Journal of Computational Physics 229 (2010) 343–360

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

An adaptive algorithm for simulation of stochastic reaction–diffusion processes

Lars Ferm, Andreas Hellander, Per Lötstedt *,[1]

Division of Scientific Computing, Department of Information Technology, Uppsala University, P.O. Box 337, SE-75105 Uppsala, Sweden

ARTICLE INFO

Article history:
Received 15 April 2009
Received in revised form 22 September 2009
Accepted 23 September 2009
Available online 30 September 2009

MSC:
65C40
65C05
60H35

PACS:
02.50.Ga
02.50.Ey
02.70.Uu
83.10.Rs
87.10.Rt

Keywords:
Master equation
Chemical reactions
Diffusion
Hybrid method
Adaptivity
URDME

ABSTRACT

We propose an adaptive hybrid method suitable for stochastic simulation of diffusion dominated reaction–diffusion processes. For such systems, simulation of the diffusion requires the predominant part of the computing time. In order to reduce the computational work, the diffusion in parts of the domain is treated macroscopically, in other parts with the tau-leap method and in the remaining parts with Gillespie's stochastic simulation algorithm (SSA) as implemented in the next subvolume method (NSM). The chemical reactions are handled by SSA everywhere in the computational domain. A trajectory of the process is advanced in time by an operator splitting technique and the timesteps are chosen adaptively. The spatial adaptation is based on estimates of the errors in the tau-leap method and the macroscopic diffusion. The accuracy and efficiency of the method are demonstrated in examples from molecular biology where the domain is discretized by unstructured meshes.

1. Introduction

The number of molecules of each species in a biological cell is often small and a mesoscopic, stochastic model for the chemical reactions is necessary to explain experimental data [5,41]. The macroscopic, deterministic equation for the concentrations of the chemical species is the reaction rate equation (RRE). This is an accurate model when the copy numbers are large but this is often not the case e.g. in the nucleus of a cell. Many computational methods have been developed in the

* Corresponding author. Tel.: +46 18 4712972; fax: +46 18 523049.
E-mail addresses: ferm@it.uu.se (L. Ferm), andreas.hellander@it.uu.se (A. Hellander), perl@it.uu.se (P. Lötstedt).

[1] Financial support has been obtained from the Swedish Foundation for Strategic Research and the Swedish National Graduate School in Mathematics and Computing.

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.09.030

last decade for the well stirred mesoscopic, stochastic problem when the distribution of the species in space is ignored. Recently, methods for the space dependent case have appeared.

Continuous time discrete (state) space Markov processes are well established as a mathematical tool to analyze the behavior of biochemical reaction networks in systems biology. Most models assume that the system is well stirred and that the model can be analyzed by solving the chemical master equation (CME) for the probability density function (PDF) or, if the dimension of the model is too high, by simulation of the process with e.g. the stochastic simulation algorithm (SSA) [25]. However, there are scenarios where diffusive transport needs to be included in the model [16,22,42]. If the spatial distribution of molecules is important, diffusion can be accounted for by discretizing the domain and allowing species to jump between adjacent computational cells (or subvolumes, compartments, voxels) [23,33]. Chemical reactions occur between the molecules in each subvolume as in the well stirred case and in this setting the PDF is the solution of the reaction–diffusion master equation (RDME). Also in this case the system can be simulated by a stochastic method [7,18,19,22,30,38,52]. Based on the next reaction method (NRM) [24], the next subvolume method (NSM) [18] is an efficient algorithm for simulation of reaction–diffusion processes and it has been implemented for Cartesian meshes in [28] and for general, unstructured meshes in [14]. Methods for stochastic reaction–diffusion models are compared in [4,15].

One challenging problem when simulating stochastic models in the well stirred case is stiffness. If a few of the reactions are very fast this leads to very small timesteps in the algorithm and frequent sampling of the fast reaction channels. Often, this is caused by some of the species being present in a much higher copy number than the others for which the stochastic fluctuations are less important. Due to this, it may be difficult to simulate the system on the time scale of the slower, often more interesting dynamics. This has led to the development of many approximate, hybrid and multiscale methods, for examples of these methods and reviews see e.g. [8,21,39,43]. The most popular approximate method in the well stirred case is the tau-leap method [26] and it has also been used for diffusion [46]. It approximates the number of events taking place in a time interval by a Poissonian random variable, and thus several events may be "leaped over" in one timestep.

For systems governed by the RDME, the situation can be even worse. High copy number species, possibly diffusing faster than some of the less abundant species, may render the system very stiff. Almost all events generated by the algorithm will be diffusion events occurring on a short time scale. This problem will inevitably arise if models become more detailed and explicitly include e.g. second messengers or small metabolites. If high concentrations are localized to some region in space and time, which may be the case in models of e.g. transient release of intracellular calcium pools or a step increase in second messenger concentration due to a transient stimulus, any method dealing with the stiffness needs to be adaptive in space and time.

Realizations of a process governed by a RDME are generated on an unstructured mesh in [19]. The diffusion coefficients in the RDME are derived from a finite element discretization of the diffusion operator. This discretization is also the macroscopic approximation of the diffusion. A hybrid method is proposed where the diffusion of some species is advanced macroscopically in all timesteps in the whole computational domain. The inefficiency caused by the diffusion of species present in large numbers in the subvolumes is then reduced considerably. The macroscopic part is coupled to the mesoscopic simulation of the trajectories by operator splitting.

Starting with the framework in [19], we develop in this paper an adaptive, multilevel algorithm that automatically chooses between the mesoscopic NSM, the explicit tau-leap method, and a macroscopic treatment for the diffusion. This is done by using an operator splitting scheme [51] so that we in each timestep can evolve the RDME using a different method in different regions in space. Provided that the copy number is high enough, the tau-leap method is more efficient than SSA and the cost of integration of the macroscopic diffusion equation is negligible compared to the two stochastic methods. The selection of the method to update the degrees of freedom (dofs) in every timestep is based on error estimates for the expected values when diffusion is advanced with tau-leaping [45] or macroscopically and on the risk of a dof becoming negative in a timestep. The same technique is applicable to both structured Cartesian meshes and unstructured meshes. The simulation time can decrease significantly with this approach with full control of the local errors in the approximations. Two different methods are mixed in [46] where the reactions are treated stochastically and the diffusion by a deterministic approximation.

The contents of the paper are as follows. The modeling background and the RDME are presented in Section 2. The adaptive method is proposed and analyzed in Section 3. The space operator is split according to Strang [51] in Section 3.1 and then the space and time adaptive algorithm is described in Sections 3.2, 3.3 and 3.4. The computational work is estimated in Section 3.5. The method is first applied to diffusion of one species in two dimensions (2D) in Section 4.1 to illustrate the behavior of the algorithm. Then in Section 4.2 a more realistic model with a domain modeling a yeast cell in three dimensions (3D) is simulated on an unstructured mesh in three scenarios using an extension of the URDME software [14] resulting in considerable savings in computing time in some scenarios.

2. Reaction–diffusion master equation

Let the computational domain Ω in space be covered by non-overlapping computational cells or subvolumes $\mathcal{C}_j$, $j = 1, \ldots, K$. The chemical system has N active species X_{ij}, $i = 1, \ldots, N$, in the K cells, $j = 1, \ldots, K$. The non-negative integer x_{ij} is the copy number of species i in cell j. The state of the system is the array $\mathbf{x}$ with $N \times K$ components x_{ij}. The jth column of $\mathbf{x}$ holds the number of molecules of the species in cell j and is denoted by $\mathbf{x}_j$. The copy numbers of species i in all cells are found in the ith row of $\mathbf{x}$ and is denoted by $\mathbf{x}_{i.}$. The state of the system is changed instantaneously by a chemical reaction or

by a molecule diffusing from one cell to an adjacent cell. The probability of the system to be in state $\mathbf{x}$ at time t is given by the PDF $p(\mathbf{x}, t)$.

A chemical reaction r in $\mathcal{C}_j$ is a transition from one state $\mathbf{x}_j$ before the reaction to the state $\mathbf{x}_j + \nu_r$ after the reaction. The probability per unit time or propensity that reaction r will occur in $\mathcal{C}_j$ is a_r and depends on $\mathbf{x}_j$. Usually, a_r is a low order polynomial or a rational polynomial. The vector ν_r of a reaction is the state-change vector. It consists of small integer numbers and is independent of j. A chemical reaction in cell j can be written

$$\mathbf{x}_j \overset{a_r(\mathbf{x}_j)}{\rightarrow} \mathbf{x}_j + \nu_r. \tag{2.1}$$

An example of a bimolecular reaction in $\mathcal{C}_k$ is

$$X_{ik} + X_{jk} \overset{a_1(\mathbf{x}_k)}{\rightarrow} X_{lk}. \tag{2.2}$$

In this case, $a_1(\mathbf{x}_{\cdot k}) = c_{1k} x_{ik} x_{jk}$ according to the law of mass action and the state-change vector is $\nu_{1i} = \nu_{1j} = -1,\ \nu_{1l} = 1$. Split ν_r into two parts

$$\nu_r = \nu_r^+ + \nu_r^-, \quad \nu_{ri}^+ = \max(\nu_{ri}, 0), \quad \nu_{ri}^- = \min(\nu_{ri}, 0),$$

and let $\mathbf{y} \geqslant 0$ denote that $y_i \geqslant 0$ for every i. The master equation for the PDF p in a system without diffusion and R reactions is

$$\frac{\partial p(\mathbf{x}, t)}{\partial t} = \mathcal{M} p(\mathbf{x}, t) \equiv \sum_{j=1}^{K} \sum_{\substack{r=1 \\ \mathbf{x}_j - \nu_r^+ \geqslant 0}}^{R} a_r(\mathbf{x}_j - \nu_r) p(\mathbf{x}_1, \ldots, \mathbf{x}_j - \nu_r, \ldots, \mathbf{x}_{\cdot K}, t) - \sum_{j=1}^{K} \sum_{\substack{r=1 \\ \mathbf{x}_j + \nu_r^- \geqslant 0}}^{R} a_r(\mathbf{x}_j) p(\mathbf{x}, t). \tag{2.3}$$

With one cell in (2.3), $K = 1$, we have the chemical master equation (CME) for a well stirred system, see [23, Chapter 7,33, Chapter V].

Diffusion for a species is modeled as a special kind of reaction with first order kinetics

$$X_{ik} \overset{b_{kj}}{\rightarrow} X_{ij}. \tag{2.4}$$

One molecule of species i in $\mathcal{C}_k$ moves to adjacent $\mathcal{C}_j$ with propensity b_{kj} and state-change vector η_{kj} given by

$$b_{kj} = q_{kj} x_{ik}, \quad k \neq j, \quad \eta_{kj,j} = 1, \quad \eta_{kj,k} = -1, \quad \eta_{kj,i} = 0, \quad i \neq j, k, \quad b_{jj} = 0, \quad q_{jj} = 0. \tag{2.5}$$

The constant q_{kj} depends on the intensity of the diffusion γ and the geometry and the size of the cells $\mathcal{C}_k$ and $\mathcal{C}_j$. In order to simplify the notation we assume here that the diffusion constant γ is the same for all species, but this will not be the case later in the numerical experiments.

In a system without chemical reactions and only diffusion, the master equation can be written in the same manner as the CME in (2.3), see [23, Chapter 8,33, Chapter XIV],

$$\frac{\partial p(\mathbf{x}, t)}{\partial t} = \mathcal{D} p(\mathbf{x}, t) \equiv \sum_{i=1}^{N} \sum_{k=1}^{K} \sum_{j=1}^{K} b_{kj}(\mathbf{x}_{i\cdot} - \eta_{kj}) p(\mathbf{x}_{1\cdot}, \ldots, \mathbf{x}_{i\cdot} - \eta_{kj}, \ldots, \mathbf{x}_{N\cdot}, t) - b_{kj}(\mathbf{x}_{i\cdot}) p(\mathbf{x}, t). \tag{2.6}$$

Summation over the cells is restricted by constraints on $\mathbf{x}$ in the same way as in (2.3). Diffusion between $\mathcal{C}_k$ and $\mathcal{C}_j$ is possible only when they have a point (1D), an edge (2D) or a facet (3D) in common. Hence, most q_{kj} and terms in (2.6) are zero.

The RDME for a chemical system with both reactions and diffusion is derived from (2.3) and (2.6) by adding them together

$$\frac{\partial p(\mathbf{x}, t)}{\partial t} = \mathcal{M} p(\mathbf{x}, t) + \mathcal{D} p(\mathbf{x}, t). \tag{2.7}$$

For the mesocopic model to be valid, there is a lower bound on the size h of the cells due to the reaction radius ρ_R of the molecules. A discussion of these matters is found in [6,18,20,29,47].

The biochemical models in this paper are assumed to be such that the state space is finite, i.e. there is a $x_{\max} > 0$ such that if $x_{ij} \geqslant x_{\max},\ i = 1, \ldots, N,\ j = 1, \ldots, K$, then $p(\mathbf{x}, t) = 0$. This restriction can be motivated for physical and biological reasons.

Let $|\mathcal{C}_j|$ be the length (1D), area (2D), or volume (3D) of cell j. The concentration of species i in cell j is $|\mathcal{C}_j|^{-1} x_{ij}$. The expected value of the concentration $\phi_{ij}(t)$ is defined by

$$\phi_{ij} = \bar{x}_{ij}(t)/|\mathcal{C}_j| = \sum_{\mathbf{x} \geqslant 0} |\mathcal{C}_j|^{-1} x_{ij}(t) p(\mathbf{x}, t). \tag{2.8}$$

The macroscopic equation satisfied by ϕ_{ij} in a system without diffusion is the RRE.

The corresponding equation for a reactive and diffusive system derived in [19] from (2.3), (2.6) and (2.7) is the reaction–diffusion equation

$$\frac{d\phi_{i\cdot}^T}{dt} = \omega_i(\phi) + \gamma D \phi_{i\cdot}^T \tag{2.9}$$

for each i. Without diffusion, $\gamma = 0$, we have the RRE. The off-diagonal elements of the diffusion matrix are the same for each species $\gamma D_{jk} = q_{kj}|\mathcal{C}_k|/|\mathcal{C}_j| \geqslant 0$, $j \neq k$, and the diagonal elements are $\gamma D_{jj} = -\sum_{k\neq j} q_{jk} < 0$. The coefficients q_{kj} are inferred from a discretization of the Laplace operator Δ with Neumann boundary conditions. A detailed discussion of the relation between the elements of D and q is found in [19, Sections 2 and 3]. A finite difference approximation is the simplest choice on a Cartesian mesh and a finite element (FE) approximation is the preferred method on an unstructured mesh as in [19].

If the mesh is a Cartesian lattice with a constant mesh size h and Δ is approximated by the standard 3-point, 5-point, or 7-point stencil in 1D, 2D, or 3D, respectively, then for all cells $|\mathcal{C}_j| = h^d$ where d is the dimension and $q_{kj} = \gamma/h^2$. With a FE discretization using standard piecewise linear test and basis functions and mass lumping on a triangulated, unstructured mesh, $D = A^{-1}S$, where S is symmetric and negative semi-definite and A is a diagonal matrix with the diagonal elements $A_{jj} = |\mathcal{C}_j|$ [19]. Let Q be defined by $Q_{jk} = q_{kj}$ for $k \neq j$ and $Q_{jj} = -\sum_k q_{jk}$. Then

$$Q = \gamma ADA^{-1} = \gamma SA^{-1} = \gamma D^T, \tag{2.10}$$

and $\sum_j Q_{jk} = 0$.

The vertices or nodes (or subvolume centers) are connected by the edges in a graph in Fig. 2.1. The diffusion takes place along the edges in both the unstructured and the structured mesh.

Let the number of particles in a subvolume be scaled by Υ. If the system is well stirred, then $K = 1$ and an interpretation is that the volume of the system grows or that the copy number in a fixed volume grows as Υ grows. It is proved in [35] that under certain usually satisfied assumptions on the propensities a_r, the random vector $\mathbf{X}_\Upsilon$ representing the state of the well stirred system converges to the solution of the RRE in probability in an interval $[0, t]$ as Υ increases i.e.

$$\lim_{\Upsilon\to\infty} P\left(\sup_{s\leqslant t} \|\Upsilon^{-1}\mathbf{X}_\Upsilon(s) - \boldsymbol{\phi}_{\cdot 1}(s)\| > \delta\right) = 0 \tag{2.11}$$

for any $\delta > 0$.

The linear propensities of the diffusion satisfy the conditions in [35]. The state vector of a diffusive system is always finite (but large) since there is a lower bound on the mesh size h. Let the copy number x_{ij} be scaled by an increasing number Υ and the cell size $|\mathcal{C}_j|$ in every cell. The random state array $\mathbf{X}_\Upsilon$ in (2.11) now has NK components. Each element $X_{\Upsilon ij}$ of $\mathbf{X}_\Upsilon$ represents the copy number per volume of species i in subvolume j. It follows from [35] that $\Upsilon^{-1}\mathbf{X}_\Upsilon$ with PDF solving the RDME (2.7) converges to the concentrations $\boldsymbol{\phi}$ solving (2.9) as $\Upsilon \to \infty$ in the same way as in (2.11). The conclusion is that a macroscopic, deterministic treatment of a species in a cell is accurate when the copy number is large. The convergence rate is as expected from [36] of $\mathcal{O}(\Upsilon^{-1/2})$ in numerical experiments in [19] and in Section 4.1.

3. Method of solution

The algorithm for realization of one trajectory of the diffusive chemical system governed by the RDME (2.7) is a hybrid method blending SSA, tau-leaping and a macroscopic approximation with an automatic choice between the different levels of modeling for systems with many more diffusion events than reaction events. The advantage is reduced computing time and control of the errors introduced in the simulation of the diffusion by the tau-leap method and the approximation at the macro level.

The basic idea is to deal with the stiffness caused by frequent diffusion events by applying approximate methods (explicit tau-leaping or a macroscopic scheme) to update those species for which diffusion is described sufficiently accurately by the respective method. Since the distribution of the species varies in space and time, the selection of the method must be adaptive. First, a time interval Δt is determined in which the relative error in the state variable caused by tau-leaping or the macroscopic Euler scheme is below a specified tolerance. Then all degrees of freedom (dofs) are partitioned into two sets, one set for variables suitable for tau-leaping and one set suitable for the macro level approximation. Looking at the first set, we remove the dofs that suffer a large risk of becoming negative by tau-leaping in that interval. Those dofs are placed in a third SSA set which will be updated by NSM in the time interval and we arrive at the final partitioning with three sets

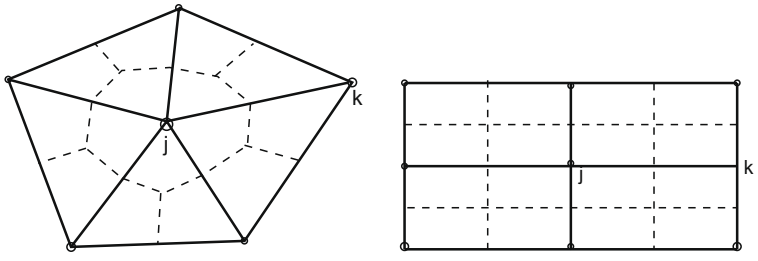

Fig. 2.1. Vertices (o), edges (solid lines), and the subvolume boundaries (dashed lines) in an unstructured mesh (left) and a Cartesian structured mesh (right).

$\mathcal{V}_\tau, \mathcal{V}_m, \mathcal{V}_{SSA}$. On the boundaries between sets the most accurate method has precedence and this allows us to classify every edge in the mesh to tell us which method to use to simulate the diffusion. The reactions are all treated by SSA.

The diffusion operator in (2.7) is split into three parts corresponding to the SSA, tau-leap, and macroscopic edges. Then each part is advanced in time consecutively in a Strang splitting scheme introducing an error proportional to Δt^3 locally and to Δt^2 globally. Since the tau-leap and the Euler methods are first order accurate, the global error of the complete scheme is of $\mathcal{O}(\Delta t)$.

In the following sections, we consider the different components of the method in more detail. In Section 3.1, the ideas underlying operator splitting are reviewed. In Section 3.2, the criteria for the partitioning are given and the ensuing section deals with the time adaptivity. A detailed account of the full algorithm is found in Section 3.4. The computational work is estimated in Section 3.5.

3.1. Operator splitting

Suppose that the right hand side of the RDME (2.7) consists of two sums $\mathcal{A}p$ and $\mathcal{B}p$ such that

$$\frac{\partial p(\mathbf{x},t)}{\partial t} = \mathcal{A}p(\mathbf{x},t) + \mathcal{B}p(\mathbf{x},t), \tag{3.1}$$

and that it is much simpler to solve $\partial_t p = \mathcal{A}p$ and $\partial_t p = \mathcal{B}p$ separate from each other than the full equation in (3.1). Then, by first solving $\partial_t p = \mathcal{B}p$ numerically in an interval $[t, t+\Delta t]$ and then $\partial_t p = \mathcal{A}p$ in $[t, t+\Delta t]$ with the output from the first equation as initial data for the second equation, we have an approximation of $p(\mathbf{x}, t+\Delta t)$ of $\mathcal{O}(\Delta t)$ globally [40]. The error is proportional to the commutator of $\mathcal{A}$ and $\mathcal{B}$. Since the operators are independent of time, the solution in the first step can be written

$$p_1(\mathbf{x}, t+\Delta t) = e^{\Delta t \mathcal{B}} p(\mathbf{x}, t)$$

and in the second step

$$p_2(\mathbf{x}, t+\Delta t) = e^{\Delta t \mathcal{A}} p_1(\mathbf{x}, t+\Delta t)$$

and consequently the solution p_O obtained by operator splitting is

$$p_O(\mathbf{x}, t+\Delta t) = p_2(\mathbf{x}, t+\Delta t) = e^{\Delta t \mathcal{A}} e^{\Delta t \mathcal{B}} p(\mathbf{x}, t) \tag{3.2}$$

with the difference $p_O(\mathbf{x}, t+\Delta t) - p(\mathbf{x}, t+\Delta t)$ of $\mathcal{O}(\Delta t)$. A realization of the stochastic process $\mathbf{X}_O$ in $[t, t+\Delta t]$ with the PDF p_O in (3.2) is achieved by a trajectory $\mathbf{X}_1$ generated with the reactions involved in $\mathcal{B}$ in $[t, t+\Delta t/2]$ and followed by $\mathbf{X}_2$ simulated by the reactions defining $\mathcal{A}$ in the same interval.

In Strang splitting [51] the time accuracy is improved. The solution of (3.1) is approximated in the interval by first integrating $\partial_t p = \mathcal{A}p$ between t and $t+\Delta t/2$, then $\partial_t p = \mathcal{B}p$ in $[t, t+\Delta t]$, and finally $\partial_t p = \mathcal{A}p$ in $[t+\Delta t/2, t+\Delta t]$ with the preceding output p as input to the next step. The local error in each step is of $\mathcal{O}(\Delta t^3)$ with the bounded operators in (2.7) making the approximation second order accurate globally. The trajectory $\mathbf{X}_S$ is computed by using the reactions in $\mathcal{A}$, then $\mathcal{B}$, and finally $\mathcal{A}$ again.

Now consider the RDME (2.7) and split the right hand side into three parts

$$\frac{\partial p(\mathbf{x},t)}{\partial t} = \mathcal{A}p(\mathbf{x},t) + \mathcal{B}p(\mathbf{x},t) + \mathcal{C}p(\mathbf{x},t). \tag{3.3}$$

With the Strang technique applied recursively to (3.3), the approximation is

$$p(\mathbf{x}, t+\Delta t) = e^{\Delta t(\mathcal{A}+\mathcal{B}+\mathcal{C})} p(\mathbf{x},t) = e^{\frac{1}{2}\Delta t \mathcal{A}} e^{\frac{1}{2}\Delta t \mathcal{B}} e^{\Delta t \mathcal{C}} e^{\frac{1}{2}\Delta t \mathcal{B}} e^{\frac{1}{2}\Delta t \mathcal{A}} p(\mathbf{x},t) + \mathcal{O}(\Delta t^3). \tag{3.4}$$

Let $\mathbf{X}(t)$ be a realization with SSA of the process with the PDF in (3.3) and let $\mathbf{X}_S(t)$ be the realization with the PDF given by the approximation (3.4) in $[t, t+\Delta t]$. The reaction and diffusion propensities are time independent and the split process can be considered as one where each one of $\mathcal{A}, \mathcal{B}$, and $\mathcal{C}$ is active in one third of the time interval and silent in the remaining two thirds. The stages to generate one trajectory $\mathbf{X}_S$ using SSA are

Algorithm 3.1

1. Advance $\mathbf{X}_{S1}$ $\Delta t/2$ in time with the process defined by $\mathcal{A}$ starting with $\mathbf{X}_S(t)$.
2. Advance $\mathbf{X}_{S2}$ $\Delta t/2$ in time with the process defined by $\mathcal{B}$ starting with $\mathbf{X}_{S1}(t+\Delta t/2)$.
3. Advance $\mathbf{X}_{S3}$ Δt in time with the process defined by $\mathcal{C}$ starting with $\mathbf{X}_{S2}(t+\Delta t/2)$.
4. Advance $\mathbf{X}_{S4}$ $\Delta t/2$ in time with the process defined by $\mathcal{B}$ starting with $\mathbf{X}_{S3}(t+\Delta t)$.
5. Advance $\mathbf{X}_{S5}$ $\Delta t/2$ in time with the process defined by $\mathcal{A}$ starting with $\mathbf{X}_{S4}(t+\Delta t/2)$.
6. Let $\mathbf{X}_S(t+\Delta t) = \mathbf{X}_{S5}(t+\Delta t/2)$.

The PDF corresponding to $\mathbf{X}_S(t)$ is $p_S(\mathbf{x}, t)$ satisfying

$$p_S(\mathbf{x}, t+\Delta t) = e^{\frac{1}{2}\Delta t \mathcal{A}} e^{\frac{1}{2}\Delta t \mathcal{B}} e^{\Delta t \mathcal{C}} e^{\frac{1}{2}\Delta t \mathcal{B}} e^{\frac{1}{2}\Delta t \mathcal{A}} p_S(\mathbf{x}, t). \tag{3.5}$$

Compared to a direct splitting obtained by applying the procedure leading to (3.2) twice

$$p(\mathbf{x}, t+\Delta t) = e^{\Delta t\mathcal{A}} e^{\Delta t\mathcal{B}} e^{\Delta t\mathcal{C}} p(\mathbf{x}, t) + \mathcal{O}(\Delta t^2), \tag{3.6}$$

higher accuracy is achieved with (3.4) requiring little extra work.

The following weak convergence result, valid e.g. for the moments, is easily proved:

Proposition 3.2. *Let g be a bounded function, let $\mathbf{X}(t)$ be the realization of the process defined by the stable integration of $p(\mathbf{x}, t)$ in (3.3), and let $\mathbf{X}_S(t)$ be defined by $p_S(\mathbf{x}, t)$ in (3.5) with $\mathbf{X}(0) = \mathbf{X}_S(0)$ and $p(\mathbf{x}, 0) = p_S(\mathbf{x}, 0)$. Then*

$$E[g(\mathbf{X}_S(t^n)) - g(\mathbf{X}(t^n))] = \mathcal{O}(\Delta t^2), \quad t^n = n\Delta t.$$

Proof. The difference between the PDFs is $p(\mathbf{x}, t^n) - p_S(\mathbf{x}, t^n) = \mathcal{O}(\Delta t^2)$ by standard theory for numerical solution of ordinary differential equations [27]. Hence, by definition

$$E[g(\mathbf{X}(t^n)) - g(\mathbf{X}_S(t^n))] = \sum_{\mathbf{x}} g(\mathbf{x})(p(\mathbf{x}, t^n) - p_S(\mathbf{x}, t^n)) = \mathcal{O}(\Delta t^2),$$

since the state space is finite by the assumption in Section 2. □

In our adaptive algorithm, the master operator in (2.7) is split into three parts

$$\mathcal{A} = \mathcal{D}_m,\ \mathcal{B} = \mathcal{D}_\tau,\ \mathcal{C} = \mathcal{M} + \mathcal{D}_{SSA}, \tag{3.7}$$

where $\mathcal{D} = \mathcal{D}_m + \mathcal{D}_\tau + \mathcal{D}_{SSA}$. In a trajectory $\mathbf{X}_S$, the diffusion in steps 1 and 5 of Algorithm 3.1 is approximated macroscopically (corresponding to $\mathcal{D}_m$). In Section 4.1 the diffusion in steps 2 and 4 is approximated by the tau-leap method ($\mathcal{D}_\tau$), and the diffusion and the chemical reactions in step 3 are simulated by SSA. In Section 4.2, we use the splitting $\mathcal{A} = \mathcal{D}_m, \mathcal{B} = \mathcal{M} + \mathcal{D}_{SSA}, \mathcal{C} = \mathcal{D}_\tau$ and NSM is used for steps 2 and 4 for higher efficiency. The error in the moments of $\mathbf{X}_S$ due to the time splitting is of $\mathcal{O}(\Delta t^2)$ according to the proposition but additional errors are introduced by the macro level and tau-leap diffusion. We will show below that those errors are of $\mathcal{O}(\Delta t)$ and consequently we can ignore the splitting error for small Δt.

3.2. Space adaptivity

The reactions and the diffusion of the species are treated in three different ways:

1. SSA for the reactions and diffusion for a small copy number of the species in each subvolume
2. tau-leaping for intermediate copy numbers
3. deterministic, macroscopic diffusion for large copy numbers.

The method of diffusing the molecules is different in different parts of the domain and for different species and varies in time. The method is determined by estimates of the errors in tau-leaping and macroscopic diffusion.

3.2.1. SSA

The direct method by Gillespie [25] is applied to all reactions and to diffusion between certain vertices. Let $\mathcal{E}^i_{SSA}$ be the set of pairs of vertices (or subvolumes) (j,k) between which diffusion of species i is treated with SSA and let $t_0 = t^n$ and $\ell = 0$. Sum all reaction propensities a_r and the propensities $b_{jk},\ (j,k) \in \mathcal{E}^i_{SSA}$, at t_ℓ to obtain

$$\beta_\ell = \sum_{j=1}^{K}\sum_{r=1}^{R} a_r(\mathbf{x}^\ell_j) + \sum_{i=1}^{N}\sum_{(j,k)\in\mathcal{E}^i_{SSA}} b_{jk}(\mathbf{x}^\ell_{i.}). \tag{3.8}$$

The next event will occur after time δt_ℓ which is exponentially distributed with parameter β_ℓ. Update the time $t_{\ell+1} = t_\ell + \delta t_\ell$. If $t_{\ell+1} \leqslant t^{n+1}$, then choose either a reaction event r with probability $a_r(\mathbf{x}_j)/\beta_\ell$ or a diffusion event with probability $b_{jk}(\mathbf{x}_{i.})/\beta_\ell$, update the state vector at $t_{\ell+1}, \ell := \ell + 1$, and continue with a new SSA step. If $t_{\ell+1} > t^{n+1} = t^n + \Delta t^n$, then interrupt the SSA iteration and let the present state vector be the state vector at t^{n+1}. SSA is used for the exact stochastic step in Section 4.1. The NSM algorithm in [18] is an efficient implementation of SSA for reaction–diffusion processes and is used as implemented in URDME [14] in the experiments in Section 4.2.

3.2.2. τ-leaping

The diffusion of species i from subvolume k to subvolume j in the interval $[t^n, t^{n+1}]$ of length Δt^n is approximated by the tau-leap method [26] in the following way. Let $\mathcal{E}^i_\tau$ be the set of pairs of vertices (k,j) with tau-leap approximation of the diffusion between them for species i. The number of molecules u^n_{kj} of species i moving from k to j during $[t^n, t^{n+1}]$ with $(k,j) \in \mathcal{E}^i_\tau$ is Poisson distributed with parameter $q_{kj}x^n_{ik}\Delta t^n$ and a probability mass function (PMF) $\mathcal{P}(q_{kj}x^n_{ik}\Delta t^n)$. The total number of molecules moving from adjacent cells to j is $v^n_{+j} = \sum_k u^n_{kj}$ and the number of molecules moving away from j is $v^n_{-j} = \sum_k u^n_{jk}$ in the interval. Thus, the total change in cell j is

$$\Delta v_j^n = v_{+j}^n - v_{-j}^n. \tag{3.9}$$

Since each u_{kj}^n is $\mathcal{P}(q_{kj}x_{ik}^n\Delta t^n)$, the sums v_{+j}^n and v_{-j}^n are Poisson distributed with parameters λ_j^+ and λ_j^-, respectively, where

$$\begin{aligned}\lambda_j^+ &= \sum_{k,k\neq j} q_{kj}x_{ik}^n\Delta t^n = \Delta t^n\sum_{k,k\neq j} Q_{jk}x_{ik}^n,\\ \lambda_j^- &= \sum_{k,k\neq j} q_{jk}x_{ij}^n\Delta t^n = -\Delta t^n Q_{jj}x_{ij}^n,\end{aligned} \tag{3.10}$$

with the definition of Q from (2.10). The summation is over k such that $(k,j)\in\mathcal{E}_\tau^i$. The difference Δv_j^n between two Poisson distributed random numbers is Skellam distributed [48]. The probability for $\Delta v_j^n = \alpha$ is

$$P\left(\Delta v_j^n = \alpha\right) = p_{\mathcal{S}\alpha} = \exp\left(-\left(\lambda_j^+ + \lambda_j^-\right)\right)\left(\frac{\lambda_j^+}{\lambda_j^-}\right)^{\alpha/2} I_\alpha\left(2\sqrt{\lambda_j^+\lambda_j^-}\right) \tag{3.11}$$

using the modified Bessel function I_α. The Skellam distribution of Δv_j^n is well approximated by the normal distribution $\mathcal{N}\left(\mu_j,\sigma_j^2\right)$ in particular for large λ_j^+ and λ_j^-, see [31,50], with

$$\begin{aligned}\mu_j &= \lambda_j^+ - \lambda_j^- = \Delta t^n\sum_k Q_{jk}x_{ik}^n,\\ \sigma_j^2 &= \lambda_j^+ + \lambda_j^- = \Delta t^n\left(\sum_{k,k\neq j} Q_{jk}x_{ik}^n - Q_{jj}x_{ij}^n\right) = \Delta t^n\sum_k |Q|_{jk}x_{ik}^n,\end{aligned} \tag{3.12}$$

where $|Q|_{jk} = Q_{jk}$ when $j\neq k$ and $|Q|_{jj} = -Q_{jj}$ and consequently $|Q|_{jk}\geqslant 0$. The variance σ_j^2 in (3.12) is positive if $x_{ik}^n\neq 0$ at least for some k and it is a local weighted summation of x_{ik}^n with the largest weight on x_{ij}^n.

The new number of molecules in cell j at t^{n+1} is

$$x_{ij}^{n+1} = x_{ij}^n + \Delta v_j^n \tag{3.13}$$

after updating by Δv_j^n in (3.9). There is a risk that $x_{ij}^{n+1} < 0$ which is unacceptable for physical reasons. A number of remedies have been suggested to avoid this predicament in the tau-leap method [2,3,9,13,44,53]. Approximation of Δv_j^n by a binomial distribution also guarantees non-negativity in [13,53]. The numbers u_{jk}^n are reduced successively in [45] to avoid negative states and in [9] the reactions are simulated by SSA if the propensity times the timestep is larger than a given parameter. Our solution to this problem is as follows for the diffusion. The probability of obtaining a negative number of molecules is from (3.11)

$$P\left(x_{ij}^{n+1}\leqslant -1\right) = P\left(\Delta v_j^n\leqslant -1 - x_{ij}^n\right) = \sum_{\alpha=-\infty}^{-1-x_{ij}^n} p_{\mathcal{S}\alpha}. \tag{3.14}$$

There is no known simple closed form of the cumulative distribution function of the Skellam distribution and the probability P in (3.14) is expensive to evaluate as a sum of the PDFs. However, the approximation with the normal distribution with mean μ_j and variation σ_j^2 from (3.12) yields

$$P\left(x_{ij}^{n+1}\leqslant -1\right) \approx \frac{1}{\sqrt{2\pi}\sigma_j}\int_{-\infty}^{-1-x_{ij}^n} \exp\left(-(x-\mu_j)^2/\left(2\sigma_j^2\right)\right)dx. \tag{3.15}$$

By taking a smaller timestep, the risk in (3.14) decreases but with the same Δt^n for all diffusion events the generation of the trajectory becomes less efficient.

In our algorithm, we will accept the tau-leaping in a vertex if the estimated probability P for failure in (3.14) is sufficiently small $P<\epsilon_1$ for some $\epsilon_1>0$. Otherwise, diffusion in the subvolume is simulated by SSA in the time interval. If there still is a negative x_{ij}^{n+1} after these precautions, then the non-negativity is enforced in the same way as in [45]. The error due to the finite timestep is estimated in Section 3.3.

3.2.3. Macroscopic diffusion

It is shown in [19] that the mean values $\bar{\mathbf{x}}_{i\cdot}$ of the copy numbers of species i in the subvolumes exactly satisfy the equation

$$\frac{d\bar{\mathbf{x}}_{i\cdot}^T}{dt} = Q\bar{\mathbf{x}}_{i\cdot}^T \tag{3.16}$$

in a system with only diffusion. By the properties of Q in (2.10), $\bar{\mathbf{x}}_{i\cdot}^T$ is bounded for all time. Furthermore, the variance is bounded by $C\|\bar{\mathbf{x}}_{i\cdot}\|$ and the random variation about the mean value is $\sim\sqrt{\|\bar{\mathbf{x}}_{i\cdot}\|}$ [19]. Thus, the quotient between the standard deviations and the mean values is $\sim 1/\sqrt{\|\bar{\mathbf{x}}_{i\cdot}\|}$ and for large copy numbers, the effect of diffusion is well approximated by (3.16). We will derive the same approximation from the tau-leaping in the previous section but now including a local error estimate.

When the parameter $\lambda_{kj}^n = q_{kj}x_{ik}^n\Delta t^n$ is large in the Poisson distribution for the number of molecules u_{kj}^n diffusing from k to j, the distribution of u_{kj}^n is close to the normal distribution $\mathcal{N}\left(\lambda_{kj}^n, \lambda_{kj}^n\right)$ (see e.g. [26,37]). Then we can write

$$u_{kj}^n = \lambda_{kj}^n + \xi_{kj}^n,$$

where ξ_{kj}^n is $\mathcal{N}\left(0, \lambda_{kj}^n\right)$. Therefore, the total outflow and inflow of molecules to cell j is

$$\Delta x_{ij}^n = \sum_{k,k\neq j} u_{kj}^n - \sum_{k,k\neq j} u_{jk}^n = \Delta t^n\left(\sum_{k,k\neq j} q_{kj}x_{ik}^n - \sum_{k,k\neq j} q_{jk}x_{ij}^n\right) + \Xi_j^n = \Delta t^n\left(Q\left(\mathbf{x}_{i\cdot}^n\right)^T\right)_j + \Xi_j^n,\ \Xi_j^n \equiv \sum_{k,k\neq j}\xi_{kj}^n - \sum_{k,k\neq j}\xi_{jk}^n.$$

The mean value of Ξ_j^n is 0 and we infer

$$\bar{x}_{ij}^{n+1} \equiv E\left[x_{ij}^{n+1}\right] = E\left[x_{ij}^n + \Delta x_{ij}^n\right] = \bar{x}_{ij}^n + \Delta t^n E\left[\left(Q\left(\mathbf{x}_{i\cdot}^n\right)^T\right)_j\right] + E\left[\Xi_j^n\right] = \bar{x}_{ij}^n + \Delta t^n\left(Q\left(\bar{\mathbf{x}}_{i\cdot}^n\right)^T\right)_j. \tag{3.17}$$

This is the time discretization of (3.16) with the Euler forward method [27]. The variance of Ξ_j^n is σ_j^2 in (3.12). Since Δx_{ij}^ℓ, $\ell = 0,\ldots,n-1$, are independent and x_{ij}^0 is given, the variance of x_{ij}^n is

$$\mathrm{Var}\left[x_{ij}^n\right] = \mathrm{Var}\left[\sum_{\ell=0}^{n-1}\Delta x_{ij}^\ell\right] = \sum_{\ell=0}^{n-1}\mathrm{Var}\left[\Delta x_{ij}^\ell\right] = \sum_{\ell=0}^{n-1}\mathrm{Var}\left[\Xi_j^\ell\right] = \sum_{\ell=0}^{n-1}\Delta t^\ell\left(|Q|\left(\mathbf{x}_{i\cdot}^\ell\right)^T\right)_j. \tag{3.18}$$

The random error in the macroscopic approximation of the diffusion in (3.16) and (3.17) is estimated in the next proposition.

Proposition 3.3. *Assume that*

$$\sqrt{\left(|Q|\left(\mathbf{x}_{i\cdot}^\ell\right)^T\right)_j}\Big/\bar{x}_{ij}^\ell \leqslant \epsilon_2, \quad \ell = 0,\ldots,n-1, \tag{3.19}$$

for some small $\epsilon_2 > 0$. Then the deviation δ_{ij}^n of x_{ij}^n from $\bar{x}_{ij}^n$ satisfies with probability 0.95

$$\frac{\delta_{ij}^n}{\bar{x}_{ij}^n} \in \left[-1.96\frac{\epsilon_2\chi_{ij}^n}{\bar{x}_{ij}^n}, 1.96\frac{\epsilon_2\chi_{ij}^n}{\bar{x}_{ij}^n}\right],$$

where

$$\chi_{ij}^n = \sqrt{\textstyle\sum_{\ell=0}^{n-1}\Delta t^\ell(\bar{x}_{ij}^\ell)^2}.$$

Proof. Replace $\left(|Q|\left(\mathbf{x}_{i\cdot}^n\right)^T\right)_j$ in (3.18) by the upper bound in the assumption and the proposition follows from the properties of the normal distribution. □

An approximation of χ_{ij}^n is

$$\chi_{ij}^n \approx \sqrt{\int_0^{t^n}(\bar{x}_{ij}(t))^2\,dt} = \sqrt{t^n}\sqrt{\overline{(\bar{x}_{ij})^2}}, \quad \overline{(\bar{x}_{ij})^2} \equiv (t^n)^{-1}\int_0^{t^n}(\bar{x}_{ij}(t))^2\,dt.$$

Consequently, $\left(\epsilon_2\chi_{ij}^n\right)/\bar{x}_{ij}^n \sim \epsilon_2\sqrt{t^n}$. The quotient in (3.19) is small when $\bar{x}_{ij}^n$ is large since the numerator is $\sim \sqrt{\bar{x}_{ij}^n}$. As an example consider a uniform Cartesian mesh in 2D with step size h with the standard 5-point stencil approximating the Laplacian in Section 2. Then

$$\left(|Q|\left(\mathbf{x}_{i\cdot}^n\right)^T\right)_j = h^{-2}\gamma\left(x_{i,j-1}^n + x_{i,j+1}^n + x_{i-1,j}^n + x_{i+1,j}^n + 4x_{ij}^n\right),$$

and with a smooth behavior of x_{ij}^n in the neighborhood of j, the condition (3.19) is $2\sqrt{2\gamma}/\left(h\sqrt{x_{ij}^n}\right) \lesssim \epsilon_2$.

A simple macroscopic diffusion is sufficient for good accuracy if the conditions in Proposition 3.3 are satisfied. The pairs of vertices (k,j) where the deterministic diffusion is satisfactory for species i are collected in the set $\mathcal{E}_m^i$. The great advantage is that the computational cost of diffusion at the macro level is negligible compared to updating the copy numbers with the tau-leap method or SSA.

3.3. Timestep selection

The timesteps are selected such that a tolerance ϵ_3 is satisfied by the relative local discretization error in every step. The expected error in the tau-leap stage of the Strang splitting in Algorithm 3.1 is derived from the expression for general reactions in [45]. Another way of determining the timesteps for tau-leaping is proposed in [11]. The local error in the macroscopic approximation in the $\mathcal{D}_m$ stage in Algorithm 3.1 depends on the method to integrate (3.16) and follows from standard error estimates in [27]. Advancing the chemical reactions and some diffusion events in time with SSA introduces no additional errors.

The local error of tau-leaping for a system with a master Eq. (2.3) and R reactions is derived in [45]. Let $\mathbf{X}_\tau(t)$ denote the tau-leap trajectory and compare it with the SSA trajectory $\mathbf{X}(t)$. If $\mathbf{X}_\tau(t) = \mathbf{X}(t) = \mathbf{x}$ then

$$E[\mathbf{X}_\tau(t+\Delta t) - \mathbf{X}(t+\Delta t)] = -0.5\Delta t^2 \sum_{r_1=1}^{R}\sum_{r_2=1}^{R} \boldsymbol{\nu}_{r_1} a_{r_2}(\mathbf{x})(a_{r_1}(\mathbf{x}+\boldsymbol{\nu}_{r_2}) - a_{r_1}(\mathbf{x})) + \mathcal{O}(\Delta t^3). \tag{3.20}$$

With

$$\begin{aligned} &r_1 \to (jk), \quad r_2 \to (lm), \quad a_{r_1}(\mathbf{x}) = q_{jk}x_{ij}, \quad a_{r_2}(\mathbf{x}) = q_{lm}x_{il}, \\ &\boldsymbol{\nu}_{r_1} = \boldsymbol{\eta}_{jk}, \quad \boldsymbol{\nu}_{r_2} = \boldsymbol{\eta}_{lm}, \quad \eta_{lm,m} = 1, \quad \eta_{lm,l} = -1, \quad \eta_{lm,i} = 0, \quad i \neq l \text{ or } m, \end{aligned}$$

from the definition of mesoscopic diffusion in (2.5), the sum on the right hand side in (3.20) is

$$\begin{aligned} \sum_{jk}\sum_{lm} \boldsymbol{\eta}_{jk} q_{lm} x_{il} q_{jk} \eta_{lm,j} &= \sum_{jk} \boldsymbol{\eta}_{jk} q_{jk} \sum_{lm} q_{lm} x_{il} \eta_{lm,j} = \sum_{jk} \boldsymbol{\eta}_{jk} q_{jk} \left(\sum_{l} q_{lj} x_{il} \eta_{lj,j} - \sum_{m} q_{jm} x_{ij} \eta_{jm,j} \right) \\ &= \sum_{jk} \boldsymbol{\eta}_{jk} q_{jk} \left(\sum_{l} q_{lj} x_{il} - x_{ij} \sum_{m} q_{jm} \right). \end{aligned} \tag{3.21}$$

Introduce the diffusion of species i in cell j

$$d_{ij} = \sum_{l} q_{lj} x_{il} - x_{ij} \sum_{m} q_{jm} = \sum_{l \neq j} Q_{jl} x_{il} + Q_{jj} x_{ij} = \left(Q\mathbf{x}_{i\cdot}^T\right)_j.$$

Then the expected local error for diffusion tau-leaping in cell l is obtained from (3.20), (3.21), and the above definition

$$\begin{aligned} E[\mathbf{X}_\tau(t+\Delta t) - \mathbf{X}(t+\Delta t)]_{il} &= -0.5\Delta t^2 \sum_{jk} \eta_{jk,l} q_{jk} d_{ij} + \mathcal{O}(\Delta t^3) = -0.5\Delta t^2 (Q\mathbf{d}_{i\cdot})_l + \mathcal{O}(\Delta t^3) \\ &= -0.5\Delta t^2 \left(Q^2 \mathbf{x}_{i\cdot}^T\right)_l + \mathcal{O}(\Delta t^3). \end{aligned} \tag{3.22}$$

The conclusion from our derivations is

Proposition 3.4. *The expected value of the difference after one timestep Δt between the tau-leap trajectory $\mathbf{X}_\tau$ and the SSA trajectory $\mathbf{X}$ starting at $\mathbf{X}_\tau(t) = \mathbf{X}(t)$ is for species i*

$$E[\mathbf{X}_\tau(t+\Delta t) - \mathbf{X}(t+\Delta t)]_{i\cdot} = -0.5\Delta t^2 Q^2 \mathbf{x}_{i\cdot}^T + \mathcal{O}(\Delta t^3).$$

Assume that the integration is numerically stable with Δt. Then it is proved for linear propensities in [45] that globally at t^n, the difference will be of $\mathcal{O}(\Delta t)$, one order lower than the error due to the operator splitting in Section 3.1.

The system of differential equations for the mean values (3.16) is solved either by the Euler forward or the Euler backward method. The new $\mathbf{x}$ at t^{n+1} is compared to the analytical solution $\mathbf{x}(t^{n+1})$ with the same initial data $\mathbf{x}^n = \mathbf{x}(t^n)$. In the Euler methods, we have

$$\begin{aligned} &\left(\mathbf{x}_{i\cdot}^{n+1}\right)^T = \left(\mathbf{x}_{i\cdot}^{n}\right)^T + \Delta t Q \left(\mathbf{x}_{i\cdot}^{v}\right)^T, \\ &\left(\mathbf{x}_{i\cdot}^{n+1}\right)^T - \mathbf{x}_{i\cdot}(t^{n+1})^T = s0.5\Delta t^2 Q^2 (\mathbf{x}_{i\cdot}(t^n))^T + \mathcal{O}(\Delta t^3), \end{aligned} \tag{3.23}$$

with $v = n$ and $s = -1$ for the forward method (cf. (3.17)) and $v = n+1$ and $s = 1$ for the backward method. The backward method has no restrictions on Δt for stability for our Q and it is proved in [19] that the non-negativity $\mathbf{x}$ is preserved for any positive timestep. For a sufficiently small timestep, $\mathbf{x}$ remains non-negative also with the forward method. Globally, both Euler methods are first order accurate.

The modulus of the leading term in the local errors is the same for the tau-leap method and the two Euler twins in Proposition 3.4 and (3.23). The timestep Δt^n is chosen here such that the estimated relative local error is less than a prescribed relative tolerance ϵ_3 at every vertex where the diffusion is simulated with tau-leaping or at the macro level. The local relative error in species i at vertex l at t^n (lre_{il}^n) due to the approximations of the diffusion is

$$lre_{il}^n = \left| 0.5\Delta t^2 \left(Q^2 \left(\mathbf{x}_{i\cdot}^n\right)^T\right)_l \Big/ x_{il}^n \right|.$$

Then in order to have $lre_{il} \leqslant \epsilon_3$ everywhere the timestep is

$$\Delta t^n \leqslant \left(\min_i \frac{2\epsilon_3}{\max_l \left| \left(Q^2 \left(\mathbf{x}_{i\cdot}^n\right)^T\right)_l \Big/ x_{il}^n \right|} \right)^{1/2}. \tag{3.24}$$

3.4. Algorithm

A realization of the process governed by (2.7) is initialized by generating Q in (2.10) including all vertices in the mesh. Determine the error tolerances ϵ_2 and ϵ_3 in (3.19) and (3.24).

The timestep Δt^n in the Strang splitting, tau-leaping and macroscopic diffusion is chosen to satisfy (3.24) based on the solution $\mathbf{x}^n$. Then a vertex j is classified for species i as either being a macro vertex, $j \in \mathcal{V}_m^i$, a tau vertex, $j \in \mathcal{V}_\tau^i$, or a SSA vertex, $j \in \mathcal{V}_{SSA}^i$, with $\mathcal{V}_m^i \cup \mathcal{V}_\tau^i \cup \mathcal{V}_{SSA}^i = \mathcal{V}$. A macro vertex j for species i fulfills the inequality in (3.19) at t^n. If the probability to obtain a negative copy number at a vertex in $\mathcal{V} \setminus \mathcal{V}_m^i$ exceeds ϵ_1 in (3.15), i.e. if

$$x_{ij}^n + \mu_j - \kappa(\epsilon_1)\sigma_j < -1, \tag{3.25}$$

then j is a SSA vertex for species i. The Skellam distribution is always approximated by the normal distribution as in (3.15). With $\epsilon_1 = 0.01, \kappa = 2.33$, and with $\epsilon_1 = 0.025, \kappa = 1.96$. The remaining vertices are classified as tau vertices in $\mathcal{V}_\tau^i$.

The diffusion between a pair of vertices (j,k) is performed with the macroscopic approximation, tau-leaping or SSA depending on the classification of j and k. If both j and k belong to $\mathcal{V}_m^i$ then the diffusion is macroscopic in both directions on the edge, $(j,k),(k,j) \in \mathcal{E}_m^i$. The diffusion matrix Q_m is the submatrix of Q where rows and columns corresponding to vertices in $\mathcal{V}_m^i$ are included. The diagonal element in Q_m is adjusted so that $\sum_j Q_{m,jk} = 0$ for all k. The submatrix Q_m has the same properties as the full matrix Q in (2.10) and the condition at the boundary of the subgraph consisting of vertices in $\mathcal{V}_m^i$ is a discretization of a Neumann condition. If $\mathcal{V}_m^i = \mathcal{V}$ and the diffusion is deterministic everywhere, then $Q_m = Q$ and the solution is given by a discretization of the Laplace operator on the whole mesh, see Section 2.

The total number of molecules in the macro vertices is preserved with the Euler methods. Let $\mathbf{e}$ be the vector with all elements equal to 1 and let $\tilde{x}_{ij}$ denote the components of $\mathbf{x}_{i\cdot}$ with $j \in \mathcal{V}_m^i$. Then by (3.23) we have

$$\mathbf{e}^T\left(\tilde{\mathbf{x}}_{i\cdot}^{n+1}\right)^T = \mathbf{e}^T\left(\tilde{\mathbf{x}}_{i\cdot}^{n}\right)^T + \Delta t^n \mathbf{e}^T Q_m \left(\tilde{\mathbf{x}}_{i\cdot}^{\nu}\right)^T = \mathbf{e}^T\left(\tilde{\mathbf{x}}_{i\cdot}^{n}\right)^T,\ \nu = n \text{ or } n+1,$$

and the total number of molecules of species i, $\mathbf{e}^T(\tilde{\mathbf{x}}_{i\cdot}^n)^T$, is constant.

The diffusion between two vertices j and k is updated by tau-leaping, if $j \in \mathcal{V}_\tau^i$, there is an edge between j and k, and $k \in \mathcal{V}_\tau^i \cup \mathcal{V}_m^i$. After summation over all permissible k, the new state at j is then given by (3.9) and (3.13). SSA is used for simulation of the diffusion between the remaining pairs of vertices (j,k) with $j \in \mathcal{V}_{SSA}^i$ and $k \in \mathcal{V}$. Both tau-leaping and SSA keep the total number of molecules constant.

Real numbers are not permitted in the tau-leap algorithm and SSA. Therefore, the copy numbers updated by macroscopic diffusion are rounded to nearest integers. Let $\lfloor x_{ij}^n \rfloor$ and $\lceil x_{ij}^n \rceil$ be the integers such that $\lfloor x_{ij}^n \rfloor \leqslant x_{ij}^n \leqslant \lceil x_{ij}^n \rceil$ and $\lfloor x_{ij}^n \rfloor + 1 = \lceil x_{ij}^n \rceil$. Then x_{ij}^n is rounded with probability $p = x_{ij}^n - \lfloor x_{ij}^n \rfloor$ to $\lceil x_{ij}^n \rceil$ and with probability $1-p$ to $\lfloor x_{ij}^n \rfloor$.

Tau-leaping, SSA, and the Euler forward and backward integrations conserve the total number of molecules. By rounding the real numbers from the macroscopic integration to integers as above, the mean value of the total number of molecules is conserved. Since only the expected value is preserved in each step, there may be a drift away from the initial total number. A possible remedy is to instead apply the rounding to the macroscopic fluxes. The number of molecules x_{ij}^n is non-negative and therefore our algorithm for the diffusion is stable in the mean.

In conclusion, the algorithm for simulation of the chemical system in the time interval $[0,T]$ is

Algorithm 3.5

1. Assemble Q, specify ϵ_2 and ϵ_3, let $t^0 = 0$ and $n = 0$ and initialize $\mathbf{x}^0$.
2. Compute the timestep Δt^n at t^n.
3. Classify the vertices for each species i separately.
   (a) Decide if a vertex j is a macro vertex, if yes then $j \to \mathcal{V}_m^i$
   (b) Decide if a vertex j is a SSA vertex, if yes then $j \to \mathcal{V}_{SSA}^i$
   (c) The tau vertices are in $\mathcal{V}_\tau^i = \mathcal{V} \setminus \mathcal{V}_m^i \setminus \mathcal{V}_{SSA}^i$
4. Determine the approximation of the diffusion for all the pairs (j,k) of vertices with an edge between them for each species i separately.
   (a) If $j,k \in \mathcal{V}_m^i$, then use macroscopic diffusion for (j,k) and (k,j), $(j,k),(k,j) \to \mathcal{E}_m^i$
   (b) Compute Q_m
   (c) If $j \in \mathcal{V}_\tau^i$ and $k \in \mathcal{V}_\tau^i \cup \mathcal{V}_m^i$, then use the tau-leap method for the diffusion from k to j and from j to k, $(j,k),(k,j) \to \mathcal{E}_\tau^i$
   (d) If $j \in \mathcal{V}_{SSA}^i$ and $k \in \mathcal{V}$, then use SSA for the diffusion from k to j and from j to k, $(j,k),(k,j) \to \mathcal{E}_{SSA}^i$
5. First step in Algorithm 3.1 with macroscopic diffusion for all species i and $(j,k) \in \mathcal{E}_m^i$, $\mathcal{A} = \mathcal{D}_m$, conversion of real numbers to integers.
6. Second step in Algorithm 3.1 with tau-leap diffusion for all species i and $(j,k) \in \mathcal{E}_\tau^i$, $\mathcal{B} = \mathcal{D}_\tau$.
7. Third step in Algorithm 3.1 with SSA for diffusion for all species i and $(j,k) \in \mathcal{E}_{SSA}^i$ and all reactions at every vertex for all species, $\mathcal{C} = \mathcal{M} + \mathcal{D}_{SSA}$.
8. Fourth step in Algorithm 3.1 with tau-leap diffusion.
9. Fifth step in Algorithm 3.1 with macroscopic diffusion.
10. The trajectory is advanced to t^{n+1}. If $t^{n+1} \geqslant T$, then STOP. Otherwise, let $n := n+1$ and go to 2.

Other methods for approximating the diffusion can replace SSA, tau-leaping, and macroscopic integration in the algorithm as long as there are effective criteria to choose between them.

3.5. Computational work

The computational work for reactions and approximation of the diffusion in one timestep of length Δt^n can be estimated for SSA, tau-leap, and the macro level as follows assuming that the reaction events are few.

The expected value of the sum of L timesteps δt_ℓ in one cycle of SSA in step 7 of Algorithm 3.5 is according to (3.8)

$$E\left[\sum_{\ell=1}^{L}\delta t_\ell\right]=\sum_{\ell=1}^{L}1/\beta_\ell\approx L/\beta_1,\quad \beta_1=\sum_{j=1}^{K}\sum_{r=1}^{R}a_r\left(\mathbf{x}_j^n\right)+\sum_{i=1}^{N}\sum_{(j,k)\in\mathcal{E}_{SSA}^i}Q_{kj}x_{ij}^n.$$

Hence, the expected number of steps to reach Δt^n is $L\approx\Delta t^n\beta_1$. The major part of the computational work in every SSA-step is spent on determining which reaction or diffusion event occurs. Let $|\mathcal{E}_{SSA}^i|$ denote the number of elements in $\mathcal{E}_{SSA}^i$. Then this search is proportional to the sum of $|\mathcal{E}_{SSA}^i|$ and the number of reaction channels KR. The total work with straightforward SSA in $[t^n,t^{n+1}]$ is then

$$W_{SSA}\approx\Delta t^n c_{SSA}\beta_1(KR+|\mathcal{E}_{SSA}|),\quad \mathcal{E}_{SSA}=\bigcup_{i=1}^{N}\mathcal{E}_{SSA}^i, \tag{3.26}$$

for some constant c_{SSA}.

With tau-leaping in $[0,\Delta t^n]$, a Poisson number is generated for each pair (j,k) in $\mathcal{E}_\tau^i$. Using Knuth's algorithm [34] the computational work for one random number is $c_\tau\Delta t^n Q_{kj}x_{ij}^n$ [1] at t^n where c_τ is a small constant. Then the total work is

$$W_\tau\approx\Delta t^n c_\tau\sum_{i=1}^{N}\sum_{(j,k)\in\mathcal{E}_\tau^i}Q_{kj}x_{ij}^n. \tag{3.27}$$

Faster methods for generation of Poisson numbers are found in [32]. The gain in computing time when moving one pair (j,k) for species i from being simulated by SSA to the tau-leap method is derived from (3.26) and (3.27)

$$\Delta W_{SSA\to\tau}\approx\Delta t^n Q_{kj}x_{ij}^n(c_{SSA}(KR+|\mathcal{E}_{SSA}|)-c_\tau). \tag{3.28}$$

Since $c_{SSA}(KR+|\mathcal{E}_{SSA}|)-c_\tau$ very likely is positive, the savings in CPU time are substantial, especially when x_{ij}^n is large. This comparison is less favorable for tau-leaping when SSA is replaced by NSM as we do in Section 4.

The work in the macroscopic diffusion is independent of the size of the elements of $\mathbf{x}_i^n$. The computing time is proportional to the dimension of Q_m. With the Euler forward method the cost of evaluating $Q_{m,kj}x_{ij}^n$ in (3.23) is denoted by c_m which is a small constant. Then the work in the time interval is

$$W_m\approx\sum_{i=1}^{N}\sum_{(j,k)\in\mathcal{E}_m^i}c_m=c_m|\mathcal{E}_m|,\quad \mathcal{E}_m=\bigcup_{i=1}^{N}\mathcal{E}_m^i. \tag{3.29}$$

The Euler backward method is implicit in (3.23) and a system of linear equations has to be solved. The most efficient way of solving this system is a multigrid method with a work proportional to $|\mathcal{V}_m|\log|\mathcal{V}_m|$ where $|\mathcal{V}_m|$ is the number of macroscopic vertices for all species. Essentially it has the form of (3.29) but with a larger c_m. The marginal gain in transferring a pair of vertices (j,k) from $\mathcal{E}_\tau^i$ to $\mathcal{E}_m^i$ is then

$$\Delta W_{\tau\to m}\approx\Delta t^n c_\tau Q_{kj}x_{ij}^n-c_m. \tag{3.30}$$

Clearly, $\Delta W_{\tau\to m}>0$ for a sufficiently large x_{ij}^n.

4. Numerical results

In this section we test the adaptive diffusion approximation in a few different cases. In the first example in Section 4.1, we illustrate the principles of the method in a 2D model problem with a single diffusing species and the convergence of the method is confirmed in a `Matlab` implementation. In Section 4.2 we apply the algorithm to a more biologically relevant geometry in 3D: a model of a yeast cell. The potential of our approach is shown in physiologically relevant scenarios.

4.1. Diffusion

The adaptive method in Algorithm 3.5 is applied to diffusion of one species with the diffusion constant $\gamma=0.001$ on the unit square $0\leqslant x,y\leqslant 1$ in 2D. The mesh is generated with the PDE toolbox in `Matlab` and has 103 vertices. The vertices and the edges in the primal mesh and the subvolumes in the dual mesh are displayed in Fig. 4.1. With the parameter c, the initial state for the concentration ϕ in the vertices is

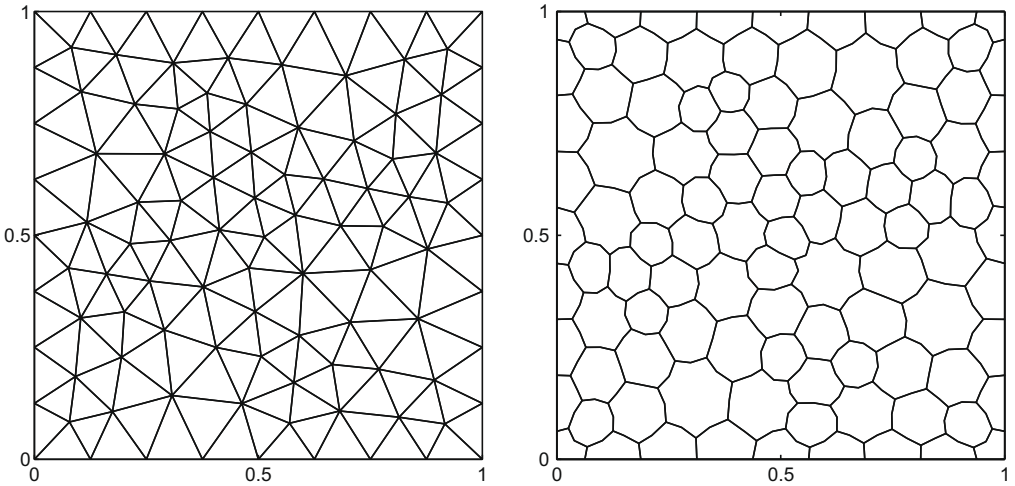

Fig. 4.1. The primal mesh (left) and the dual mesh (right).

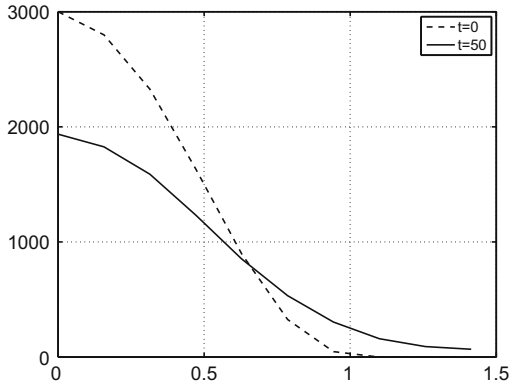

Fig. 4.2. The initial and final FE solution along the main diagonal.

$$\phi(x,y) = \begin{cases} 0.5c(1+\cos(\pi r)) & \text{if } r = \sqrt{x^2+y^2} \leqslant 1, \\ 0 & \text{otherwise.} \end{cases} \tag{4.1}$$

The corresponding number of molecules x_j in the subvolumes $\mathcal{C}_j$, $j = 1,\dots,K$, is rounded to an integer as in Algorithm 3.5. The macroscopic diffusion is advanced in time by the backward Euler method.

The solution is simulated in the time interval $0 \leqslant t \leqslant 50$. A FE solution is computed with $\Delta t = 0.05$ for comparison and it is denoted by $\boldsymbol{\phi}^{FE}$. The initial and final FE solutions when $c = 3000$ are plotted in Fig. 4.2 along the main diagonal from the origin. Since the diffusion propensities are linear, the mean values of the concentrations at the vertices taken over the trajectories are given by $\boldsymbol{\phi}^{FE}$. The relative difference δ between our solution $\boldsymbol{\phi}$ and the FE solution is measured as

$$\delta = \frac{1}{\max_j \phi_j^{FE}} \sqrt{\sum_j \left(\phi_j - \phi_j^{FE}\right)^2 |\mathcal{C}_j|}. \tag{4.2}$$

The diffusion is partitioned individually in every trajectory into SSA, tau-leap or macroscopic simulation with the tolerances $\epsilon_1 = 0.01$ and $\epsilon_2 = \epsilon_3 = 0.03$ defined in Sections 3.2.2, 3.2.3 and 3.3.

The average solution obtained after adaptive simulations of m trajectories is compared to the FE solution in Table 4.1 at $t = 50$ with $c = 3000$ in the initial state (4.1). When the number of trajectories is increased by a factor 10, the difference is reduced by about $1/\sqrt{10}$ as expected by the central limit theorem. The total number of failures with negative copy numbers in the tau-leap method is k_τ and the fraction of failures at all timesteps at all tau-leap vertices is f_τ in the table. The number of necessary corrections is low with $\epsilon_1 = 0.01$.

The variation of the timestep in one trajectory is found to the left in Fig. 4.3. The percentage of tau-leap vertices and SSA vertices is shown in the right panel of the same figure. The number of tau-leap vertices increases as the number of cells with few molecules decreases, cf. Fig. 4.2. There is no macroscopic diffusion in this case.

Table 4.1
Simulations with m trajectories. Comparison with the FE solution (δ defined in (4.2)) and the number of sign errors in tau-leap (k_τ) and the quotient between failures and the total number of tau-leap steps (f_τ).

$\log_{10} m$	δ	k_τ	f_τ
1	10.5	39	0.00033
2	3.5	228	0.00020
3	0.90	2567	0.00022
4	0.27	24970	0.00022

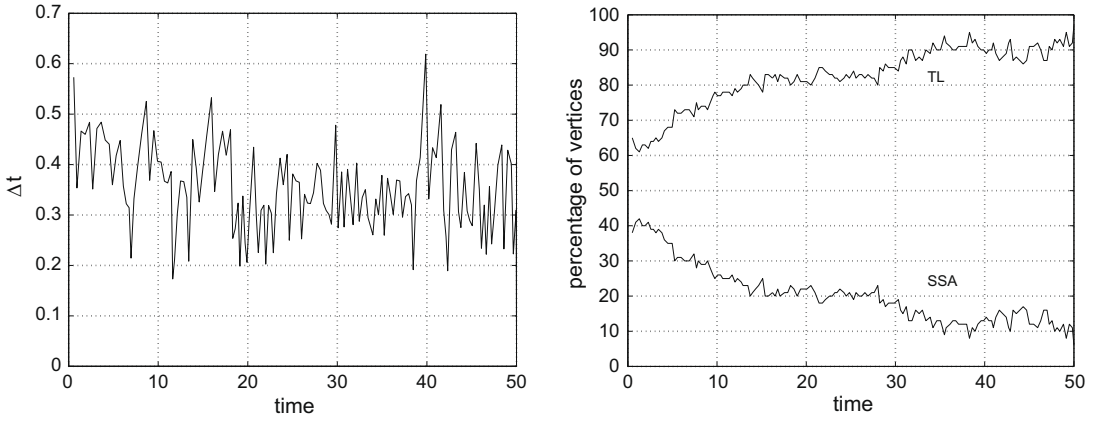

Fig. 4.3. The variation of the timestep Δt (left) and the percentage of SSA and tau-leap (TL) (right) for a single trajectory.

When c in (4.1) is increased, we expect the concentrations in the trajectory to approach the macroscopic solution; see the end of Section 2. We take $c = 3 \times 10^m$, $m = 0, 1, \ldots, 9$, and record the outcome for one trajectory. The percentage of SSA, tau-leaping, and macroscopic diffusion for the whole time interval for ten c-values is displayed to the left in Fig. 4.4. As c increases, different methods are predominant in the simulation. The convergence of $\boldsymbol{\phi}$ to $\boldsymbol{\phi}^{FE}$ at $t = 50$ is demonstrated to the right in the same figure. We expect the deviation of $\boldsymbol{\phi}$ from $\boldsymbol{\phi}^{FE}$ to decay as $1/\sqrt{c}$ in agreement with the discussion in the end of Section 2 and (2.11) (cf. also Table 4.1). In one comparison, the same timestep sequence is used for both our solution and the FE solution and the convergence is as expected. The reduction for increasing m is even faster than $1/\sqrt{10}$ per decade for large c where the macroscopic approximation, which is equivalent to FE, prevails. A trajectory determined with the automatic timestep selection is also compared to the FE solution with $\Delta t = 0.01$. Then there is a remaining error due to the time discretization for $m > 4$. The computing time is almost independent of c with m between 1 and 7 but increases linearly when the number of molecules is high ($m = 8, 9$). There are no failures in the tau-leap method except for $m = 2$ when there are 8 and $m = 3$ when there are 3.

The performance in the left panel of Fig. 4.4 is explained by a simple 1D model as follows. Let $\xi \in [0, 1]$ be the space coordinate and $x(\xi)$ the number of molecules in the interval with $x(\xi) = 0.5c(1 + \cos(\pi\xi))$ as in (4.1). Then μ and σ in (3.12) are

$$\mu \approx \Delta t\gamma x'' = -0.5c\Delta t\gamma\pi^2\cos(\pi\xi), \quad \sigma \approx h^{-1}\sqrt{4\Delta t\gamma x} = 2h^{-1}\sqrt{\Delta t\gamma c}\cos(0.5\pi\xi).$$

A somewhat stricter criterion for SSA simulation at ξ than in (3.25) is that

$$x + \mu - \kappa(\epsilon_1)\sigma \approx 0.5c(1 + (1 - \Delta t\gamma\pi^2)\cos(\pi\xi)) - 2h^{-1}\kappa\sqrt{\Delta t\gamma c}\cos(0.5\pi\xi) < 0. \tag{4.3}$$

In our case, $\Delta t\gamma\pi^2$ is small and can be ignored. Thus, (4.3) is satisfied when $\xi > 2\pi^{-1}\arccos(2\kappa\sqrt{\Delta t\gamma}/(h\sqrt{c}))$, i.e. the larger c is, the closer to 1 ξ is and the fewer vertices are updated by SSA but there is always a $\xi < 1$ no matter how large c is (cf. Fig. 4.4 where SSA is active for all c). On the other hand, if c is sufficiently small, then (4.3) is fulfilled by any ξ in $[0, 1]$ and SSA is applied everywhere. The test for macroscopic diffusion in (3.19) in the jth vertex at ξ_j is

$$\frac{\sqrt{(|Q|\mathbf{x})_j}}{x_j} \approx \frac{2\sqrt{c\gamma}\cos(0.5\pi\xi_j)}{hc\cos(0.5\pi\xi_j)^2} = \frac{2\sqrt{\gamma}}{h\sqrt{c}\cos(0.5\pi\xi_j)} \leqslant \epsilon_2. \tag{4.4}$$

If c is so large that $c \geqslant 4\gamma/(h\epsilon_2)^2$, then the condition in (4.4) is satisfied by $\xi \in [0, \xi_{\max}]$ with $\xi_{max} = 2\pi^{-1}\arccos(2\sqrt{\gamma}/(h\epsilon_2\sqrt{c}))$ which grows with increasing c. Indeed, there is no macroscopic diffusion for c below about 3×10^4 in Fig. 4.4.

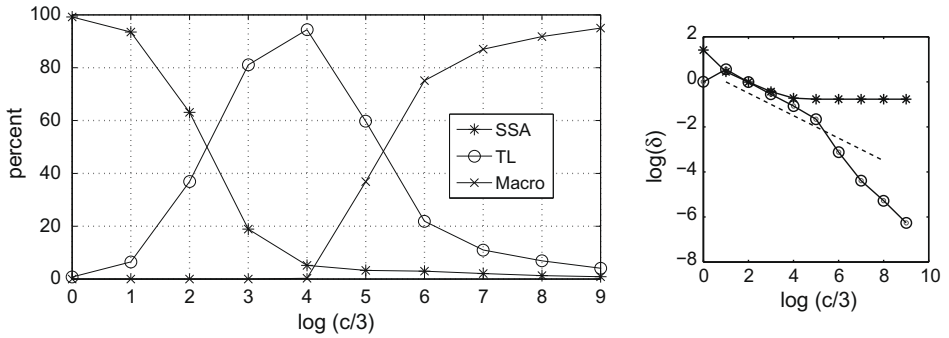

Fig. 4.4. Percentage of vertices in the time interval [0,50] classified as SSA, tau-leap (TL) and macroscopic versus concentration (left). The difference δ to the FE solution at $t = 50$ with the same timestep (o) and adaptive timesteps (*) (right). The theoretical convergence rate $1/\sqrt{10}$ per decade in c is included for comparison (dashed).

4.2. Reaction–diffusion systems in 3D

We will apply the adaptive algorithm to a biochemical reaction system in a 3D geometry in this section. The benefits of our new scheme will be evident in different modeling scenarios. There is one situation where the present algorithm will be more expensive than a purely stochastic simulation of the full system. The purpose with the following test cases is not to draw any biological conclusions, but rather to show in which cases our approach will be useful and when it is not. However, we will stay within the range of physiologically relevant values for the geometry, the copy numbers of the species and the diffusion constants. The test cases we will consider could be parts of a larger, more detailed model of a complicated biochemical pathway.

In this section we have used the URDME software [14], an efficient implementation of NSM [18] on unstructured meshes, and extended it with the hybrid functionality. The macroscopic diffusion operator is integrated by the explicit Euler forward scheme. The order between NSM and tau-leaping is here interchanged in Algorithm 3.5 as this was the splitting order determined to be most efficient for this example. The experience from operator splitting applied to the numerical solution of partial differential equations is that there may be an unexpected loss of accuracy due to the splitting in very stiff problems such as in combustion applications. Then the order of the operators may be crucial to improve the situation [49] but we have not explored that matter in our examples.

The geometry, the mesh and the assembly of the stiffness and mass matrices as well as postprocessing is handled by `COMSOL Multiphysics 4.3`. All computations in this section have been performed on a 2.0 GHz MacBook Core Duo with 2 GB RAM.

4.2.1. Model

The geometry is depicted to the left in Fig. 4.5. It is a simple model made up of three subdomains: a cube with side length 8 μm, a sphere with volume $62 \times 10^{-15} l$ or 62 *fl* and a smaller, concentric sphere with volume 4.1 *fl*. This geometry is here

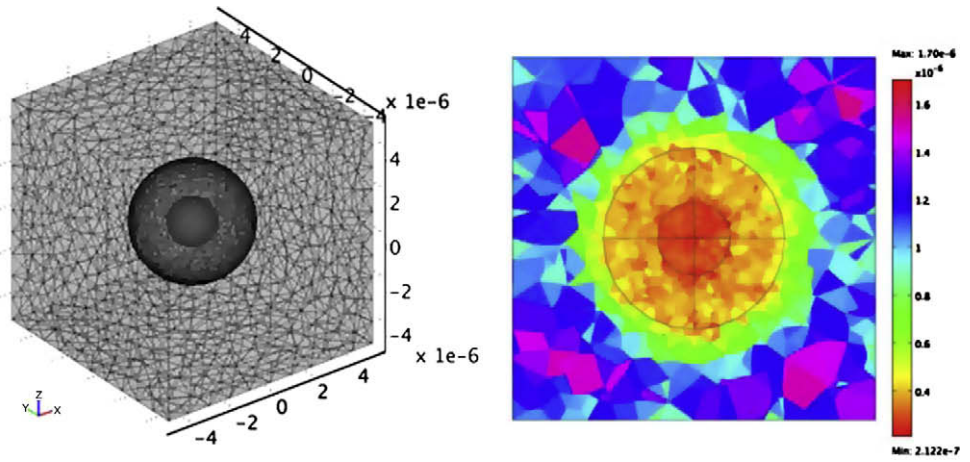

Fig. 4.5. The geometry of the yeast cell model with the nucleus, the cell, and the surrounding box (left). The variation of the subvolume size in the domain (right).

Table 4.2
The chemical reactions of the bistable model. The constants take the values $k_1 = 150\ s^{-1}, k_a = 1.2 \times 10^8\ s^{-1}\ M^{-1}, k_d = 10\ s^{-1}$ and $k_4 = 6\ s^{-1}$.

$E_A \xrightarrow{k_1} E_A + A$	$E_B \xrightarrow{k_1} E_B + B$	$E_A + B \underset{k_d}{\overset{k_a}{\rightleftharpoons}} E_AB$	$E_B + A \underset{k_d}{\overset{k_a}{\rightleftharpoons}} E_BA$
$E_AB + B \underset{k_d}{\overset{k_a}{\rightleftharpoons}} E_AB_2$	$E_BA + A \underset{k_d}{\overset{k_a}{\rightleftharpoons}} E_BA_2$	$A \xrightarrow{k_4} \emptyset$	$B \xrightarrow{k_4} \emptyset$

Table 4.3
The chemical reactions of the nuclear module. The parameter k_a takes the same value as in Table 4.2, $k_5 = 1\ s^{-1}$ and $k_{dn} = 1\ s^{-1}$.

$A + X \xrightarrow{k_a} A + X^*$	$X^* \xrightarrow{k_5} X$	$A \xrightarrow{k_{dn}} \emptyset$
$B + Y \xrightarrow{k_a} B + Y^*$	$Y^* \xrightarrow{k_5} Y$	$B \xrightarrow{k_{dn}} \emptyset$

thought of as a simplified model of a baker's yeast cell with a nucleus immersed in a cube. In the following, we will refer to the different subdomains as the cell (the two spheres together), the cytosol (the region between the two spheres) and the nucleus (the inner sphere).

The mesh has a different maximal element size in the different subdomains. It is denser inside the nucleus and in the cytoplasm than in the cell exterior with the largest elements farthest away from the cell. The total number of vertices is 6889 giving a total of 82.6×10^4 to 96.4×10^4 degrees of freedom (dofs) or variables depending on the scenario below. The size of the subvolumes is shown in color at an intersection through the center of the domain to the right in Fig. 4.5.

To introduce a clear space dependence in the model, the reaction network in our test case will consist of one cytosolic and one nuclear module. The two modules are coupled by allowing two of the species of the cytosolic module to enter the nucleus, where they can interact with the nuclear module.

The cytosolic module is a bistable system taken from [18], see Table 4.2. It was used in [19] to demonstrate that we can recover the properties of the model using unstructured meshes and in [14] as a benchmark problem. All species except A and B are confined to the cytosol region between the two spheres.

The proteins diffuse with $\gamma = 1\ \mu m^2/s$ (large proteins), and the entry of A and B into the nucleus is modeled as a diffusion event (equivalent to a first order reaction) with $\gamma = 0.01\ m^2/s$.

The modeling of the penetration of A and B into the nucleus is obviously a simplification. A more realistic approach would be to introduce additional membrane bound species at the membrane modeling receptors and pore species and let molecules pass the membrane through a series of reactions involving these additional species. Approximation of the transport as a diffusion event here serves two purposes: we avoid adding more species to the model as the idea with this example is to measure the performance of the hybrid method, and we also introduce a case where we have different diffusion constants in different regions of space. This allows us to illustrate the behavior of the partitioning algorithm in Section 4.2.4.

The reactions of the nuclear module are found in Table 4.3. This is a simple model composed of species X and Y which are activated by the proteins A and B, respectively.

The model described above is simulated in Scenario 1. In order to illustrate the potential issue of stiffness in spatial models, two additional model scenarios are considered where we include smaller, abundant molecules in the model. In Scenario 2, they will both be confined to the cytoplasm in concentrations relevant for e.g. GTP and cAMP, and in Scenario 3 one of them will initially be present only outside the cell in a macroscopic concentration relevant to e.g. a metabolite in the medium. In the first case, we will see that a purely stochastic simulation is the most efficient, in the second case the hybrid method combines NSM and tau-leap to make the simulation more efficient, and in the third case all three methods will be used by the adaptive hybrid method giving a speed-up of more than 3000 compared to NSM alone.

4.2.2. Scenario 1: Simulation of the cytosolic and nuclear module

For reference, we simulate only the cytosolic and nuclear module to a final time $t = 100$ s using URDME and a purely stochastic simulation. Initially, 300 molecules each of the enzymes E_A and E_B are spread randomly in the cytoplasm and 300 molecules of X and Y in the nucleus, all other species are zero.

The distribution of A molecules in the cytosol is depicted to the left in Fig. 4.6 and to the right there is a close-up view of the nucleus and the distribution of X^*. The CPU time of this simulation was 81 s and the number of events 53.8×10^6 (of which 87% were diffusion events), the total number of X^* was 31 molecules and a stochastic description is obviously desirable.

We simulated the same system with the adaptive hybrid algorithm. In this case, the system is not sufficiently stiff for the method to be competitive. Even though tau-leap is selected in some subvolumes, the computational savings are not enough to compensate for the overhead. The simulation took 1500 s, more than ten times slower than the pure NSM simulation.

4.2.3. Scenario 2: Two small species inside the cell

Often, small abundant species are not explicitly treated in models. Instead, it is assumed that their concentration does not change significantly during the time interval under study and their effect is modeled implicitly. It is highly plausible that

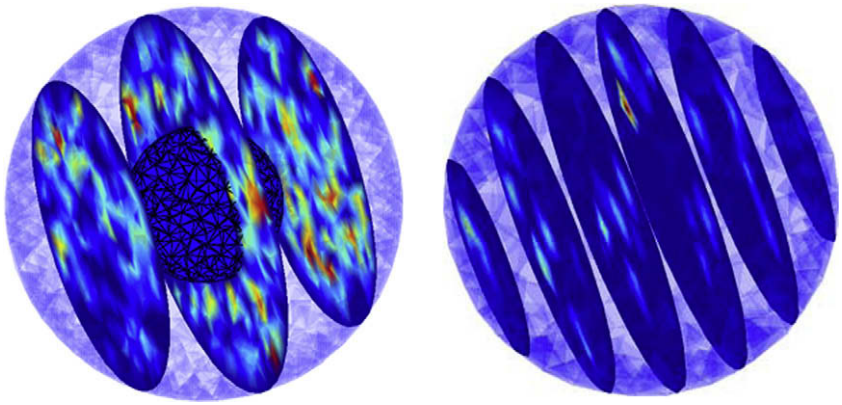

Fig. 4.6. The distribution of A in the cytosol (left) and that of X^* inside the nucleus after 100 s (right). Blue corresponds to a low number and red to a high number. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

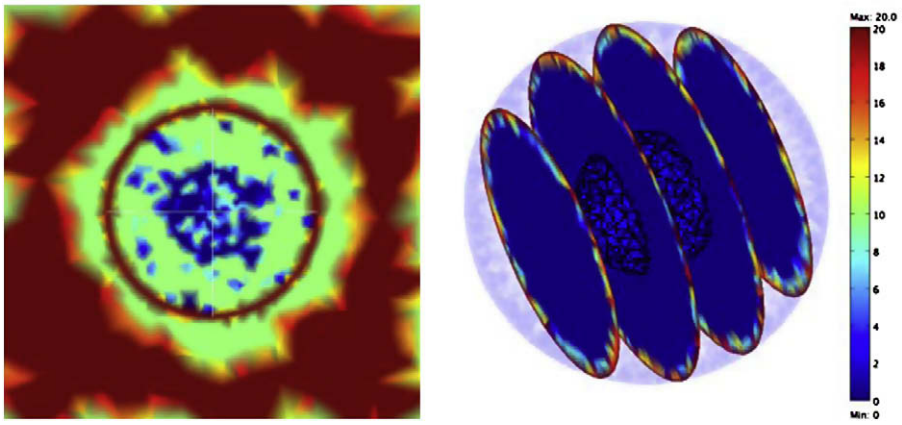

Fig. 4.7. The partitioning chosen by the adaptive hybrid algorithm for the small species is shown to the left at $t = 0.01$ s. Red color indicates a macroscopic diffusion, green tau-leap and blue that the subvolume is simulated with NSM. To the right we show the small species as it enters the cell at the plasma membrane. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

such assumptions are erroneous, at least locally in space and time, and they require more *a priori* knowledge about the system. A computationally expensive alternative avoiding such assumptions is to include them explicitly in the models. One recent example where small molecules such as ATP, GTP, GDP, and cAMP are included explicitly in a well stirred stochastic model of the Ras/cAMP/PKA pathway can be found in [12], where the levels of the guanine nucleotide pools are suggested to play an important role for the dynamics of the system.

In this scenario, two smaller species will both be confined to the cytoplasm, and can be thought of as e.g. cAMP and GTP. At $t = 0$, we simulate a step increase in the cytoplasm of the concentration of one of the species to 0.14 mM (4.2×10^6 molecules) while the other has a concentration of 10 μM (3×10^5 molecules). These numbers correspond roughly to the copy numbers of GTP and cAMP used in the model from [12]. In this test scenario, neither of the small species interact with the proteins in the cytosolic and nuclear module but they diffuse with $\gamma = 250$ m^2/s making the model stiff. We could easily let them react with e.g. A and B in a more detailed model but this is not necessary in order to illustrate the computational difficulties their inclusion introduce.

Using a pure NSM simulation to reach $t = 0.001$ s, we need 181 s of CPU time and NSM has generated 83.2×10^6 diffusion events and only 97 chemical reactions.

Next we simulate the system with our algorithm with the probability of computing negative copy numbers $\epsilon_1 = 0.05$ and the relative error tolerances $\epsilon_2 = \epsilon_3 = 0.05$ for the diffusion. The method automatically detects that the small molecules are sufficiently many to be simulated with the tau-leap method in most of the domain (about 10.4% of the total degrees of freedom are tau-leaped) and this makes the simulation 10 times faster than the pure SSA simulation. The simulation from $t = 0$ to $t = 0.001$ s took 18.6 s of CPU time. A simulation to the final time $t = 100$ s would still take about 3 weeks, but this is an improvement compared to over 7 months using only NSM.

4.2.4. Scenario 3. A small species in high concentration in the cell exterior

In this final scenario, one of the small species inside the cell is removed and instead it is present outside the cell in an initial concentration of 20 mM and it can be thought of as e.g. a small metabolite in the medium. We let it enter the cell by slow diffusion events over the plasma membrane ($\gamma = 0.25$ nm^2/s). In this case, it is impossible to simulate the system on a time scale relevant to the dynamics in the nuclear and cytosolic module. It took 558 s with only NSM using URDME to evolve the system to $t = 1 \times 10^{-5}$ s. Not a single reaction event occurred in this time interval.

The adaptive method detects that macroscopic diffusion can be used for the small species in almost all subvolumes in the exterior of the cell. Since the work for macroscopic diffusion is small, this saves a lot of computational time. Our algorithm simulates the system to $t = 0.01$ s using 187 s of CPU time, about 3000 times faster than SSA alone. The simulation cost for the hybrid algorithm is about the same as in Scenario 2 above. Adding species in macroscopic concentration does not affect the CPU time while it makes a purely stochastic simulation of the system nearly impossible.

The left panel of Fig. 4.7 shows the partitioning into SSA, tau-leap and macroscopic degrees of freedom for the small, extracellular species. Near the plasma membrane, the subvolumes become smaller, cf. Fig. 4.5, and the hybrid algorithm chooses tau-leap instead of the macroscopic solver. At the membrane, the much smaller diffusion constant allows for a macroscopic treatment even though the subvolumes are smaller there too. In most of the cytoplasm, tau-leap is chosen since the concentration is not yet sufficiently large. Inside and near the nucleus, NSM is the dominant method. The small species enters the cell through the plasma membrane in the right panel of Fig. 4.7.

5. Conclusions

An algorithm has been proposed and analyzed for chemical systems with a spatial variation modeled by the reaction–diffusion master equation (RDME). The assumption is that diffusion events outnumber reaction events in a realization of the system and a special treatment of the diffusion is necessary. The RDME operator is split into three parts and advanced in time by a Strang splitting procedure. The timesteps are chosen adaptively based on an estimated local error. The molecular diffusion is simulated at the mesoscopic level by the stochastic simulation algorithm (SSA) or tau-leaping and at the macroscopic level by the diffusion equation. The algorithm switches automatically and dynamically between the different approximations with a control of the errors. All reactions are handled by SSA. The method is applied to diffusion in 2D and a model of a yeast cell in 3D with more than twelve species and realistic parameters in three scenarios. With few molecules, SSA is the method of choice. The diffusion of species with higher concentrations are simulated with tau-leaping and for the largest concentrations, macroscopic diffusion is switched on. The CPU time is reduced by a factor 3000 in one example allowing macroscopic diffusion instead of diffusion by SSA making simulations in short intervals feasible on a desktop computer.

The method proposed in this paper treats all reactions stochastically with NSM independently of the choice of method for diffusion. The algorithm deals efficiently with stiffness due to diffusion. However, once the cost due to diffusion events has been reduced, simulations might still be slow due to fast reactions. A natural next step to further improve the current method is to develop similar criteria for the reaction contributions and devise appropriate approximations for the reaction terms as well. An alternative is to advance the reactions by modifications of SSA developed for stiff, well stirred systems, see e.g. [10,17].

Acknowledgment

The authors have benefitted from fruitful discussions with Stefan Engblom. Detailed referee reports helped us improve the presentation in the paper. Financial support has been obtained from the Swedish Foundation for Strategic Research and the Swedish Graduate School in Mathematics and Computing at Uppsala University.

References

[1] J.H. Ahrens, U. Dieter, Computer methods for sampling from gamma beta Poisson and binomial distributions, Computing 12 (1974) 223–246.
[2] D.F. Anderson, Incorporating postleap checks in tau-leaping, J. Chem. Phys. 128 (2008) 054103.
[3] A. Auger, P. Chatelain, P. Koumoutsakos, R-leaping: accelerating the stochastic simulation algorithm by reaction leaps, J. Chem. Phys. 125 (2006) 084103.
[4] F. Baras, M. Malek Mansour, Reaction–diffusion master equation: a comparison with microscopic simulations, Phys. Rev. E 54 (1996) 6139–6148.
[5] N. Barkai, S. Leibler, Circadian clocks limited by noise, Nature 403 (2000) 267–268.
[6] O.G. Berg, P.H. von Hippel, Diffusion-controlled macromolecular interactions, Ann. Rev. Biophys. Chem. 14 (1985) 131–160.
[7] D. Bernstein, Simulating mesoscopic reaction–diffusion systems using the Gillespie algorithm, Phys. Rev. E 71 (2005) 041103.
[8] K. Burrage, T. Tian, P. Burrage, A multi-scaled approach for simulating chemical reaction systems, Prog. Biophys. Mol. Biol. 85 (2004) 217–234.
[9] Y. Cao, D. Gillespie, L. Petzold, Avoiding negative populations in explicit Poisson tau-leaping, J. Chem. Phys. 123 (2005) 054104.
[10] Y. Cao, D. Gillespie, L. Petzold, Multiscale stochastic simulation algorithm with stochastic partial equilibrium assumption for chemically reacting systems, J. Comput. Phys. 206 (2005) 395–411.
[11] Y. Cao, D. Gillespie, L. Petzold, Efficient step size selection for the tau-leaping simulation method, J. Chem. Phys. 124 (2006) 044109.
[12] P. Cazzaniga, D. Pescini, D. Besozzi, G. Mauri, S. Colombo, E. Martegani, Modeling and stochastic simulation of the Ras/cAMP/PKA pathway in the yeast Saccharomyces cerevisiae evidences a key regulatory function for intracellular guanine nucleotides pools, J. Biotechnol. 133 (3) (2008) 377–385.
[13] A. Chatterjee, D.G. Vlachos, M.A. Katsoulakis, Binomial distribution based τ-leap accelerated stochastic simulation, J. Chem. Phys. 122 (2005) 024112.
[14] J. Cullhed, S. Engblom, A. Hellander, The URDME manual version 1.0. Technical Report 2008-022, Dept. of Information Technology, Uppsala University, Uppsala, Sweden, 2008, <http://www.it.uu.se/research>.

[15] M. Dobrzyński, J.V. Rodríguez, J.A. Kaandorp, J.G. Blom, Computational methods for diffusion-influenced biochemical reactions, Bioinformatics 23 (2007) 1969–1977.
[16] K. Doubrovinski, M. Howard, Stochastic model for Soj relocation dynamics in *Bacillus subtilis*, Proc. Natl. Acad. Sci. USA 102 (2005) 9808–9813.
[17] W. E, D. Liu, E. Vanden-Eijnden, Nested stochastic simulation algorithm for chemical kinetic systems with disparate rates, J. Comput. Phys. 221 (2007) 158–180.
[18] J. Elf, M. Ehrenberg, Spontaneous separation of bi-stable biochemical systems into spatial domains of opposite phases, Syst. Biol. 1 (2004) 230–236.
[19] S. Engblom, L. Ferm, A. Hellander, P. Lötstedt, Simulation of stochastic reaction–diffusion processes on unstructured meshes, SIAM J. Sci. Comput. 31 (2009) 1774–1797.
[20] R. Erban, J. Chapman, Stochastic modelling of reaction–diffusion processes: algorithms for bimolecular reactions, Phys. Biol. 6 (2009) 046001.
[21] R. Erban, J. Chapman, P.K. Maini, A practical guide to stochastic simulations of reaction–diffusion processes. Technical Report, Mathematical Institute, University of Oxford, Oxford, United Kingdom, 2008.
[22] D. Fange, J. Elf, Noise-induced min phenotypes in *E. coli*, PLoS Comput. Biol. 2 (2006) 637–648.
[23] C.W. Gardiner, Handbook of Stochastic Methods, third ed., Springer Series in Synergetics, Springer-Verlag, Berlin, 2004.
[24] M.A. Gibson, J. Bruck, Efficient exact stochastic simulation of chemical systems with many species and many channels, J. Phys. Chem. 104 (2000) 1876–1889.
[25] D.T. Gillespie, A general method for numerically simulating the stochastic time evolution of coupled chemical reactions, J. Comput. Phys. 22 (4) (1976) 403–434.
[26] D.T. Gillespie, Approximate accelerated stochastic simulation of chemically reacting systems, J. Chem. Phys. 115 (4) (2001) 1716–1733.
[27] E. Hairer, S.P. Nørsett, G. Wanner, Solving Ordinary Differential Equations I. Nonstiff Problems, second ed., Springer-Verlag, Berlin, Heidelberg, 1993.
[28] J. Hattne, D. Fange, J. Elf, Stochastic reaction–diffusion simulation with MesoRD, Bioinformatics 21 (2005) 2923–2924.
[29] S.A. Isaacson, The reaction–diffusion master equation as an asymptotic approximation of diffusion to a small target, SIAM J. Appl. Math. 70 (2009) 77–111.
[30] S.A. Isaacson, C.S. Peskin, Incorporating diffusion in complex geometries into stochastic chemical kinetics simulations, SIAM J. Sci. Comput. 28 (2006) 47–74.
[31] N.L. Johnson, On an extension of the connexion between Poisson and χ_2 distributions, Biometrika 46 (1959) 352–363.
[32] N.L. Johnson, S. Kotz, A.W. Kemp, Univariate Discrete Distributions, Wiley, New York, 1992.
[33] N.G. van Kampen, Stochastic Processes in Physics and Chemistry, fifth ed., Elsevier, Amsterdam, 2004.
[34] D.E. Knuth, The art of computer programming, Seminumerical Algorithms, vol. II, Addison Wesley, Reading, MA, 1969.
[35] T.G. Kurtz, Solutions of ordinary differential equations as limits of pure jump Markov processes, J. Appl. Prob. 7 (1970) 49–58.
[36] T.G. Kurtz, Limit theorems for sequences of jump Markov processes approximating ordinary differential processes, J. Appl. Prob. 8 (1971) 344–356.
[37] T.G. Kurtz, Strong approximation theorems for density dependent Markov chains, Stoch. Proc. Appl. 6 (1978) 223–240.
[38] S. Lampoudi, D. Gillespie, L. Petzold, The multinomial simulation algorithm for discrete stochastic simulation of reaction–diffusion systems, J. Chem. Phys. 130 (2009) 094104.
[39] H. Li, Y. Cao, D. Gillespie, L. Petzold, Algorithms and software for stochastic simulation of biochemical reacting systems, Biotechnol. Prog. 24 (2008) 56–61.
[40] G.I. Marchuk, Splitting and alternating direction methods, in: P.G. Ciarlet, J.L. Lions (Eds.), Handbook of Numerical Analysis, North-Holland, Amsterdam, 1990, pp. 197–462.
[41] H.H. McAdams, A. Arkin, It's a noisy business genetic regulation at the nanomolar scale, Trends Gen. 15 (1999) 65–69.
[42] R. Metzler, The future is noisy: the role of spatial fluctuations in genetic switching, Phys. Rev. Lett. 87 (2001) 068103.
[43] J. Pahle, Biochemical simulations: stochastic, approximate stochastic and hybrid approaches, Briefings Bioinf. 10 (2009) 53–64.
[44] M.F. Pettigrew, H. Resat, Multinomial tau-leaping method for stochastic kinetic simulations, J. Chem. Phys. 126 (2007) 084101.
[45] M. Rathinam, L.R. Petzold, Y. Cao, D. Gillespie, Consistency and stability of tau-leaping schemes for chemical reaction systems, Multiscale Model. Simul. 4 (2005) 867–895.
[46] D. Rossinelli, B. Bayati, P. Koumoutsakos, Accelerated stochastic and hybrid method for spatial simulations of reaction–diffusion systems, Chem. Phys. Lett. 451 (2008) 136–140.
[47] P. Sjöberg, O.G. Berg, J. Elf, Taking the reaction–diffusion master equation to the microscopic limit. Technical Report, Dept. of Molecular Biology, Uppsala University, Uppsala, Sweden, 2009. Available from: <arXiv:0905.4629v1>.
[48] J.G. Skellam, The frequency distribution of the difference between two Poisson variates belonging to different populations, J. Royal Statist. Soc. 109 (1946) 296.
[49] B. Sportisse, An analysis of operator splitting techniques in the stiff case, J. Comput. Phys. 161 (2000) 140–168.
[50] J. Strackee, J.J. Denier van der Gon, The frequency distribution of the difference between two Poisson variates, Statistica Neerlandica 16 (1962) 17–23.
[51] G. Strang, On the construction and comparison of difference schemes, SIAM J. Numer. Anal. 5 (1968) 506–517.
[52] A.B. Stundzia, C.L. Lumsden, Stochastic simulation of coupled reaction–diffusion processes, J. Comput. Phys. 127 (1996) 196–207.
[53] T. Tian, K. Burrage, Binomial leap methods for simulating stochastic chemical kinetics, J. Chem. Phys. 121 (2004) 10356–10364.