} = \sqrt{1 + \frac{1}{5} e^{-24\pi^2/\gamma}} \approx 1.05.
\]

We studied the spatial distribution of the error, and compared it to the quality of tetrahedra. The error is generally biggest close to the boundaries, and the mesh generators also have the most trouble generating well shaped tetrahedra there. But we have not been able to find any closer correlation between local tetrahedra shapes and the error. The diffusive character of the equation makes compound effects important, the error depends on the shape of many nearby tetrahedra, and it is difficult to draw any strict conclusions.

### 5.2 Macroscopic diffusion in a sphere

In this section, the experiment from Section 5.1 is repeated in a different domain. A sphere is used instead of a cube. This is perhaps more close to the realistic case, since living cells typically have smooth boundaries. The problem studied is

\[
\begin{align*}
\dot{\phi} &= \gamma \Delta \phi, & r &\in \Omega, t > 0, \\
\mathbf{n} \cdot \nabla \phi &= 0, & r &\in \partial \Omega, t > 0, \\
\phi &= \phi_0, & r &\in \Omega, t = 0,
\end{align*}
\]

(25)

where \( \Omega = \{(r, \varphi, \theta) \mid 0 \leq r \leq r_0, 0 \leq \varphi \leq 2\pi, 0 \leq \theta \leq \pi\} \), \( \phi_0 = \frac{1}{2} + \left( \frac{\sin(\alpha r)}{\alpha r} - \frac{\cos(\alpha r)}{\alpha} \right) \cos \theta \), \( \gamma = 0.1 \), \( r_0 = \frac{3}{\sqrt{\pi}} \), so that \( |\Omega| = 1 \), and \( \alpha \approx 3.355 \) so
Table 1: Errors as compared to the analytical solution for solutions of the diffusion equation in a cube. The solutions are for the FEM discretization with lumped mass matrix and truncation of negative off-diagonal elements in the stiffness matrix, and for the FVM discretization. $\% < 0$ denotes the fraction of non-zero off-diagonal entries that were negative. Note how the $\ell_2$-error levels out at $h_{\text{max}} \approx 0.2$, and the $\ell_\infty$-error is quite constant.

<table>
<thead>
<tr>
<th>$h_{\text{max}}$</th>
<th>$K$</th>
<th>$% &lt; 0$</th>
<th>$\ell_2$-error Truncated-FEM</th>
<th>$\ell_\infty$-error Truncated-FEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8</td>
<td>35</td>
<td>19.5</td>
<td>0.049886 0.060118</td>
<td>0.062861 0.083472</td>
</tr>
<tr>
<td>0.6</td>
<td>55</td>
<td>23.0</td>
<td>0.020692 0.027772</td>
<td>0.057441 0.071183</td>
</tr>
<tr>
<td>0.4</td>
<td>103</td>
<td>20.9</td>
<td>0.015434 0.017982</td>
<td>0.060318 0.063162</td>
</tr>
<tr>
<td>0.2</td>
<td>517</td>
<td>20.3</td>
<td>0.0076388 0.0050587</td>
<td>0.062744 0.025518</td>
</tr>
<tr>
<td>0.1</td>
<td>3444</td>
<td>18.4</td>
<td>0.0068282 0.0076041</td>
<td>0.049759 0.069093</td>
</tr>
<tr>
<td>0.08</td>
<td>6458</td>
<td>18.2</td>
<td>0.0064264 0.0073339</td>
<td>0.052204 0.063490</td>
</tr>
</tbody>
</table>

Table 2: Errors as compared to the analytical solution for the consistent mass-lumped FEM and finite difference discretizations of the diffusion equation in a cube. Note how the errors decay nicely in $\ell_2$ and $\ell_\infty$, except for some initial hesitation in the FEM case.

<table>
<thead>
<tr>
<th>$h_{\text{max}}$</th>
<th>$K$</th>
<th>$\ell_2$-error</th>
<th>$\ell_\infty$-error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8</td>
<td>35</td>
<td>0.051390 0.069782</td>
<td>1 8 0.0055027 0.0055027</td>
</tr>
<tr>
<td>0.6</td>
<td>55</td>
<td>0.025364 0.065286</td>
<td>0.333 64 0.0027698 0.0027698</td>
</tr>
<tr>
<td>0.4</td>
<td>103</td>
<td>0.017709 0.060457</td>
<td>0.143 512 0.00059582 0.00059582</td>
</tr>
<tr>
<td>0.2</td>
<td>517</td>
<td>0.0080449 0.060457</td>
<td>0.0667 4096 0.00013078 0.00013078</td>
</tr>
<tr>
<td>0.1</td>
<td>3444</td>
<td>0.0041003 0.023198</td>
<td>0.0526 8000 0.000082404 0.000082404</td>
</tr>
<tr>
<td>0.08</td>
<td>6458</td>
<td>0.0030147 0.018126</td>
<td></td>
</tr>
</tbody>
</table>

that the boundary condition is fulfilled. The analytical solution is then

$$\phi(r, t) = \frac{1}{2} + e^{-\gamma\alpha^2 t} \left( \frac{\sin(\alpha r)}{\alpha^2 r^2} - \frac{\cos(\alpha r)}{\alpha r} \right) \cos \theta.$$  \hspace{1cm} (26)

The system is integrated to $t = 1$ with the Crank-Nicolson method [2] with a time step $k = 0.01$. The temporal error is then small compared to the spatial error. As earlier, (25) is discretized in space to

$$\hat{\phi} = \gamma D\phi.$$  \hspace{1cm} (27)

In the FEM discretization $D = A^{-1}\hat{S}$, and in the FVM discretization $D = A^{-1}S$. For reference we also do simulation with the consistent mass-lumped FEM discretization, where $D = A^{-1}S$. Errors are measured in $\ell_2$ and $\ell_\infty$, with the norms defined in (24) as before.

The convergence rates in $\ell_2$ are illustrated in Figure 4. Note that neither the truncated FEM nor FVM appear to converge at all, their errors apparently leveled out already at the coarsest mesh used. Errors in $\ell_2$ and $\ell_\infty$ for the simulations with the truncated FEM and FVM discretizations are listed in Table 3. Note that just as for the cube, around 20% of the non-zero off-diagonal...
entries in the stiffness matrix came out negative, and were zeroed out. The errors for the reference simulations with consistent, mass-lumped FEM are listed in Table 4. The listed errors are all absolute, the relative errors are about twice as big. The continuous $L^2$-norm of the analytical solution is

$$\|\phi\|_{L^2} = \sqrt{\frac{2\pi^3}{3} + \frac{5}{4} e^{-2\gamma \alpha^2} \left( \frac{2\pi^2}{\alpha^2} + \frac{\sin(2\alpha r_0)}{\alpha r_0} - \frac{4\sin^2(\alpha r_0)}{\alpha^2 r_0} \right)} \approx 0.505.$$

5.3 Convergence of stochastic sampling

To verify that sampling with NSM converges to the macroscopic FEM and FVM solutions, we compare stochastic simulations with the solutions obtained in Section 5.1. Sampling is made both with the FEM and FVM discretizations on the mesh with 517 nodes. Each simulation start with $m$ molecules in the system distributed according to $\phi(x, y, z, 0)$ in (21). The nodal values of the initial condition are rounded to the nearest upper or lower integers with probability depending on the fractional parts. The analytical solution is scaled by the number of molecules in the system to

$$u(r, t) = m(1 + \cos(2\pi x) \cos(2\pi y) \cos(2\pi z) e^{-12\pi^2 \gamma t}).$$

The error as compared to the corresponding macroscopic solution is expected
Table 3: Errors for the truncated FEM and FVM discretizations of the diffusion equation in a sphere, as compared to the analytical solution. % < 0 denotes the fraction of non-zero off-diagonal entries that were negative.

<table>
<thead>
<tr>
<th>$h_{max}$</th>
<th>$K$</th>
<th>% &lt; 0</th>
<th>$\ell_2$-error</th>
<th>$\ell_\infty$-error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>93</td>
<td>19.5</td>
<td>0.0060407</td>
<td>0.014914</td>
</tr>
<tr>
<td>0.4</td>
<td>100</td>
<td>22.2</td>
<td>0.0045586</td>
<td>0.012202</td>
</tr>
<tr>
<td>0.3</td>
<td>198</td>
<td>20.4</td>
<td>0.0013071</td>
<td>0.00443</td>
</tr>
<tr>
<td>0.2</td>
<td>468</td>
<td>17.8</td>
<td>0.0068839</td>
<td>0.0032002</td>
</tr>
<tr>
<td>0.1</td>
<td>3433</td>
<td>19.6</td>
<td>0.00011391</td>
<td>0.00066905</td>
</tr>
<tr>
<td>0.09</td>
<td>4542</td>
<td>18.4</td>
<td>0.00011046</td>
<td>0.0005655</td>
</tr>
</tbody>
</table>

Table 4: Errors for the consistent mass-lumped FEM discretization of the diffusion equation in a sphere, as compared to the analytical solution.

<table>
<thead>
<tr>
<th>$h_{max}$</th>
<th>$K$</th>
<th>$\ell_2$-error</th>
<th>$\ell_\infty$-error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>93</td>
<td>0.0071676</td>
<td>0.017668</td>
</tr>
<tr>
<td>0.4</td>
<td>100</td>
<td>0.0056443</td>
<td>0.016337</td>
</tr>
<tr>
<td>0.3</td>
<td>198</td>
<td>0.0049008</td>
<td>0.011617</td>
</tr>
<tr>
<td>0.2</td>
<td>468</td>
<td>0.0023455</td>
<td>0.0054619</td>
</tr>
<tr>
<td>0.1</td>
<td>3433</td>
<td>0.00063668</td>
<td>0.0015666</td>
</tr>
<tr>
<td>0.09</td>
<td>4542</td>
<td>0.00048816</td>
<td>0.0014262</td>
</tr>
</tbody>
</table>

to behave as $O(m^{-1/2})$ in $\ell_2$ and $\ell_\infty$, in accordance with the results of Kurtz [16]. This is also what we observe. The errors are presented in Table 5 and Figure 5, where we can see them decay nicely at the expected rates. The errors presented are the average of the errors from 100 trajectories with the same configurations. The average is taken over the $\ell_2$ and $\ell_\infty$ errors, it is not the error of the average solution.

Table 5: The $\ell_2$-errors of the stochastic simulation using the FVM and FEM discretizations of the diffusion equation, as compared to the macroscopic solutions with the same diffusion matrices.

<table>
<thead>
<tr>
<th>$\log_{10} m$</th>
<th>$\ell_2$-error</th>
<th>$\ell_\infty$-error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FVM</td>
<td>FEM</td>
</tr>
<tr>
<td>3</td>
<td>0.49882</td>
<td>0.52086</td>
</tr>
<tr>
<td>4</td>
<td>0.14648</td>
<td>0.15703</td>
</tr>
<tr>
<td>5</td>
<td>0.046240</td>
<td>0.049112</td>
</tr>
<tr>
<td>6</td>
<td>0.014562</td>
<td>0.015547</td>
</tr>
</tbody>
</table>

5.4 Truncation of the stiffness matrix

Recall that the mass-lumped FEM discretization of the diffusion equation is

$$\dot{\phi} = \gamma D\phi, \quad D = A^{-1}S.$$ \hspace{1cm} (29)

This is a consistent $O(h^2)$ approximation, but $\gamma D^T$ does not in general fulfill
The error measured in $\ell_2$ of the stochastic simulation.

The error measured in $\ell_\infty$ of the stochastic simulation.

Figure 5: Convergence towards the macroscopic solution in $\ell_2$ and $\ell_\infty$ of the stochastic sampling. The dashed line indicates order $\frac{1}{2}$ convergence.

the property (8) which is necessary for a probabilistic interpretation. This is remedied by introducing the truncated diffusion matrix, where the negative off-diagonal elements are nulled out, and the diagonal is adjusted so that the property (9) is preserved. Denote the truncated diffusion matrix by $D = D + D_\epsilon$. The correction $D_\epsilon$ is strongly dependent on the quality of the mesh. Introduce the weighted Frobenius norm

$$\|D\|_F^2 = \sum_{i,j} |C_i| D_{ij}^2.$$ (30)

Ideally the correction would vanish as the mesh is refined, at least in comparison to the real diffusion matrix $D$. This is unfortunately not the case for meshes generated by COMSOL, at least not in general. This is demonstrated with the recently studied cube as counterexample. The norms of the matrices $D$ and $D_\epsilon$ are studied on a sequence of meshes. The norms are listed in Table 6 and illustrated in Figure 6. It can be seen that both $\|D\|_F$ and $\|D_\epsilon\|_F$ grow approximately as $O(h^{-2})$. $D_\epsilon$ is declining in comparison to $D$, but not close to at an acceptable rate. The correction is in the Frobenius norm about one order of magnitude smaller than the diffusion matrix itself.

Table 6: Norms of the matrices $D$ and $D_\epsilon$. Note that the correction decline in relation to the consistent diffusion matrix, but alas how slowly.

<table>
<thead>
<tr>
<th>$h_{max}$</th>
<th>$K$</th>
<th>$% &lt; 0$</th>
<th>$|D|_F$</th>
<th>$|D_\epsilon|_F$</th>
<th>$|D_\epsilon|_F/|D|_F$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6</td>
<td>55</td>
<td>23.0</td>
<td>58.22</td>
<td>6.9811</td>
<td>0.12</td>
</tr>
<tr>
<td>0.4</td>
<td>103</td>
<td>20.9</td>
<td>90.236</td>
<td>9.6825</td>
<td>0.11</td>
</tr>
<tr>
<td>0.2</td>
<td>517</td>
<td>20.3</td>
<td>320.933</td>
<td>27.666</td>
<td>0.086</td>
</tr>
<tr>
<td>0.1</td>
<td>3444</td>
<td>18.4</td>
<td>1234.7</td>
<td>91.045</td>
<td>0.074</td>
</tr>
<tr>
<td>0.08</td>
<td>6458</td>
<td>18.2</td>
<td>1906.0</td>
<td>136.55</td>
<td>0.072</td>
</tr>
<tr>
<td>0.06</td>
<td>14772</td>
<td>18.0</td>
<td>3374.9</td>
<td>234.70</td>
<td>0.070</td>
</tr>
<tr>
<td>0.04</td>
<td>48833</td>
<td>17.7</td>
<td>7618.9</td>
<td>522.88</td>
<td>0.069</td>
</tr>
<tr>
<td>0.02</td>
<td>379928</td>
<td>17.5</td>
<td>30449</td>
<td>2029.6</td>
<td>0.067</td>
</tr>
</tbody>
</table>
5.5 Comparison of mesh generators

In this section, the experiments from Section 5.1 are repeated on different sets of meshes. The first set was generated with the open source software Gmsh 2.4.2 [8], and obey the empty sphere criterion of Delaunay meshes [7]. Gmsh proved not to be a reliable Delaunay mesh generator. For some of the resolutions used, a lot of fiddling was required in order to obtain a mesh fulfilling the empty sphere criterion. Moreover, the meshes produced were of quite modest quality. The rate of negative off-diagonal elements in the stiffness matrix was generally higher than for meshes generated by COMSOL. This was also reflected in the simulation results. The errors were consistently higher than for the corresponding simulations on non-Delaunay meshes generated by COMSOL. The errors for the truncated FEM and FVM are presented in Table 7, and the errors for a reference solution with consistent, mass-lumped FEM in Table 8. It is all illustrated in Figure 7. Note how jagged the curves are, indicating varying quality of the different meshes. Delaunay meshes enjoy an optimality property for meshes on a given set of nodes, but put little constraint on the positioning of the nodes themselves. Our results indicate that COMSOL is more successful in positioning the nodes, and to an extent that makes that more important than the Delaunay property.
It is also worth noting that the use of Delaunay meshes opens up an opportunity for a different type of FVM. From a Delaunay mesh, a dual Voronoï mesh can be constructed, and there exist convergence results for FVM adapted to Voronoï meshes [21]. This was not exploited, the FVM defined in Section 3.2 was used also on the Delaunay meshes. Using a Voronoï-based FVM might be a good idea, but that is outside the scope of this report.

We also tried to derive a Delaunay triangulation from the sets of points generated by COMSOL, but without success. Difficulties were encountered on the boundary, where tetrahedra with all nodes lying in or almost in a plane were constructed. The circumcircle of such a tetrahedron will be huge, but may lie almost completely outside the domain, and thus not break the Delaunay property. This is still not acceptable, since the corresponding subvolumes would be orders of magnitude bigger than the computational domain Ω. In order to construct a reasonable Voronoï mesh one would have to modify the boundary nodes. This was not attempted, but there exists software that claims to have this capability. TetGen [23] is one such software.

Finally we tried improving the meshes generated by COMSOL with the mesh
Table 7: Errors as compared to the analytical solution for the truncated FEM and FVM discretizations on Delaunay meshes from Gmsh. \( \% < 0 \) denotes the fraction of non-zero off-diagonal entries that were negative. Note how the \( \ell_2 \) error levels out earlier than for the meshes generated by COMSOL (see Table 1).

<table>
<thead>
<tr>
<th>( K )</th>
<th>( % &lt; 0 )</th>
<th>( \ell_2 )-error</th>
<th>( \ell_\infty )-error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Truncated FEM</td>
<td>FVM</td>
<td>Truncated FEM</td>
</tr>
<tr>
<td>87</td>
<td>12.5</td>
<td>0.037003</td>
<td>0.035902</td>
</tr>
<tr>
<td>122</td>
<td>27.0</td>
<td>0.052774</td>
<td>0.020627</td>
</tr>
<tr>
<td>143</td>
<td>18.8</td>
<td>0.047618</td>
<td>0.025553</td>
</tr>
<tr>
<td>275</td>
<td>20.6</td>
<td>0.022818</td>
<td>0.008561</td>
</tr>
<tr>
<td>1300</td>
<td>22.6</td>
<td>0.027704</td>
<td>0.008010</td>
</tr>
<tr>
<td>2481</td>
<td>24.0</td>
<td>0.026806</td>
<td>0.009647</td>
</tr>
<tr>
<td>5044</td>
<td>24.8</td>
<td>0.025740</td>
<td>0.01049</td>
</tr>
</tbody>
</table>

Table 8: Errors as compared to the analytical solution for the consistent FEM discretization on Delaunay meshes from Gmsh. Note how the errors decay nicely both in \( \ell_2 \) and \( \ell_\infty \).

<table>
<thead>
<tr>
<th>( K )</th>
<th>( \ell_2 )-error</th>
<th>( \ell_\infty )-error</th>
</tr>
</thead>
<tbody>
<tr>
<td>87</td>
<td>0.054681</td>
<td>0.23149</td>
</tr>
<tr>
<td>122</td>
<td>0.087662</td>
<td>0.40400</td>
</tr>
<tr>
<td>143</td>
<td>0.032710</td>
<td>0.099270</td>
</tr>
<tr>
<td>275</td>
<td>0.034022</td>
<td>0.20180</td>
</tr>
<tr>
<td>1300</td>
<td>0.014965</td>
<td>0.079190</td>
</tr>
<tr>
<td>2481</td>
<td>0.0092434</td>
<td>0.037865</td>
</tr>
<tr>
<td>5044</td>
<td>0.0060818</td>
<td>0.047412</td>
</tr>
</tbody>
</table>

improvement program Stellar [15]. Stellar managed to make the meshes better. The fraction of non-zero off-diagonal elements that were negative reduced with between two and five percentage points, except for the coarsest mesh (35 nodes) where the fraction stayed unchanged, as seen in Table 9. This did, however, have little impact on the solution quality. The convergence curves, on display in Figure 8, leveled out in a way much similar to what we have seen before.

6 The MinD/E system of E. coli

In this section we will apply the methods developed to a biologically relevant problem, which has been much studied analytically [20], experimentally [22] and computationally [12] during the last decade. We will follow the problem setup used by Fange and Elf in [6], where the RDME is sampled on a Cartesian mesh. The Min system of the Escherichia coli bacterium directs cell division to the middle of the bacterium. It consists of the three proteins MinC, MinD and MinE, which inhibit polymerization of the protein FtsZ. Formation of the FtsZ polymer is the first step of the creation of a membrane separating the two halves of the bacterium. By oscillating between the endpoints of the bacterium, the Min proteins prevent non-symmetric cell division. In the inter-regulation of the Min system, MinC is considered solely dependent. It is important for
inhbiting FtsZ polymerization, but does not affect the behaviour of the Min system itself. It can therefore be disregarded in this model. MinD and MinE are closely interdependent. The Min proteins may bind to the cell membrane, which is an important process in the system. In this model a membrane-bound protein is considered a different molecular species than its cytosolic counterpart. The reduced Min system studied consists of five molecular species which are involved in five reactions. The species are

\[
\begin{align*}
\text{MinD}^{\text{ATP}}_{\text{cyt}} & \quad \text{MinD–ATP complex in the cytosol,} \\
\text{MinD}^{\text{ADP}}_{\text{cyt}} & \quad \text{MinD–ADP complex in the cytosol,} \\
\text{MinE}_{\text{cyt}} & \quad \text{MinE molecule in the cytosol,} \\
\text{MinD}^{\text{ATP}}_{\text{mem}} & \quad \text{membrane-bound MinD–ATP complex, and} \\
\text{MinD}^{\text{ATP}}_{\text{mem}} & \quad \text{membrane-bound MinD–MinE–ATP complex.}
\end{align*}
\]

Cytosolic \( \text{MinD}^{\text{ATP}} \) may bind to the membrane. A membrane-bound \( \text{MinD}^{\text{ATP}} \) catalyzes the binding of further \( \text{MinD}^{\text{ATP}} \) molecules to the membrane. Cytosolic MinE may bind to a membrane-bound \( \text{MinD}^{\text{ATP}} \) molecule, forming a \( \text{MinD}^{\text{ATP}}–\text{MinE} \) complex. A membrane-bound \( \text{MinD}^{\text{ATP}}–\text{MinE} \) complex may leave the membrane during splitting of the proteins and hydrolysis of ATP to ADP, releasing \( \text{MinD}^{\text{ADP}} \), MinE and a phosphate group in the cytosol. Finally \( \text{MinD}^{\text{ADP}} \) may absorb a phosphate group, reconstructing \( \text{MinD}^{\text{ATP}} \).
Table 9: Errors as compared to the analytical solution for the truncated FEM and FVM discretizations on meshes generated by COMSOL and improved by Stellar. $\% < 0$ denotes the fraction of non-zero off-diagonal entries that were negative. Note how the $\ell_2$-error levels out at approximately the same level as when Stellar is not used (see Table 1).

<table>
<thead>
<tr>
<th>$h_{\text{max}}$</th>
<th>$K$</th>
<th>$% &lt; 0$</th>
<th>$\ell_2$-error</th>
<th>$\ell_\infty$-error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Truncated FEM</td>
<td>FVM</td>
</tr>
<tr>
<td>0.8</td>
<td>35</td>
<td>19.5</td>
<td>0.049886</td>
<td>0.060118</td>
</tr>
<tr>
<td>0.6</td>
<td>60</td>
<td>18.2</td>
<td>0.03089</td>
<td>0.031981</td>
</tr>
<tr>
<td>0.4</td>
<td>116</td>
<td>16.2</td>
<td>0.018767</td>
<td>0.020267</td>
</tr>
<tr>
<td>0.2</td>
<td>598</td>
<td>16.6</td>
<td>0.0087937</td>
<td>0.009123</td>
</tr>
<tr>
<td>0.1</td>
<td>3806</td>
<td>16.3</td>
<td>0.0064795</td>
<td>0.009123</td>
</tr>
<tr>
<td>0.08</td>
<td>7048</td>
<td>15.6</td>
<td>0.0064675</td>
<td>0.010109</td>
</tr>
</tbody>
</table>

The reaction formulas are

$$\text{MinD}^{\text{ATP}}_{\text{cyt}} \xrightarrow{k_d} \text{MinD}^{\text{ATP}}_{\text{mem}}, \quad (31)$$

$$\text{MinD}^{\text{ATP}}_{\text{cyt}} + \text{MinD}^{\text{ATP}}_{\text{mem}} \xrightarrow{k_{dD}} 2\text{MinD}^{\text{ATP}}_{\text{mem}}, \quad (32)$$

$$\text{MinE}_{\text{cyt}} + \text{MinD}^{\text{ATP}}_{\text{mem}} \xrightarrow{k_{de}} \text{MinD}^{\text{ADP}}_{\text{cyt}} + \text{MinE}_{\text{cyt}}, \quad (33)$$

$$\text{MinD}^{\text{ADP}}_{\text{cyt}} \xrightarrow{k_{ATP}} \text{MinD}^{\text{ATP}}_{\text{cyt}}, \quad (35)$$

with the reaction rate constants $k_d = 1.25 \cdot 10^{-8}$ ms$^{-1}$, $k_{dD} = 5.56 \cdot 10^7$ m$^3$mol$^{-1}$s$^{-1}$, $k_{de} = 5.56 \cdot 10^7$ m$^3$mol$^{-1}$s$^{-1}$, $k_c = 0.7$ s$^{-1}$, and $k_{ATP} = 0.5$ s$^{-1}$. The diffusion constant is $\gamma_{\text{cyt}} = 2.5 \cdot 10^{-12}$ m$^2$/s in the cytosol and $\gamma_{\text{mem}} = 1.0 \cdot 10^{-14}$ m$^2$/s on the membrane. The computational domain $\Omega$ is a cylinder of length 3.5 $\mu$m with half-sphere caps of radius 0.5 $\mu$m on its ends. As initial condition, 2000 $\text{MinD}^{\text{ATP}}_{\text{cyt}}$, 2000 $\text{MinD}^{\text{ADP}}_{\text{cyt}}$ and 1000 $\text{MinE}_{\text{cyt}}$ are distributed over the domain according to a uniform probability distribution. The model is, except for the meshing, identical to the one of the wild type bacterium in [6].

We studied the model by sampling single trajectories on a set of unstructured meshes with varying resolution, using both FEM and FVM discretizations for the diffusion jump coefficients. The meshes had 116, 168, 318, 402, 603, 1203, 3183, 8283 and 14347 subvolumes. The system was simulated during fifteen minutes, and the state was sampled each second. Two properties of the membrane-bound $\text{MinD}$ were studied more closely: the period of oscillations and the time-averaged concentration. These properties measured from simulations on two of the meshes are shown in Figure 9 and Figure 10. The period was calculated with a discrete Fourier transform, where the first minute was disregarded as being a start-up phase. We observed a distinct peak at periods of between 30 and 32 seconds in all simulations. This is in line with the observations in [6], where periods of 31 seconds were observed. The result of the Fourier transform has an approximate resolution of one second. The local concentration was calculated as the copy-number of $\text{MinD}_{\text{mem}}$ in the subvolume divided by the volume of it.

In [6, Figure 4], time-averaged concentration of membrane-bound $\text{MinD}$ as
Figure 9: Single trajectories of the system were sampled with diffusion coefficients from FEM and FVM on a mesh with 1203 subvolumes. Subfigures (b) and (d) show the copy numbers of membrane-bound MinD proteins in one half of the cell oscillating over time in the FEM-based and FVM-based simulations respectively.
Figure 10: Repetition of the experiments presented in Figure 9 on a finer mesh, here with 14347 subvolumes. Subfigures (b) and (d) show the copy numbers of membrane-bound \textit{MinD} proteins in one half of the cell oscillating over time in the FEM-based and FVM-based simulations respectively.
function of the coordinate along the long axis of the bacterium is presented. The figure was produced by stochastic sampling of a RDME on a Cartesian mesh using the software MesoRD [11]. Our simulations show a similar pattern, but the curves are not as smooth. To study this difference more closely, we repeated their experiments with MesoRD on three Cartesian meshes of different resolution. The subvolumes were cubic with side lengths 0.2, 0.1 and 0.05 µm, the meshes had 512, 3760 and 32496 subvolumes respectively. Before studying the results, we need to explain the different handling of boundaries in the URDME and MesoRD models of the Min system.

The membrane-bound Min proteins diffuse on the membrane. There is no standardized way of handling this kind of diffusion on curved surfaces, neither in URDME nor in MesoRD. It can only be accomplished by modifying the model files.

In URDME, membrane-bound proteins are modeled as different molecular species than their cytosolic counterparts. For these species, the diffusion jump coefficients of cytosolic subvolumes are zeroed out so that they can only move within the subvolumes neighbouring the boundary. The diffusion on the membrane is though still modeled as three-dimensional diffusion, the jump coefficients depend on the shapes of the subvolumes. It would have been more accurate to calculate them from the two-dimensional boundary mesh.

In MesoRD the domain is divided into two compartments, the cytosol and the membrane. The membrane is a shell around the cytosol. It is thin but has a certain thickness, so that diffusion on the membrane is three-dimensional. Each molecular species is restricted to one compartment, i.e. also here membrane-bound proteins are modeled as different species than their cytosolic counterparts. With this membrane modeling, two problems can occur. If the membrane is too thin membrane subvolumes may become disconnected, and if it is too thick the third dimension becomes a problem. It is very difficult, sometimes impossible, to find a membrane thickness where neither of these phenomena appear somewhere in the mesh.

![Figure 11](image_url)

Figure 11: A cross-section of the MesoRD mesh with subvolumes with 0.05 µm sides. Cytosolic subvolumes are marked with points and membrane subvolumes with crosses.
In the used MesoRD model of the Min problem, disconnected sections of the membrane is a significant problem unless the resolution is high. The cytosolic and membrane subvolumes in a cross-section of the finest mesh used is shown in Figure 11. Since molecules can only jump between subvolumes with a common face, the membrane is divided into disconnected sections. In the coarser meshes, all membrane subvolumes in the cross-section are disconnected. This leads to one-dimensional diffusion - molecules on the cylindrical region of the membrane can only move parallel to the long axis. On the half-spherical caps at the ends of the bacterium, membrane subvolumes become completely disconnected.

In Figure 12, the time-averaged concentration of membrane-bound $MinD$, as function of the length coordinate of the bacterium. The average is taken over 25 oscillation periods from trajectories calculated with FEM (plain line) and FVM (dashed line) based URDME, and with MesoRD (line with asterisks). The curves are linearly scaled to $[-1,1] \times [0,1]$.

Figure 12: Time-averaged concentration of membrane-bound $MinD_{\text{mem}}$ is plotted against the coordinate on the long axis of the bacterium. The plot shows the average concentrations over 25 oscillation periods, calculated with URDME using FEM and FVM based diffusion on an unstructured mesh with 3183 subvolumes, and with MesoRD using the mesh with 3760 subvolumes. The URDME meshes had elements of sizes ranging between 0.09 and 0.2 µm, and the MesoRD mesh had a fixed step size of 0.1 µm. For the MesoRD curve, a spatial average is taken over all subvolumes with the same coordinate on the long axis (46 different coordinates). In the URDME curves, the long axis is divided into 40 sub-intervals which are averaged over. The only distinct effect of the above mentioned discretization problems is the dip of the MesoRD curve close to the ends of the
Figure 13: Time-averaged concentration of membrane-bound MinD, as function of the length coordinate of the bacterium. The curves are calculated using FEM and FVM based URDME, and with MesoRD, using higher resolution than used in Figure 12.

We believe that this is caused by membrane subvolumes being disconnected. Also note the difference in level between the URDME and MesoRD curves. Figure 13 shows a repetition of the experiment on meshes with higher resolution. For the URDME curves we used a mesh with 14347 subvolumes, for MesoRD the mesh had 32496 subvolumes. The URDME meshes had elements of sizes ranging between 0.05 and 0.13 µm, and the MesoRD mesh had a fixed step size of 0.05 µm. Note how the difference in level between the curves almost disappeared. The dips of the MesoRD curve at the ends are also gone. The membrane subvolumes being connected to a higher extent in this mesh is a likely cause for this.

We believe that the differences between the curves for the coarser meshes originate from the different approaches to boundary treatment. But it is hard to say that one approach should be more correct, or if you wish less wrong, than the other. The discretization errors of URDME do not show in an obvious way neither in this plot, nor in more detailed boundary plots as those in Figure 9 and Figure 10.

7 Conclusions

The FEM-discretization of the diffusion equation used in URDME suffers from convergence problems. As an alternative, we have implemented a node-centered
FVM and compared the convergence properties of the two methods. We could conclude that the FVM used was no better than the previous FEM approach.

It was verified that the stochastic sampling converges to the deterministic solution obtained by numerical integration of the diffusion equation, in accordance with Kurtz’s theorem. Then deterministic convergence studies were carried out, comparing the two methods. Experiments were made on two qualitatively different geometries and meshes from three different generators. The message was clear, there is no significant difference in error between the two methods, and changing the mesh generator does not make much difference either, at least not for the better. One may discuss how serious this is. The convergence curves level out at errors of a few percent, which in most applications is less than the stochastic noise. The errors in the diffusion coefficients should not pose any serious risk of not reproducing the qualitative behaviour of the system. On the other hand, the convergence curves were found to level out at quite coarse meshes, sometimes just fine enough to give a reasonable representation of the geometry. It is important to keep in mind that the model breaks down if the subvolumes become too small, since the subvolume boundaries truncate the reaction radii of the molecules. This limit is, naturally, problem dependent and quite poorly understood. This makes it difficult to estimate the severity of the convergence problems of the diffusion modeling. The impact they have on biologically relevant problems ought to be further investigated.

The experiments made with the E. coli model indicate that URDME is good enough to reliably reproduce the macroscopic, qualitative behaviour of a cellular chemical system. The oscillations were present and had very similar period times on all meshes tested, despite the heavy variation in resolution. One should though take into account that biological chemical systems often are quite robust, since the life of the organism may depend on them. The discretization errors of the diffusion seemed to have limited impact on the solution, but it should be noted that it is difficult to measure due to the stochastic effects and the lack of an analytical solution. The way the boundary was treated seems to be a bigger source of error.

Despite that the studied finite volume discretization did not improve the accuracy, it is our belief that a consistent discretization can be made with a FVM. A quite easy choice would be to use a cell-centered Voronoï-based FVM. In [21] such a method is proved to converge for an elliptic convection-diffusion equation. It is not that far-fetched to expect similar properties for the time-dependent problem. The main difficulty lies in mesh generation. COMSOL is not capable of generating tetrahedral Delaunay meshes. The freely available software Gmsh is, but its meshes are not of the quality one gets from COMSOL. A possible solution would be to take the points from a COMSOL mesh, and generate the Delaunay and Voronoï meshes from that, for instance with TetGen [23]. Some interfacing will be needed to get the components to cooperate. Another solution could be to use a non-linear FVM. In [18] such a method is presented and proved to converge for the elliptic problem. The elements of the stiffness matrix are shown to obey the sign constraints we need. The disadvantage with this method is that the stiffness matrix depends on the solution, and thus has to be recalculated during the integration of the parabolic problem, which imposes a penalty in computational work. These two methods will be studied in subsequent publications. Also the treatment of diffusion along curved surfaces would be an interesting object of future research.
Acknowledgements

We would like to thank our supervisors Andreas Hellander and Per Löststedt for their guidance, encouragement and good ideas. We are also grateful to David Fange for generously letting us run MesoRD on his machine.

References


27
A The COMSOL fem struct

It is possible to export a COMSOL model to MATLAB as a MATLAB struct, typically named fem. URDME queries COMSOL for the FEM mass and stiffness matrices of a model in fem-struct representation. COMSOL has no support
for FVM, so the FVM stiffness matrix has to be assembled in separate code using mesh information and diffusion coefficients stored in the \texttt{fem-struct}. The purpose of this section is to explain where this information can be found, for future reference.

Mesh information is stored in \texttt{fem.xmesh}, and can be accessed by calls to the MATLAB function \texttt{xmeshinfo}, supplied by COMSOL. The stiffness matrix is a \(Ndofs \times Ndofs\) matrix, where \(Ndofs = Nnodes \times Nspecies\) is the number of DOFs in the system. The row and column indices correspond to DOF numbers specified by COMSOL. Mesh information sufficient for assembling the stiffness matrix is given by three matrices: \(p\), which gives the coordinates of the nodes, \(t\), which gives the node numbers of the vertices of the elements, and \(d\), which gives the DOF numbers of the vertices of the elements. The matrices are retrieved by the following code snippet.

\begin{verbatim}
p = xmeshinfo(fem, 'out', 'nodes');
p = p.coords;
t = xmeshinfo(fem, 'out', 'elements', 'meshtype', 'tet');
% In 2D, change 'tet' for 'tri'.
d = t.dofs;
t = t.nodes;
\end{verbatim}

\(p\) is a \(2 \times Nnodes\) matrix in 2D, and a \(3 \times Nnodes\) matrix in 3D. Column \(j\) gives the coordinates of node \(j\), which is the centroid of the cell \(C_j\). \(t\) is a \(3 \times Ndofs\) matrix in 2D, and a \(4 \times Ndofs\) matrix in 3D. Column \(k\) gives the column indices in \(p\) for the vertices of the triangle or tetrahedron \(T_k\). \(d\) is a \(3 \times Nspecies \times Ndofs\) matrix in 2D, and a \(4 \times Nspecies \times Ndofs\) matrix in 3D. Column \(l\) gives the DOF numbers for the vertices of the triangle or tetrahedron \(T_l\). Internally the columns are grouped by species, i.e. the first 3 (4) rows give the DOF numbers for all vertices in the triangle (tetrahedron) for the first species. The DOF numbers determine positions in the stiffness matrix. The jump coefficient for diffusion from DOF \(k\) to DOF \(j\) is given by \(\gamma D_{kj}\).

To assemble the stiffness matrix for FVM we need the macroscopic diffusion coefficients for different species. They are found in the \texttt{Application-struct} struct inside of the \texttt{fem-struct}, under \texttt{fem.appl\{1\}.equ.D}. The \texttt{equ-field} is for sub domain coefficient/application data. For each of the species \(D\) is either a number or a constant. The value of the symbolic constant is found under \texttt{fem.const}. If we have more than one species \(D\) is an array of \{\texttt{Nx1 cells}\} where each cell is either a number or a constant and \(N\) is the number of species.