Paper

September 9, 2013

1 INTRODUCTION

- General importance of Stochastic Chemical Reaction Networks (SCRNs) and why they are necessary
- Mention SSA and other methods and their limitations (SDM, PPM, tau-leaping, ssSSA) [“expert systems”]
- Describe what multiscale networks are (multiscale in time and copy numbers)
- Argue about the importance of having tools for simulating such systems

In this paper we extend the known simulation methods for PDMP models to an adaptive scheme that incorporates multiple scales in both time and species copy-numbers and also takes advantage of the time-scale separation of subnetworks with fast dynamics. The rest of this section formally introduces Stochastic Chemical Reaction Networks and Piecewise Deterministic Markov Processes. In section 2 we then recapitulate the existing theoretical foundation followed by the description of our implementation in section 3. Thereafter we show numerical examples in section 4 and provide a discussion of our approach and conclusions in section 5.

Stochastic Chemical Reaction Network

A Stochastic Chemical Reaction Network (SCRN) with s_0 species given by x_i with $i = 1, \ldots, s_0$ and r_0 reactions is defined by

- the initial population $x_i(0)$ of each species $i = 1, \ldots, s_0$
- the reactions $\sum_{i=1}^{s_0} \nu_{ik} x_i \longrightarrow \sum_{i=1}^{s_0} \nu'_{ik} x_i$ for each reaction $k = 1, \ldots, r_0$ (we define the stochiometry vector as $\xi_k = \nu'_k - \nu_k$)
- the reaction propensities $\lambda_k(x)$ for each reaction $k = 1, \ldots, r_0$

The corresponding Markov Jump Process of the network in the random-time-change representation is

$$X(t) = x(0) + \sum_{k=1}^{r_0} Y_k \left(\int_0^t \lambda_k \left(X(s) \right) ds \right) (\nu'_k - \nu_k)$$

where the $\{Y_k : \; k = 1, \ldots, r_0\}$ is a family of independent Poisson processes.

In this work we limit ourselves to the case of mass-action-kinetics with constitutive, unary and binary reactions. We can then write the propensities as

λ_k	Reaction	ν_k
κ'_k	$\emptyset \to stuff$	0
$\kappa'_k x_i$	$S_i \to stuff$	e_i
$\kappa'_k V^{-1} x_i x_j$	$S_i + S_j \to stuff$	$e_i + e_j$
$\kappa'_k V^{-1} x_i (x_i - 1)$	$2S_i \to stuff$	$2e_i$

where V corresponds to the volume of the system and the κ'_k correspond to the molecular reaction rates.

Simulating a SCRN can become computationally expensive when the system is stiff (i.e. some reactions occur on a very small timescale, but are not important for the global behaviour). In such cases an approximation of the SCRN can provide considerable speedups in simulation time. One possibility for such an approximation is to split the fully stochastic system into a stochastic and a deterministic part which can be modeled as a Piecewise Deterministic Markov Process which is described in the following subsection. Another popular method for approximating SCRNs is *τ-leaping* where multiple reactions are lumped into one simulation step by approximating the number of reactions with a poissonian random variable [3].

Piecewise Deterministic Markov Processes

Here we give a short description of Piecewise Deterministic Markov Processes (PDMP) in the context of Stochastic Chemical Reaction Networks (SCRNs). We refer to [2] for more details.

The state of a PDMP consists of a set of continuous and a set of discrete variables and both a vector field describing the dynamics of the continuous variables and a jump intensity function describing stochastic reactions. In the context of a SCRN the reactions $1, \ldots, r_0$ are also partitioned into two disjunct sets

$$M_D = \{k | \text{ reaction } k \text{is deterministic}\}$$

$$M_S = \{k | \text{ reaction } k \text{is stochastic}\}$$

and the species $1, \ldots, s_0$ are partitioned into two disjunct sets

$$S_D = \{i | \text{ species } i \text{is discrete}\}$$

$$S_C = \{i|\text{ species }i\text{is continuous}\}$$

where it has to be $\nu'_{ik} = \nu_{ik}$ for $i \in S_D$ and $k \in M_D$, i.e. discrete species can only be changed by stochastic reactions.

The vector field of the PDMP is then defined as

$$f:\ \mathbb{R}^{s_0} \to \mathbb{R}^{s_0},\ \frac{d}{dt}x_i(t) = f(x) = \sum_{k \in M_D} \lambda_k(x(t))(\nu'_{ik} - \nu_{ik})\ \text{ for } i \in S_C$$

and the jump intensity function is defined as

$$\Lambda_k(x) = \lambda_k(x)\ \text{ for } k \in M_S$$

In the random-time-change representation the PDMP can be written as

$$X(t) = x(0) + \sum_{k \in M_D} \int_0^t \lambda_k(X(s))\,ds\,(\nu'_k - \nu_k) + \sum_{k \in M_S} Y_k\left(\int_0^t \lambda_k(X(s))\,ds\right)(\nu'_k - \nu_k)$$

where the $\{Y_k:\ k \in M_S\}$ is a family of independent Poisson processes.

One should note that the choice of a suitable partitioning is a non-trivial problem and for SCRNs with high variation over time there might not be a suitable partitioning. This is the motivation for introducing an adaptive scheme for approximating SCRNs.

2 MATHEMATICAL OVERVIEW

In this section we summarize existing theoretical results concerning stochastic models and SCRNs. We start by recapitulating the multiscale framework described by Kang et al. [4] followed by a short summary concerning averaging of fast subnetworks.

Multiscale SCRNs

Kang et al. [4] describe a framework for separating both time-scales and copy-number-scales in SCRNs with mass action kinetics. To this end they define scaling parameters $\alpha_i \geq 0$ for $i = 1, \ldots, s_0$, β_k for $k = 1, \ldots, r_0$ and a large parameter N_0 and embed the original process $X(t)$ into a family of processes parameterized by N

$$X_i^N(t) = \left(\frac{N}{N_0}\right)^{\alpha_i} X_i(t)$$

$$Z_i^N(t) = N^{-\alpha_i} X_i^N(t)$$

$$\kappa_k = \begin{cases} N^{-\beta_k}\kappa'_k & \text{for unary reactions} \\ N^{-\beta_k}\kappa'_k V^{-1} & \text{for binary reactions} \end{cases}$$

The propensities for the parametrized process can now be written as

Reaction	Propensity	$\lambda_k^N(z)$
$\emptyset \to$ stuff	$N^{\beta_k}\lambda_k^N(z)$	κ_k
$S_i \to$ stuff	$N^{\beta_k+\alpha_i}\lambda_k^N(z)$	$\kappa_k z_i$
$S_i + S_j \to$ stuff	$N^{\beta_k+\alpha_i+\alpha_j}\lambda_k^N(z)$	$\kappa_k z_i z_j$
$2S_i \to$ stuff	$N^{\beta_k+2\alpha_i}\lambda_k^N(z)$	$\kappa_k z_i(z_i - N^{-\alpha_i})$

With these definitions the markov process of the SCRN can be written as

$$Z_i^N(t) = Z_i^N(0) + N^{-\alpha_i}\sum_{k=1}^{r_0} Y_k\left(\int_0^t N^{\beta_k+\alpha\cdot\nu_k}\lambda_k^N\left(Z^N(s)\right)ds\right)(\nu'_{ik} - \nu_{ik})$$

where $\alpha \cdot \nu_k$ denotes the dot product of vectors α and ν_k.

In the limit $N \to \infty$ this process could explode if the exponent $-\alpha_i$ outside of the Poisson processes doesn't cancel the exponents $\beta_k + \alpha \cdot \nu_k$ inside of the Poisson processes. To make sure that the process is well-behaved for $N \to \infty$ the constraint $\alpha_i \geq \beta_k + \alpha \cdot \nu_k$ for $i = 1, \ldots, s_0$, the so called the *species balance condition*, is introduced.

If the *species balance condition* is satisfied, Kang et al. [4] show that the process $Z_i^N(t)$ converges to a PDMP in the limit $N \to \infty$. Thereby each reaction term

$$N^{-\alpha_i}Y_k\left(\int_0^t N^{\beta_k+\alpha\cdot\nu_k}\lambda_k^N\left(Z^N(s)\right)ds\right)(\nu'_{ik} - \nu_{ik})$$

converges to

$$\begin{cases} 0 & \text{if } \alpha_i > \beta_k + \alpha\cdot\nu_k \\ Y_k\left(\int_0^t \lambda_k^\infty\left(Z(s)\right)ds\right)(\nu'_{ik} - \nu_{ik}) & \text{if } \alpha_i = \beta_k + \alpha\cdot\nu_k = 0 \\ \left(\int_0^t \lambda_k^\infty\left(Z(s)\right)ds\right)(\nu'_{ik} - \nu_{ik}) & \text{if } \alpha_i = \beta_k + \alpha\cdot\nu_k > 0 \end{cases}$$

where $\lambda_k^\infty(z) = \begin{cases} \kappa_k & \text{for constitutive reactions} \\ \kappa_k z_i & \text{for unary reactions of species } i \\ \kappa_k z_i z_j & \text{for binary reactions of species } i \text{and } j \end{cases}$.

If N_0 is a large enough number then we expect a similar behaviour for the original process $X_i(t) = N_0^{\alpha_i} Z_i^{N_0}$. Intuitivly this allows us to approximate species with high copy numbers with deterministic dynamics as the fluctuations become less important.

Note: If the above constraints are not fulfilled Kang et al. [4] introduce other constraints for which the process still converges to a PDMP.

So far the above convergence result gives us an approximation of the original process depending on the copy-number-scales. Another separation can occur on the time-scales: A subnetwork with dynamics which are much faster than the dynamics of the surrounding network will reach it's stationary average (provided that a stationary distribution exists) before influencing the surrounding network. Such a subnetwork can then be collapsed to it's stationary average. This is called averaging and is described in the following subsection.

Averaging of fast subnetworks

Given a subnetwork with *fast* dynamics compared to the surrounding network (i.e. the species that are directly influenced by the subnetwork) we partition the reactions that are involved in the subnetwork into two sets, one that is only influencing the subnetwork, and the other one that is connecting the subnetwork to the surrounding network. We will then define a timescale for both of these sets of reactions so that we can define a formal criterion to check whether the subnetwork has *fast* dynamics. The final step is then to compute the stationary distribution of the subnetwork.

Following is a more formal description of the averaging and an elaboration on three different strategies for the computation of the stationary distribution of a subnetwork.

Formal description

Consider a SCRN with s_0 species and r_0 reactions. The general idea is to identify subnetworks of *fast* species and average the state of those subnetworks according to the stationary distribution. Formally, given a subset of species $Q = \{q_1, \dots, q_m\}$ we identify the set of reactions involving species from Q as

$$R(Q) = \{k|\ k \in \{1, \dots, r_0\} \ \wedge\ \exists\, i \in Q:\ \nu_{ik} \neq \nu'_{ik}\}$$

$$R(Q) = \{k|\ \max(\nu_{ik}, \nu'_{ik}) \neq 0 \text{ for } i \in Q\}$$

Based on this we define a partition of $R(Q)$ into two subsets of reactions $R_S(Q) \subseteq R(Q)$ and $R_B(Q) \subseteq R(Q)$. $R_S(Q)$ includes those reactions that only change copy numbers of species in Q whereas $R_B(Q)$ includes those reactions that change copy numbers of species not in Q:

$$R_s(Q) = \left\{k|\ k \in R(Q)\ \wedge\ v_{jk} = \nu'_{jk} \forall j \notin Q\right\}$$

$$R_B(Q) = \left\{k|\ k \in R(Q)\ \wedge\ \exists\, j \notin Q:\ v_{jk} = \nu'_{jk}\right\}$$

$$R_s(Q) = \{k|\ \max(\nu_{ik}, \nu'_{ik}) \neq 0, \nu_{jk} = \nu'_{jk} \text{ for } i \in Q \text{ and } j \notin Q\}$$

$$R_B\left(Q\right) = \left\{k|\ \max\left(\nu_{ik},\nu'_{ik}\right)\neq 0, \nu_{jk}\neq\nu'_{jk} \text{ for } i\in Q \text{ and } j\notin Q\right\}$$

We assume that the SCRN defined by Q and $R\left(Q\right)$ has a stationary distribution given the current state of the surrounding network.

We define the timescale separation of the subnetwork given by the species Q as

$$\Delta\tau\left(Q\right) = \min_{k\in R_B}\left(\tau_B\left(k\right)\right) - \max_{k\in R_S}\left(\tau_S\left(k\right)\right)$$

where we define the logarithmic timescales $\tau_S\left(k\right)$ and $\tau_B\left(k\right)$ as

$$\tau\left(k\right) = -\frac{\log\ \kappa'_k}{\log\ N} - \frac{1}{\log\ N}\sum_{i,\nu_{ik}\neq 0}\begin{cases}\log\ x_i & x>0\\ 0 & x=0\end{cases} + \min_{\left\{i:\ \nu_{ik}\neq\nu'_{ik}\right\}}\left(\alpha_i\right)$$

$$\tau_S\left(k\right) = -\frac{\log\ \kappa'_k}{\log\ N} - \frac{1}{\log\ N}\sum_{i,\nu_{ik}\neq 0}\begin{cases}\log\ x_i & x>0\\ 0 & x=0\end{cases}$$

$$\tau_B\left(k\right) = \tau_S\left(k\right) + \min_{\left\{i:\ \nu_{ik}\neq\nu'_{ik}\right\}}\left(\alpha_i\right)$$

In words, $\Delta\tau\left(Q\right)$ is the minimum timescale of all reactions connecting the subnetwork to the outer network minus the maximum timescale of all reactions within the subnetwork. The second term ($\min_{\left\{i:\ \nu_{ik}\neq\nu'_{ik}\right\}}\left(\alpha_i\right)$) of $\tau_B\left(k\right)$ reflects the fact, that the relative change of a highly abundant species is smaller than the relative change of a low abundant species even if the reaction propensities are the same.

If $\Delta\tau\left(Q\right)\geq\zeta$ where the parameter $\zeta>0$ then we can average the subset of species Q, i.e. set their copy number to their stationary average. Of course the results for the species in Q can only be interpreted on the level of distributions and not on the level of single sample paths as those get lost in the process of averaging. Following we present three different strategies for averaging of fast subnetworks:

Finite Markov Chains Consider a subset of fast species Q fulfiling a conservation relation

$$\sum_{q_i\in Q} a_i q_i = const.$$

with $a_i\in\mathbb{Z}_{\geq 1}$. Such a subnetwork forms a Finite Markov Chain (citation). Consider that the subnetwork is also irreducible, i.e. every state of the Markov Chain can be reached from any other state in finite time. Without loss of generality we enumerate the states with $1,\ldots,L$. The irreducibility of this Markov Chain is equivalent to $\dim\left(\text{null}\left(A\right)\right) = 1$ where A is the generator matrix of the Markov Chain. In the irreducible case the stationary distribution $\pi_1, \cdots, \pi_L$ of the Markov Chain is an eigenvector of the generator matrix with eigenvalue 0:

$$A^T \cdot \begin{bmatrix} \pi_1 \\ \vdots \\ \pi_L \end{bmatrix} = 0$$

Thus we can compute $\pi_1, \cdots, \pi_L$ and perform averaging.

Pseudo-Linear subnetworks Consider a subset of fast species Q such that all the reactions consuming or producing species in Q have at most one reactant in Q. Such a subnetwork is called pseudo-linear as the additional reactants outside of the subnetwork can be considered constant for the timescale of the subnetwork. For linear SCRNs we can easily compute the stationary average by finding the stationary solution $\frac{d}{dt} x_i(t) = 0$ of the corresponding deterministic system

$$\frac{d}{dt} x_i(t) = N^{-\alpha_i} \sum_{k=1}^{r_0} \int_0^t \left(N^{\beta_k + \alpha \cdot \nu_k} \lambda_k^N \left(Z^N(s) \right) ds \left(\nu'_{ik} - \nu_{ik} \right) \right)$$

and thus we can perform averaging. Note: The stationary distribution on the other hand can't be easily computed for all linear SCRNs. We will elaborate on this in the next section.

Zero-Deficiency subnetworks Here we apply results from Anderson et al. [1] showing that a *weakly reversible* SCRN with *deficiency* 0 has a unique stationary distribution $\pi(x)$ which is a product-form:

$$\begin{cases} \pi(x) = \prod_{i=1}^{s_0} \frac{c_i^{x_i}}{x_i!} & \text{for reducible networks} \\ \pi(x) = \prod_{i=1}^{s_0} \frac{c_i^{x_i}}{x_i!} \exp(-c_i) & \text{for irreducible networks} \end{cases}$$

where $c \in \mathbb{R}_{\geq 0}^{s_0}$ is the equilibrium point of the corresponding deterministic system.

A SCRN is called *weakly reversible* if for every reaction k there is a sequence of reactions $\nu_{k_1} \rightarrow \nu'_{k_1} \rightarrow \cdots \rightarrow \nu_{k_m} \rightarrow \nu'_{k_m}$ such that $\nu'_k = \nu_{k_1}$ and $\nu_k = \nu'_{k_m}$. The *deficiency* of a SCRN is defined as $\delta = |C| - l - s$ where $|C|$ is the number of reaction complexes ($C = \{\nu_k\} \cap \{\nu'_k\}$), l is the number of linkage classes (a linkage class is a connected component of the reaction complex graph corresponding to the SCRN) and s is the dimension of the stochiometric subspace $S = span_{k \in \{1,\ldots,r_0\}} \{\nu'_k - \nu_k\}$. We refer to [1] for more details.

We will got into more details about the implementation in the following section.

3 IMPLEMENTATION

In this section we shortly summarize the well known scheme for simulating a PDMP model and then describe our contribution of an adaptive hybrid scheme for simulating SCRNs.

We use the following notation for a specific sample path:

t_0	initial time of simulation
t_1	final time of simulation
x_0	initial state of simulation
P	total number of reactions occuring from t_0 until t_1
$t_r(p)$	time of the pth reaction for $p = 1, \ldots, P$ and by definition $t_r(0) = t_0$
$M_S(t)$	the set of reactions marked as stochastic
$M_D(t)$	the set of reactions marked as deterministic

Vanilla PDMP

The typical algorithm to simulate PDMP models is to evolve the part of the model that is described deterministically until the next stochastic reaction occurs. Then the copy numbers are updated according to the reaction. This is repeated until the end-timepoint of the simulation is reached.

The time of the next stochastic reaction p is easily formulated as the root of an additional ODE state w_p defined as

$$\frac{d}{dt} w_p(t) = \sum_{k \in M_S} \lambda_k(x(t))$$

with the boundary condition of $w_p(t_r(p-1)) = \log(u_p)$ where u_p is a random variable uniformly extracted from the interval $[0, 1]$. Note that $\log(u_p) \leq 0$ and $\lambda_k(x) \geq 0$ and so starting from the time $t = t_r(p-1)$ the state $w_p(t)$ will monotonically increase until either $t = t_1$ if $p = P$ or if $p < P$ until the pth reaction occurs at $t = t_r(p+1)$. It follows from the random-time-change formulation that $w_p(t_r(p+1)) = 0$. Thus by searching for the root of $w_p(t)$ one can find the time of the pth reaction in the process of solving the system of ODEs [5]. This is called root finding or event detection and modern ODE solvers like CVODE support this [6].

The procedure described above is depicted in algorithm 1. Simulating a PDMP approximatino of a SCRN can be considerably faster if the reduction of the number of reactions is big enough. An improper PDMP approximation can also make the simulation much slower than simulating the exact model with SSA. This happens for example if there are still a lot of stochastic reactions happening and thus the ODE solver has to be started over and over again causing a performance hit because the ODE solving machinery has to be reinitialized after the state has changed from stochastic reactions.

Algorithm 1 Vanilla PDMP

```
$t \leftarrow t_0$, $x(t_0) \leftarrow x_0$, $p \leftarrow 1$
while $t < t_1$ do
    $u_p \leftarrow \sim \mathcal{U}[0,1]$
    $w_p(t) \leftarrow \log(u_p)$
    evolve
        $x$ according to $\frac{d}{dt}x(t) = \sum_{k \in M_D} \lambda_k(x(t))(\nu'_k - \nu_k)$
        $w_p$ according to $\frac{d}{dt}w_p(t) = \sum_{k \in M_S} \lambda_k(x(t))$
    until $t = t_1$ or $w_p(t) = 0$
    if $w_p(t) = 0$ then
        $r \leftarrow k$ with probability $p_k \propto \lambda_k(x(t))$ for $k \in \{M_S\}$
        $x(t) \leftarrow x(t) + \nu'_r - \nu_r$
        $p \leftarrow p + 1$
    end if
end while
```

Adaptive hybrid models for SCRNs

The vanilla PDMP algorithm is suitable for SCRNs where a PDMP approximation can be found that is both valid over the whole simulation time and that considerably reduces the number of reactions that have to be simulated. For SCRNs that show big variations in copy-numbers over time this is usually not possible. Our contribution is an extension of the vanilla PDMP algorithm to accommodate for varying scales of the underlying SCRN both in time and in copy numbers. The general idea is to define bounds for the copy numbers in a suitable manner and upon leaving these bounds a suitable the PDMP approximation of the SCRN is updated according to the current time- and copy-number-scales.

The general outline of our adaptive simulation scheme is shown in algorithm 2. In addition to the vanilla PDMP algorithm described above we observe the occurence of additional events, i.e. the crossing of copy number bounds. Of course we also have to perform this check once in a while in the case where only stochastic reactions occur. If these bounds are crossed we perform an adaptation procedure that will compute new values for the scaling parameters α_i and β_k, rescale the state-values and possibly perform averaging on suitable subnetworks. The copy number bounds are defined by parameters $\xi \geq 0$ and $\eta \geq 0$ (typical values are $\xi = 1, \eta = 0.5$). For a discrete species the upper bound is N_0^{ξ}. For a continuous species the lower and upper bounds are $N_0^{-\eta}$ and N_0^{η} respectively. Thus the parameter ξ influences when a species could be considered continuous and the parameter η influences the bound of rescaled copy numbers where no adaptation is performed.

The adaptation and averaging procedures are described in more detail in the following sections.

Algorithm 2 Adaptive PDMP

```
$t \leftarrow t_0$, $x(t_0) \leftarrow x_0$, $p \leftarrow 1$, $\mathit{flag} \leftarrow \mathrm{true}$
while $t < t_1$ do
    if $\mathit{flag} = \mathrm{true}$ then
        $u_p \leftarrow \sim \mathcal{U}[0,1]$
        $w_p(t) \leftarrow \log(u_p)$
        $\mathit{flag} \leftarrow \mathrm{false}$
    end if
    evolve
        $x$ according to $\frac{d}{dt}x(t) = \sum_{k=1}^{r_0} \lambda_k(x(t))(\nu'_k - \nu_k)$
        $w_p$ according to $\frac{d}{dt}w_p(t) = \sum_{k=1}^{r_0} \lambda_k(x(t))$
    until $t = t_1$ or $w_p(t) = 0$ or copy number bounds have been crossed
    if copy number bounds have been crossed then
        Perform adaptation procedure
    end if
    if $w_p(t) = 0$ then
        $r \leftarrow k$ with probability $p_k \propto \lambda_k(x(t))$
        $x(t) \leftarrow x(t) + \nu'_r - \nu_r$
        $p \leftarrow p + 1$
        $\mathit{flag} \leftarrow \mathrm{true}$
        if copy number bounds have been crossed then
            Perform adaptation procedure
        end if
    end if
end while
```

Computation of scaling parameters

Upon crossing the copy number bounds we want to compute a PDMP approximation based on the framework by Kang et al. described above. To this end we have to make sure that our approximation is well-behaved and at the same time we want to handle as many reactions in a deterministic way as possible given the current state.

We achieve this by formulating the *species balance conditions* as constraints of a linear program that maximizes a weighted sum of the α_i and β_k. As the α_i will decide whether a reaction term converges to a deterministic term we weigh the α_i with a factor λ that we usually set to 100.

Define

$$A_i = \frac{\log(x_i)}{\log(N_0)} + 1, \;\; i \in \{1, \ldots, s_0\}$$

$$B_k = \frac{log(\kappa'_k)}{\log(N_0)} + 1 \;\; k \in \{1, \ldots, r_0\}$$

We compute the α_i and β_k by solving the following linear program

$$\begin{array}{lcr} \text{maximize} & \lambda \sum_{i=1}^{s_0} \frac{\alpha_i}{A_i} + \sum_{i=1}^{r_0} \frac{\beta_k}{B_k} & \\ & & \\ \text{subject to} & 0 \leq \alpha_i \leq A_i & \text{for each } i \in \{1, \ldots, s_0\} \\ \text{and} & \beta_k \leq B_k & \text{for each } k \in \{1, \ldots, r_0\} \\ \text{and} & \alpha_i \geq \beta_k + \alpha \cdot \nu_k & \text{for each } i \in \{1, \ldots, s_0\}, \; k \in \{1, \ldots, r_0\} \end{array} \tag{1}$$

The algorithm for the adaptation procedure is shown in algorithm 3.

Algorithm 3 Adaptation

```
Compute scaling parameters $\alpha_i$ and $\beta_k$ by solving the linear program in 1
Update species and reaction types
Recompute copy number bounds
Perform averaging procedure
```

Averaging of fast subnetworks

After the computation of the scaling parameters in algorithm 3 we perform the averaging of fast subnetworks. The possible subnetworks suitable for averaging can be precomputed once at the start of the program (e.g. weakly reversible, zero-deficiency subnetworks). Upon averaging we compute the timescale-separation $\Delta\tau$ of all suitable subnetworks and select the subnetworks with $\Delta\tau \geq \zeta$. From these we perform a greedy strategy and repeatedly select the largest subnetwork that only contains species that haven't been selected yet. This gives us a list of disjunct subnetworks. Now we run through the list of subnetworks that were selected for averaging in the previous iteration but haven't been selected in the current iteration. For each of these subnetworks we sample the state from the stationary distribution. Finally we perform averaging by computing the stationary average of each selected subnetwork.

In the case of pseudo-linear subnetworks we simply take the rounded stationary average instead of sampling from the stationary distribution as we can't compute this for all pseudo-linear subnetworks. Similarly in the case of a reducible, weakly reversible, zero-deficiency subnetwork we approximate the stationary distribution with a multinomial distribution (scaled according to the conservation-relation) as it is very difficult to sample from the stationary distribution in the general case. We expect that these approximations won't do any harm though one should keep them in mind.

The algorithm for the averaging procedure is shown in algorithm 4.

Algorithm 4 Averaging

```
Once: Precompute subnetworks $L_A$ suitable for averaging (e.g. weakly reversible, zero-deficiency subnetworks)
Let $L_P$ be the set of previously averaged subnetworks
Set $L_C = \emptyset$
for all suitable subnetworks $Q \in L_A$ do
    if $\Delta\tau(Q) \geq 1$ then
        Set $L_C = L_C \cup Q$
    end if
end for
$L_F = \emptyset$
while $L_C \neq \emptyset$ do
    Set $Q = \text{argmax}_{W \in L_C}(|W|)$
    Set $L_C = L_C \setminus Q$
    if $\{s : s \in W, W \in L_F\} \cap Q = \emptyset$ then
        Set $L_F = L_F \cup Q$
    end if
end while
for all subnetwork $Q \in L_P \setminus L_F$ do
    Sample state for subnetwork $Q$ from the stationary distribution
end for
for all subnetwork $Q \in L_F$ do
    Compute stationary average of subnetwork $Q$
end for
```

4 NUMERICAL EXAMPLES

5 CONCLUSIONS

APPENDIX

References

[1] David F. Anderson, Gheorghe Craciun, and Thomas G. Kurtz. Product-form stationary distributions for deficiency zero chemical reaction networks. *Bulletin of mathematical biology*, 72(8):1947–1970, 2010.

[2] Mark HA Davis. Piecewise-deterministic Markov processes: A general class of non-diffusion stochastic models. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 353–388, 1984.

[3] Daniel T. Gillespie. Approximate accelerated stochastic simulation of chemically reacting systems. *The Journal of Chemical Physics*, 115:1716, 2001.

[4] Hye-Won Kang and Thomas G. Kurtz. Separation of time-scales and model reduction for stochastic reaction networks. *The Annals of Applied Probability*, 23(2):529–583, 2013.

[5] Panagiotis Kouretas, Konstantinos Koutroumpas, John Lygeros, and Zoi Lygerou. Stochastic hybrid modeling of biochemical processes. *Stochastic Hybrid Systems*, 24(9083), 2006.

[6] Radu Serban and Alan C. Hindmarsh. CVODES, the sensitivity-enabled ODE solver in SUNDIALS. In *Proceedings of the 5th International Conference on Multibody Systems, Nonlinear Dynamics and Control, Long Beach, CA*, 2005.