# A Comparison of the Magnus Expansion and Other Solvers for the Chemical Master Equation with Variable Rates


**Khanh Dinh and Roger Sidje**

**Abstract** Many traditional approaches for solving the chemical master equation (CME) cannot be used in their basic form when reaction rates change over time, for instance due to cell volume or temperature. One technique is to use the Magnus expansion to represent the solution to the CME as the action of a matrix exponential, for which Krylov-based approximation methods can be applied. In this paper, we compare two variants of the Magnus scheme with some popular ordinary differential equations (ODE) solvers, such as Adams-Bashforth, Runge-Kutta and Backward-differentiation formula (BDF). Our numerical tests show that the Magnus variants are remarkably efficient at computing the transient probability distributions of a transcriptional regulatory system where propensities vary over time due to cell volume increase.

**Keywords** Chemical master equation · Magnus expansion · Matrix exponential

## 1 Introduction

Consider a chemical reaction system consisting of $N$ molecular species $S_1, \ldots, S_N$ that interact through $M$ reactions. The *reaction rates* $c_k(t)$ are time-dependent scale factors for how likely the reaction $k$ occurs at time $t$. The *state vector* of the system is defined as

$$\boldsymbol{x}(t) = (x_1, \ldots, x_N)^T,$$

K. Dinh (✉) · R. Sidje
Department of Mathematics, University of Alabama, Tuscaloosa, USA
e-mail: kdinh@crimson.ua.edu

R. Sidje
e-mail: roger.b.sidje@ua.edu



D. M. Kilgour et al. (eds.), *Recent Advances in Mathematical and Statistical Methods*, Springer Proceedings in Mathematics & Statistics 259,
https://doi.org/10.1007/978-3-319-99719-3_24

where $x_l$ is the count for species $S_l$ at time $t$. The *propensity function* $\alpha_k(\boldsymbol{x}(t), t)$ of reaction $R_k$ at the current state $\boldsymbol{x}(t)$ and current time $t$ is defined so that the probability of such a reaction occurring during the infinitesimal time interval $[t, t + dt)$ is equal to $\alpha_k(\boldsymbol{x}(t), t)dt$. If the reaction occurs, the state vector is updated as $\boldsymbol{x}(t) := \boldsymbol{x}(t) + \boldsymbol{\nu}_k$, where the *stoichiometric vector* $\boldsymbol{\nu}_k$ stores the changes in species counts.

The chemical master equation (CME) [1] seeks $P(\boldsymbol{x}, t) = \text{Prob}\{\boldsymbol{x}(t) = \boldsymbol{x}\}$, the probability that the system is in state $\boldsymbol{x}$ at time $t$:

$$\frac{dP(\boldsymbol{x}, t)}{dt} = \sum_{k=1}^{M} \alpha_k(\boldsymbol{x} - \boldsymbol{\nu}_k, t) P(\boldsymbol{x} - \boldsymbol{\nu}_k, t) - \sum_{k=1}^{M} \alpha_k(\boldsymbol{x}, t) P(\boldsymbol{x}, t). \tag{1}$$

Let $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ be the ordered set of $n$ possible states, where $\boldsymbol{x}_i = (x_{1i}, \ldots, x_{Ni})^T$. We can rewrite (1) as a system of ordinary differential equations (ODEs) governing the change in $\boldsymbol{p}(t) = (P(\boldsymbol{x}_1, t), \ldots, P(\boldsymbol{x}_n, t))^T$ from the known initial distribution $\boldsymbol{p}_0$:

$$\begin{cases} \dot{\boldsymbol{p}}(t) = \boldsymbol{A}(t) \cdot \boldsymbol{p}(t), \\ \boldsymbol{p}(0) = \boldsymbol{p}_0, \end{cases} \tag{2}$$

where the transition rate matrix $\boldsymbol{A}(t) = [a_{ij}(t)] \in \mathbb{R}^{n \times n}$ is defined as

$$a_{ij} = \begin{cases} -\sum_{k=1}^{M} \alpha_k(\boldsymbol{x}_j, t), & \text{if } i = j, \\ \alpha_k(\boldsymbol{x}_j, t), & \text{if } \boldsymbol{x}_i = \boldsymbol{x}_j + \boldsymbol{\nu}_k, \\ 0, \text{ otherwise.} \end{cases}$$

Note that $\boldsymbol{A}$ changes over time due to the time-dependency of the reaction rates $c_k$.

The state space $\boldsymbol{X}$ can be infinite in theory, but is kept finite in practice, although $n$ can be very large. In this case, we can apply the finite state projection (FSP) [2], which reduces the state space to only the probable states during the time period of interest. The vectors $\boldsymbol{p}(t)$, $\boldsymbol{p}_0$ and matrix $\boldsymbol{A}(t)$ in (2) are then truncated to only values of this reduced finite state space. It is important to note that the CME is traditionally solved indirectly by drawing a large number of trajectories from Monte Carlo methods, such as the stochastic simulation algorithm (SSA) [3] or first reaction method (FRM) [4], and then computing the frequency at the desired time point. Their resulting error is statistical, in contrast to the analytical bound on the error when the CME is solved directly by employing the FSP. We only consider solving the CME directly in this study, and the results are compared against the frequencies from a large number of FRM trajectories.

We will discuss different approaches for solving the ODE system (2) in the next sections. For convenience, we will denote the ODE problem as

$$\dot{\boldsymbol{p}}(t) = f(t, \boldsymbol{p}(t)) \equiv \boldsymbol{A}(t) \cdot \boldsymbol{p}(t).$$

## 2 ODE Solvers

### 2.1 Adams

Adams methods form a family of linear multi-step methods, among which are explicit Adams-Bashforth and implicit Adams-Moulton. We use the **ADAMS-PECE** scheme by Shampine and Gordon [5], which implements the implicit Adams-Moulton according to

$$\begin{aligned} t_{k+1} &= t_k + h_k, \\ \boldsymbol{p}_{k+1} &= \boldsymbol{p}_k + h_k \left(\beta_r^{AM} \boldsymbol{f}_{k+1} + \beta_{r-1}^{AM} \boldsymbol{f}_k + \cdots + \beta_0^{AM} \boldsymbol{f}_{k-r+1}\right), \\ \boldsymbol{f}_{k+1} &= f(t_{k+1}, \boldsymbol{p}_{k+1}), \end{aligned}$$

where $\left\{\beta_i^{AM}\right\}_{i=0}^{r}$ are given analytically. The unknown $\boldsymbol{p}_{k+1} \approx \boldsymbol{p}(t_{k+1})$ is involved in both sides of the formula, leading to a nonlinear problem that is solved with a fixed-point scheme starting from the solution of the explicit Adams-Bashforth.

### 2.2 Runge-Kutta

Runge-Kutta methods form a class of multistage, one-step iteration ODE solvers. The explicit Runge-Kutta of order $r$ proceeds with

$$\begin{aligned} t_{k+1} &= t_k + h_k, \\ \boldsymbol{y}_i &= \boldsymbol{p}_k + h_k \textstyle\sum_{j=1}^{i-1} m_{ij}^{RK} f(t_k + h_k c_j^{RK}, \boldsymbol{y}_j); \quad i = 1, \ldots, r, \\ \boldsymbol{p}_{k+1} &= \boldsymbol{p}_k + h_k \textstyle\sum_{j=1}^{r} b_j^{RK} f\left(t_k + h_k c_j^{RK}, \boldsymbol{y}_j\right), \end{aligned}$$

in which the coefficients $\left\{m_{ij}^{RK}\right\}_{i,j=1}^{s}$, $\left\{b_i^{RK}\right\}_{i=1}^{s}$ and $\left\{c_i^{RK}\right\}_{i=1}^{s}$ are defined by the Butcher-tableau.

In this comparison, we use the solver **RK78** from the RKSUITE by Brankin et al. [6], which is a reputed Runge-Kutta method that controls the error and stepsize by using embedded Runge-Kutta formulae with orders 7 and 8.

### 2.3 Backward-Differentiation Formula

Backward-differentiation formula (BDF) methods are linear multi-step and follow the formula of order $r$:

$$t_{k+1} = t_k + h_k,$$
$$\boldsymbol{p}_{k+1} = h_k \beta_r^{BDF} f(t_{k+1}, \boldsymbol{p}_{k+1}) + \alpha_{r-1}^{BDF} \boldsymbol{f}_k + \cdots + \alpha_0^{BDF} \boldsymbol{f}_{k-r+1},$$
$$\boldsymbol{f}_{k+1} = f(t_{k+1}, \boldsymbol{p}_{k+1}),$$

where the coefficients $\{\alpha_i^{BDF}\}_{i=0}^{r-1}$ and $\beta_r^{BDF}$ are given analytically. The formula forms a nonlinear problem, because $\boldsymbol{p}_{k+1}$ appears on both sides.

We use the VODPK/BDF implementation [7] which has different options for solving the nonlinear problem:

1. **BDF-FI**: directly by functional iteration.
2. **BDF-GM-LU**: Newton root finding scheme; each linear system during the scheme is solved by SPIGMR (Scaled Preconditioned Incomplete GMRES), preconditioned by the LU decomposition.
3. **BDF-GM-LU0**: Newton root finding scheme; SPIGMR preconditioned by the incomplete LU decomposition, which discards elements not in the sparsity pattern of $\boldsymbol{A}$.
4. **BDF-LU**: each linear system during the Newton scheme is solved directly by LU decomposition.
5. **BDF-LU0**: the linear system is solved directly by incomplete LU decomposition.

## 3 The Magnus-Based Methods

### *3.1 Magnus Expansion*

The Magnus expansion [8] seeks to express the solution to (2) in the form of

$$\boldsymbol{p}(t) = \exp(\boldsymbol{\Omega}(t)) \cdot \boldsymbol{p}_0, \tag{3}$$

where $\boldsymbol{\Omega}(t)$ is an infinite series consisting of integrals and matrix commutators of $\boldsymbol{A}(t)$.

Originally a theoretical method in physics, there has been increasing interest to transform the Magnus expansion into a numerical solver for initial value problems (IVPs) in the form of (2). One such approach [9, 10], denoted **MAGNUS** in our comparative tests, truncates the Magnus series after 4 terms and approximates the integrals by the Gauss-Legendre quadrature, resulting in a fourth-order numerical scheme with constant stepsize $h$:

$$t_{k+1} = t_k + h,$$
$$\boldsymbol{A}_1 = \boldsymbol{A}\left(t_k + \left(\tfrac{1}{2} - \tfrac{\sqrt{3}}{6}\right)h\right), \quad \boldsymbol{A}_2 = \boldsymbol{A}\left(t_k + \left(\tfrac{1}{2} + \tfrac{\sqrt{3}}{6}\right)h\right),$$
$$\boldsymbol{\sigma} = \tfrac{h}{2}(\boldsymbol{A}_1 + \boldsymbol{A}_2) + \tfrac{h\sqrt{3}}{12}(\boldsymbol{A}_2\boldsymbol{A}_1 - \boldsymbol{A}_1\boldsymbol{A}_2),$$
$$\boldsymbol{p}_{k+1} = \exp(\boldsymbol{\sigma}) \cdot \boldsymbol{p}_k,$$

We approximate $\boldsymbol{p}_{k+1}$ by the matrix-free Krylov technique of EXPOKIT that only uses the action of $\boldsymbol{\sigma}$ on vectors. See [11] for more details, and also [12] for a comparison with traditional ODE solvers in the context where $\boldsymbol{A}$ is time-independent, i.e. constant, in (2).

### 3.2 Magnus with an Adaptive SSA-based State Space

During the integration time of any ODE solver for (2), most of the values in $\boldsymbol{p}(t)$ will be extremely small and therefore computing the full distribution can be expensive without gaining much accuracy. For CME problems with time-independent rates, the FSP-SSA method [13] reduces the state space $\boldsymbol{X}$ at each step to only states with potentially large probabilities during the small interval $[t_k, t_k + h]$. This is done by running SSA trajectories [3] from states in the current state space, and updating the state space to contain all states that the SSA trajectories travel through. The 'holes' in the state space are then patched by the $r$-step reachability [2], which seeks all states that can be connected to the state space with $r$ reactions or less, and expands the state space to include those. We incorporate this adaptive SSA-based state space expansion scheme into the **MAGNUS-SSA** method:

$$
\begin{gathered}
t_{k+1} = t_k + h, \\
\boldsymbol{X} \text{ is reduced to states with probability} > 10^{-16}, \\
\boldsymbol{X} \text{ is expanded by SSA over } [t_k, t_k + h] \text{ and } r\text{-step reachability with } r = 5, \\
\boldsymbol{A}_1 = \boldsymbol{A}\left(t_k + \left(\tfrac{1}{2} - \tfrac{\sqrt{3}}{6}\right)h\right), \quad \boldsymbol{A}_2 = \boldsymbol{A}\left(t_k + \left(\tfrac{1}{2} + \tfrac{\sqrt{3}}{6}\right)h\right), \\
\boldsymbol{\sigma} = \tfrac{h}{2}\left(\boldsymbol{A}_1 + \boldsymbol{A}_2\right) + \tfrac{h\sqrt{3}}{12}\left(\boldsymbol{A}_2\boldsymbol{A}_1 - \boldsymbol{A}_1\boldsymbol{A}_2\right), \\
\boldsymbol{p}_{k+1} = \exp(\boldsymbol{\sigma}) \cdot \boldsymbol{p}_k.
\end{gathered}
$$

Note that the SSA only serves here as a method for expanding the state space over the small time-stepping interval, and accounts for less than 5% of the computational runtime in the numerical tests. Computing $\boldsymbol{p}_{k+1}$ via EXPOKIT is the most time-consuming part of the algorithm. In this initial implementation, the stepsize $h$ is constant. We developed this method further in another work [14] to control the error and allow for adaptive stepsizes that can be either rejected or accepted.

## 4 Numerical Comparisons

### 4.1 The Alabama Supercomputer

All numerical tests reported here utilized resources of the Alabama Supercomputer, which houses two supercomputers called SGI UV and DMC. The user can request a job to be executed on either of them, or can simply let the operating system select

the more suitable system depending on the workload and availability. All codes were written in FORTRAN 77 and were run on the large queue of the SGI UV with 1 processor core (Xeon E5-4640 CPU operating at 2.4 GHz), 360 hr time limit and 120 GB memory limit.

## 4.2 The Transcriptional Regulatory Problem

The biological problem for comparing the ODE solvers depicts a transcriptional regulatory system [15]. The problem consists of six species:

$$\begin{aligned}
\text{M} &: \text{protein (monomer)},\\
\text{D} &: \text{transcription factor (dimer)},\\
\text{DNA} &: \text{DNA template, free of dimers},\\
\text{DNA.D} &: \text{DNA template, bound at one binding site},\\
\text{DNA.2D} &: \text{DNA template, bound at both binding sites},\\
\text{RNA} &: \text{mRNA produced by transcription},
\end{aligned}$$

which can interact through ten reactions:

$$\begin{array}{cccc}
\text{RNA} \xrightarrow{c_1} \text{RNA} + \text{M}; & \text{M} \xrightarrow{c_2} \varnothing; \\
\text{DNA.D} \xrightarrow{c_3} \text{RNA} + \text{DNA.D}; & \text{RNA} \xrightarrow{c_4} \varnothing; \\
\text{DNA} + \text{D} \xrightarrow{c_5} \text{DNA.D}; & \text{DNA.D} \xrightarrow{c_6} \text{DNA} + \text{D}; \\
\text{DNA.D} + \text{D} \xrightarrow{c_7} \text{DNA.2D}; & \text{DNA.2D} \xrightarrow{c_8} \text{DNA.D} + \text{D}; \\
\text{M} + \text{M} \xrightarrow{c_9} \text{D}; & \text{D} \xrightarrow{c_{10}} \text{M} + \text{M}.
\end{array}$$

The reaction rates are:

$$\begin{array}{llll}
c_1 = & 0.043 s^{-1}; & c_2 = & 0.0007 s^{-1}; \\
c_3 = & 0.078 s^{-1}; & c_4 = & 0.0039 s^{-1}; \\
c_5 = & \frac{0.012 \cdot 10^9}{A \cdot V(t)} s^{-1}; & c_6 = & 0.4791 s^{-1}; \\
c_7 = & \frac{0.00012 \cdot 10^9}{A \cdot V(t)} s^{-1}; & c_8 = & 0.8765 \cdot 10^{-11} s^{-1}; \\
c_9 = & \frac{0.05 \cdot 10^9}{A \cdot V(t)} s^{-1}; & c_{10} = & 0.5 s^{-1},
\end{array}$$

where $A$ is the Avogado's constant, and $V(t)$ is the cell volume at time $t$, which increases from the initial value $V(0) = 10^{-15}$ in accordance to

$$V(t) = V(0) e^{\ln(2) t/\tau}$$

during the entire cell cycle time period $\tau = 35$ minutes until the cell divides.

We wish to follow the distributions of the count of each species from the initial state where the cell has two dimers and the DNA is unbound:

$$\begin{aligned} \text{M} &= 0; \quad \text{D} &= 2; \\ \text{DNA} &= 1; \quad \text{DNA.D} &= 0; \\ \text{DNA.2D} &= 0; \quad \text{RNA} &= 0. \end{aligned}$$

The distributions from solving (2) by the ODE solvers are compared with the frequency from 100,000 FRM trajectories. The fixed FSP state space for the ODE solvers is found by finding the maximum and minimum of each species count during these trajectories, except for **MAGNUS-SSA**, which does not require a priori fixed FSP bounds and changes the state space adaptively instead. In practice, these bounds can be defined based on the knowledge of the biological problem or the experimental data, and may not require stochastic simulations.

As noted before, both **MAGNUS** and **MAGNUS-SSA** schemes were implemented here with constant stepsize, taken as $h = 1$ s in the reported numerical experiments. We also compared the actual execution times of the methods with that of the 100,000 trajectories of the FRM method (which obviously becomes more time-consuming as more trajectories are sampled, and is used here instead of the standard SSA because the reaction rates are time-dependent), and we observed that the FRM runtime was comparable to the Magnus-based methods.

## *4.3 Numerical Results*

We performed two numerical tests. The first test has small end time point and therefore results in a small state space, whereas the second one has longer end time point with a much larger state space and poses a large stiff problem for the ODE solvers. The error tolerance for all ODE solvers is $tol = 10^{-5}$.

### 4.3.1 Numerical Test 1

We seek the probability distribution at $t_f = 30s$. The FRM trajectories suggest the FSP bounds:

$$\begin{aligned} 0 \leq \quad \text{M} \quad &\leq 8; \; 0 \leq \quad \text{D} \quad \leq 3; \\ 0 \leq \quad \text{DNA} \quad &\leq 1; \; 0 \leq \text{DNA.D} \leq 1; \\ 0 \leq \text{DNA.2D} &\leq 1; \; 0 \leq \quad \text{RNA} \quad \leq 4. \end{aligned}$$

$\boldsymbol{X}$ has $n = 1440$ states and $\boldsymbol{A}$ has $nz = 8233$ nonzero elements.

The probability distributions at $t_f$ from all ODE solvers are displayed in Fig. 1. Their results agree with each other and fit the frequency from FRM.

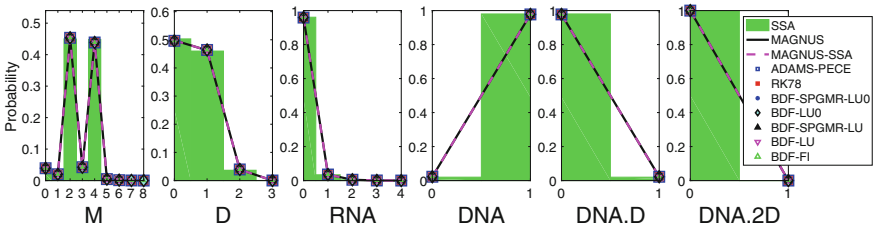


**Fig. 1** Probability distributions at $t_f = 30s$ from the ODE solvers in test 1

### 4.3.2 Numerical Test 2

We now attempt to find the probability distribution at $t_f = 10$ m. Because of the larger time range, the FSP bounds suggested by the FRM trajectories are more extensive:

$$\begin{array}{llll} 0 \le & \text{M} & \le 46; & 0 \le \text{D} \le 59; \\ 0 \le & \text{DNA} & \le 1; & 0 \le \text{DNA.D} \le 1; \\ 0 \le & \text{DNA.2D} & \le 1; & 0 \le \text{RNA} \le 12, \end{array}$$

resulting in $n = 293280$ states in $\boldsymbol{X}$ and $nz = 2091993$ nonzero elements in $\boldsymbol{A}$. The results from the ODE solvers are listed in Table 1.

Among the ODE solvers, ADAMS-PECE and RK78 did not finish, detecting that the problem was stiff. BDF-LU and BDF-FI also did not finish and reported that they are not appropriate solvers for the problem. BDF-SPGMR-LU failed before reaching $t_f$ because there was not sufficient storage, even though the work array was extended to the maximum size allowed on the Alabama Supercomputer.

The probability distributions from MAGNUS, MAGNUS-SSA and BDF-SPGMR-LU0, BDF-LU0 are compared in Fig. 2. While BDF-SPGMR-LU0 and BDF-LU0 produce wrong results, the distributions from MAGNUS and MAGNUS-SSA agree with the FRM frequencies. The Magnus-based methods are therefore the only reliable ODE solvers for this biological problem.

**Table 1** Reports from the ODE solvers in test 2

| ODE solver | Results |
|---|---|
| MAGNUS | Distributions at $t_f$ are in agreement with FRM frequencies |
| MAGNUS-SSA | Distributions at $t_f$ are in agreement with FRM frequencies |
| ADAMS-PECE | Fails before reaching $t_f$ (stiff problem detected - flag 5) |
| RK78 | Fails before reaching $t_f$ (stiff problem detected - flag 4) |
| BDF-SPGMR-LU0 | Distributions at $t_f$ do not fit the FRM frequencies |
| BDF-LU0 | Distributions at $t_f$ do not fit the FRM frequencies |
| BDF-SPGMR-LU | Reports that there is insufficient storage |
| BDF-LU | Reports that it is a wrong solver for this problem |
| BDF-FI | Reports that it is a wrong solver for this problem |

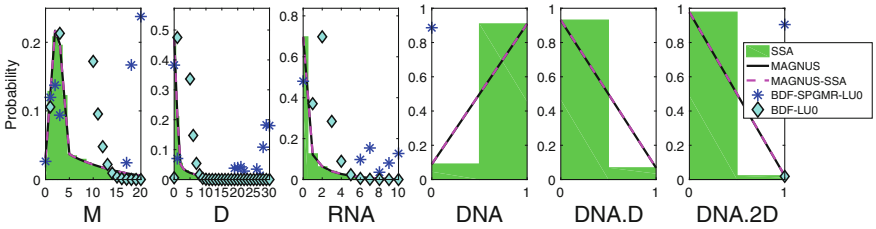


**Fig. 2** Probability distributions at $t_f = 10$ m from the ODE solvers in test 2

## 5 Conclusion

The ODE solvers in this comparison have been tested in [12] across problems in the form of (2) where $\boldsymbol{A}$ is time-independent, in which case the solution is $\boldsymbol{p}(t) = \exp(t\boldsymbol{A}) \cdot \boldsymbol{p}_0$. The authors showed that EXPOKIT [11] and BDF-LU0 [7] are the most efficient among the ODE solvers. We have considered solvers for the CME with time-dependent rates, with EXPOKIT embedded in the Magnus schemes. To our knowledge, such numerical comparisons of Magnus-based methods against other ODE solvers for large biological problems are just starting to appear in the literature.

That Adams, Runge-Kutta and BDF-FI solvers fail for $t_f = 10$ m is to be expected. The reaction rates in the transcriptional regulatory problem differ greatly in magnitude, suggesting that the ODE system is stiff. These ODE solvers behave like explicit methods and therefore are not suitable choices.

Among the remaining four BDF implementations, those relying on the complete LU decomposition are too expensive for this large problem. The incomplete LU0 decomposition, on the other hand, loses important information along the integration and therefore their solutions are unreliable.

The Magnus-based methods were the only solvers to successfully predict the probability distributions at $t_f = 10$ m, suggesting that they can be a powerful tool for solving stiff CME problems with time-dependent rates. Especially, the MAGNUS-SSA possesses the powerful advantage of flexibly changing the state space to follow the probability mass. It therefore does not demand the FSP bounds from the user, which are problem-dependent and require knowledge about the biological problem, and does not follow the entire probability distribution, which is expensive without offering meaningful accuracy. Disadvantages in current Magnus implementations, however, include the lack of an adaptive time-step scheme and the fact that the constant stepsize $h$ for the Magnus methods has to be chosen efficiently. We pursued adaptive time-stepping strategies in [14], which also contains more numerical comparisons.

## References


1. Gillespie, D.: A rigorous derivation of the chemical master equation. Phys. A **188**(1–3), 404–425 (1992)
2. Munsky, B., Khammash, M.: The finite state projection algorithm for the solution of the chemical master equation. J. Chem. Phys. **124**(4), 044104 (2006)
3. Gillespie, D.: Exact stochastic simulation of coupled chemical reactions. J. Phys. Chem. **81**(25), 2340–2361 (1977)
4. Purtan, R., Udrea, A.: A modified stochastic simulation algorithm for time-dependent intensity rates. In: 19th International Conference on Control Systems and Computer Science (2013)
5. Shampine, L.F., Gordon, M.K.: Computer solution of ordinary differential equations: the initial value problem. W.H. Freeman and Co. (1975)
6. Brankin, R.W., Gladwell, I., Shampine, L.F.: RKSUITE: a suite of Runge-Kutta codes for the initial value problem for ODEs, Softreport 91–1. Math. Dept., Southern Methodist University, Dallas, TX, USA, Technical report (1991)
7. Brown, P.N., Byrne, G.D., Hindmarsh, A.C.: VODE: a variable-coefficient ODE solver. SIAM J. Sci. Comput. **10**(5), 1038–1051 (1989)
8. Magnus, W.: On the exponential solution of differential equations for a linear operator. Comm. Pure Appl. Math. **7**(4), 649–673 (1954)
9. Iserles, A., Nørsett, S.P., Rasmussen, A.F.: Time symmetry and high-order Magnus methods. Appl. Numer. Math. **39**(3–4), 379–401 (2001)
10. MacNamara, S., Burrage, K.: Stochastic modeling of naive T cell homeostasis for competing clonotypes via the master equation. Multiscale Model. Simul. **8**(4), 1325–1347 (2010)
11. Sidje, R.B.: EXPOKIT: a software package for computing matrix exponentials. ACM Trans. Math. Softw. (TOMS) **24**(1), 130–156 (1998)
12. Sidje, R.B., Stewart, W.J.: A numerical study of large sparse matrix exponentials arising in Markov chains. Comput. Stat. Data Anal. **29**(3), 345–368 (1999)
13. Sidje, R.B., Vo, H.D.: Solving the chemical master equation by a fast adaptive finite state projection based on the stochastic simulation algorithm. Math. Biosci. **269**, 10–16 (2015)
14. Dinh, K.N., Sidje R.B.: An adaptive Magnus expansion method for solving the Chemical Master Equation with time-dependent propensities. J. Coupled Syst. Multiscale Dyn. https://doi.org/10.1166/jcsmd.2017.1124.
15. Goutsias, J.: Quasiequilibrium approximation of fast reaction kinetics in stochastic biochemical systems. J. Chem. Phys. **122**(18), 184102 (2005)