

# Improved Krylov-FSP Method for Solving the Chemical Master Equation

Huy D. Vo, Roger B. Sidje

**Abstract**—Model reduction techniques are needed to directly solve the chemical master equation (CME) due to its enormous size. We recently described an algorithm that solved the CME by combining the finite state projection, stochastic simulation algorithm and Krylov subspace approximations. In this work, we add further improvements that consist of an incomplete orthogonalization process with Krylov subspaces of variable dimension, and a refined strategy for monitoring the projection. We test these enhancements on difficult problems such as the MAPK cascade with 22 species.

**Index Terms**—stochastic gene regulation, chemical master equation, finite state projection, Krylov approximation.

## I. INTRODUCTION

Molecular interactions within a biological cell are inherently stochastic. To account for this stochasticity, the dynamics of the system is treated as a continuous-time, discrete-state Markov process, whose probability distribution is obtained by solving the chemical master equation (CME).

In recent years, there has been much interest in methods for directly approximating the probability distribution via the CME. The finite state projection (FSP) algorithm [1] was an early milestone on this direction, and many numerical methods for the CME have been proposed over the years [2], [3], [4], [5], [6], [7], [8], [9]. Direct CME solvers have been incorporated into a larger framework for selecting models and identifying parameters of gene regulatory networks [10], [11]. In a previous work [8], we described an adaptive Krylov-FSP [2] that was driven by the stochastic simulation algorithm (SSA). Our present study improves on that early method with several additions, namely incomplete orthogonalization, variable Krylov subspace dimension and a refined strategy for monitoring the projection.

Our paper is organized as follows. Section II summarizes the CME and sets up the terminology. Section III explains in details our adaptive Krylov-FSP solver. We report numerical examples solved with our method in section IV.

## II. BACKGROUND

### A. The chemical master equation

We consider a system of  $N \geq 1$  different chemical species that are interacting via  $M \geq 1$  chemical reactions in the cell, the state of the system is a vector  $\mathbf{x} \equiv (x_1, \dots, x_N)^T$  of nonnegative integers counting each species. The  $k$ th reaction ( $k = 1, \dots, M$ ) is associated with the stoichiometric vector  $\nu_k$ , and when such a reaction occurs in an infinitesimal interval  $[t, t+dt)$  with probability  $\alpha_k(\mathbf{x})dt$ , the state changes

from  $\mathbf{x}$  to  $\mathbf{x} + \nu_k$ . The state-dependent function  $\alpha_k(\mathbf{x})$  is called the *propensity function* of reaction  $k$ . The state space is a discrete subset of the lattice of  $N$ -tuple of nonnegative integers, allowing the states to be enumerated as  $\{\mathbf{x}_i\}_{i=1}^n$ , which is why we often refer to a state  $\mathbf{x}$  just by its index  $i \equiv i_{\mathbf{x}}$  in the enumeration. The chemical master equation (CME) can be formulated as a linear system of ordinary differential equations (ODEs)

$$\dot{\mathbf{p}}(t) = \mathbf{A}\mathbf{p}(t), \mathbf{p}(0) = \mathbf{p}_0, \quad (1)$$

where  $\mathbf{p} = (p_1, \dots, p_n)^T$  is a vector of probabilities indexed by the states, i.e.,  $p_i = \text{Prob}\{\mathbf{x}(t) = \mathbf{x}_i\}$ ;  $\mathbf{A} = [a_{ij}]$  is a sparse  $n \times n$  matrix defined by

$$a_{ij} = \begin{cases} \alpha_k(\mathbf{x}_j) & \text{if } \mathbf{x}_i = \mathbf{x}_j + \nu_k \\ 0 & \text{otherwise} \end{cases}. \quad (2)$$

### B. Finite state projection

Solving (1) amounts to evaluating the action of a matrix exponential, i.e., to compute the expression

$$\mathbf{p}(t) = \exp(t\mathbf{A})\mathbf{p}(0). \quad (3)$$

However, the size of the CME makes the computational cost intractable. The finite state projection [1] alleviates this problem by projecting the CME upon a finite set of states enumerated in  $J$ , and approximate the full solution by

$$\mathbf{p}(t) \approx \mathbf{p}_J(t) = \exp(t\mathbf{A}_J)\mathbf{p}_J(0), \quad (4)$$

where  $J$  is a subset of state indices, and

$$(\mathbf{A}_J)_{ij} = \begin{cases} a_{ij} & \text{if } i, j \in J \\ 0 & \text{if } i \notin J \text{ or } j \notin J \end{cases},$$

and for any  $\mathbf{v} = (v_1, \dots, v_n)^T$ ,  $\mathbf{v}_J$  is defined by

$$(\mathbf{v}_J)_i = \begin{cases} v_i & \text{if } i \in J \\ 0 & \text{otherwise} \end{cases}.$$

From there, all computations are done in practice on the effective truncations, which keep only the entries indexed by  $J$ , justifying why the FSP is a reduction method. The states indexed by  $J$  form a *finite state projection* (or projection for short). The truncation error is controlled by imposing the condition based on a bound derived in [1].

**Theorem II.1** (adapted from Theorem 2.2 in [1]). *Consider any Markov process in which the probability vector evolves according to the linear ODE (1). Let  $J$  be a subset of state indices, and  $\mathbf{p}_J$  the solution of (4). If for  $\varepsilon > 0$  and  $t_f \geq 0$  we have*

$$\mathbf{1}^T \mathbf{p}_J(t_f) \geq 1 - \varepsilon, \quad (5)$$

*then  $\mathbf{p}_J(t_f) \leq [\mathbf{p}(t_f)]_J \leq \mathbf{p}_J(t_f) + \varepsilon \mathbf{1}$ .*

Manuscript received July 19, 2016; revised August 8, 2016. This work was supported in part by NSF grant DMS-1320849.

H. D. Vo and R. B. Sidje are with Department of Mathematics, University of Alabama, Tuscaloosa, AL 35487, USA. Email: hvo@crimson.ua.edu, roger.b.sidje@ua.edu

### C. Krylov subspace approximation

A popular way to evaluate the action of the matrix exponential (3) is by the Krylov subspace approximation

$$\exp(\tau \mathbf{A})\mathbf{v} \approx \mathbf{V}_m \exp(\tau \mathbf{H}_m) \|\mathbf{v}\| \mathbf{e}_1, \quad m \ll n \quad (6)$$

where  $\|\cdot\|$  is the Euclidean norm and  $\mathbf{e}_1$  the first unit vector of length  $m$ . The matrices  $\mathbf{V}_m$  and  $\mathbf{H}_m$  are generated by the Arnoldi process. The advantage of the Krylov-based approach is that it reduces the computation of a massive matrix exponential into a small exponential of  $\mathbf{H}_m$  (with  $m \ll n$ ). Furthermore, it only requires matrix-vector products that can be kept in functional form. Our algorithm exploits these advantages to solve the CME efficiently.

### III. ADAPTIVE KRYLOV-FSP DRIVEN BY THE SSA

This section details our algorithm. The first two subsections review the fundamental features previously described in [8] and that are critical to understanding our approach. The remaining subsections describe the improvements developed in this work.

#### A. Overview of the method

We advance the solution through time steps  $t_0 := 0 < t_1 < \dots < t_K := t_f$  using the recurrence

$$\mathbf{p}(t_{k+1}) \approx \mathbf{p}_{k+1} := \exp(\tau_k \mathbf{A}_{J_k}) \mathbf{p}_k, \quad t_{k+1} := t_k + \tau_k, \quad (7)$$

where  $\tau_k$  is the stepsize and  $J_k$  the projection containing the most likely states the Markov process can occupy within the time interval  $[t_k, t_k + \tau_k]$ . Operand vectors are padded with 0 for consistency. The evaluation of (7) is done by the Krylov scheme (6), seeded with  $\tau := \tau_k$ ,  $\mathbf{A} := \mathbf{A}_{J_k}$  and  $\mathbf{v} := \mathbf{p}_k$ . In the original implementation, the Krylov dimension  $m$  was fixed, but this will be made adaptive in section III-E. We always tune the stepsize  $\tau_k$  and the projection  $J_k$  at each time step to control the Krylov subspace error and the FSP truncation error. We estimate the Krylov error as in Expokit [12] that we build upon. For the FSP error, we impose the criteria

$$\mathbb{1}^T \mathbf{p}_{k+1} \geq 1 - \varepsilon_{FSP} \frac{t_{k+1}}{t_f}, \quad t_{k+1} = t_k + \tau_k, \quad (8)$$

where  $\varepsilon_{FSP}$  is the tolerance for FSP-truncation error at time  $t_f$ . We could view (8) as controlling the accumulation of the FSP truncation error at each step so that the final error does not exceed the prescribed tolerance  $\varepsilon_{FSP}$ . If this condition is passed at step  $k$  with stepsize  $\tau_k$ , we move on to the next step with  $\mathbf{p}_{k+1}$  as the approximation to the probability distribution at time  $t_{k+1}$ . Otherwise, we reduce the stepsize repeatedly until the condition is met, then enlarge the projection before moving to the next step. Finally, there is an opportunity for further savings that we exploit at every step by dropping states that become unlikely. We shall make clear in the next subsections how to carry out these updates.

#### B. SSA-driven state space expansion

The FSP-like criteria (8) ensures that the projected system with projection  $J_k$  approximates the full CME with sufficient accuracy. Failure to meet this means that more states outside of the projection are becoming likely in the current time

interval, and we need to expand  $J_k$  to capture these states. We recap the way we expand the projection as given in [8]. Let  $J_0, \dots, J_k$  be a sequence of sets approximating the most likely reachable states at time  $t_0, \dots, t_k$ , respectively. Let  $\text{SSA}(\mathbf{x}_{t_k}, t_k, \tau_k)$  be the (random) path of a SSA run over  $[t_k, t_k + \tau_k]$  starting at a given state  $\mathbf{x}_{t_k}$ ,

$$\text{SSA}(\mathbf{x}_{t_k}, t_k, \tau_k) = \left\{ \mathbf{x}_{t_k + \sum s_i} \right\}_{t_k + \sum s_i \leq t_k + \tau_k}$$

Let  $\mathbb{J}$  be the set of all possible states, and let  $\mathbb{J}_k = J_0 \cup \dots \cup J_k$  be the states enumerated so far. Recall that  $i_{\mathbf{x}}$  is the index function that returns the index of the state  $\mathbf{x}$  from the enumeration in  $\mathbb{J}_k$  and  $\mathbf{x}_i$  its inverse (the correspondence between the two equivalent identifications of a state). Thus if  $\tilde{\mathbf{p}}(t_k)$  approximates the probability vector at time  $t_k$ , its  $i_{\mathbf{x}}$ -th component approximates the probability that the system is in state  $\mathbf{x}_i$  at time  $t_k$ . We approximate the next set of most likely reachable states using the three-stage scheme:

$$J_{k+\frac{1}{3}} = \{i \in J_k, \mu(\mathbf{x}_i) \geq \text{droptol}\} \quad (9a)$$

$$J_{k+\frac{2}{3}} = \bigcup_{i \in J_{k+\frac{1}{3}}} \left\{ \text{SSA}(\mathbf{x}_i, t_k, \tau_k) \right\} \quad (9b)$$

$$J_{k+1} = \mathcal{R}^r \left( J_{k+\frac{2}{3}} \right), \quad (9c)$$

where  $\mathcal{R}^r(J)$  is the  $r$ -step expansion of  $J$ , i.e., the set of states reachable from  $J$  in at most  $r$  reactions, which is formally defined by the recurrence:

$$\begin{aligned} \mathcal{R}^0(J) &= J, \\ \mathcal{R}(J) &= J \cup \bigcup_{k=1}^M \left\{ i \in \mathbb{J} : \mathbf{x}_i = \mathbf{x}_j + \boldsymbol{\nu}_k, j \in J \right\}, \\ \mathcal{R}^r(J) &= \mathcal{R}(\mathcal{R}^{r-1}(J)). \end{aligned}$$

The first stage (9a) is meant to drop states that have become improbable as measured by the function  $\mu$ . The next stage (9b) performs predictive SSA runs, each of which scouts a relatively cheap single path confined to a small interval of length  $\tau_k$ , the stepsize for which the Krylov approximation to the matrix exponential is expected to be numerically worthwhile. The last stage (9c) is meant to widen (or smooth) the path by  $r$ -step reachability. Note that the expansion is only called if the current projection  $J_k$  failed the FSP-like criteria (8).

The original implementation used a crude dropping strategy with  $\mu(\mathbf{x}) = (\mathbf{p}_k)_{i_{\mathbf{x}}}$  in (9a). We refine the strategy here.

#### C. Dropping states from the projection

The motivation for dropping states is that, since the probability distribution varies over time, states that are now probable may have low probabilities at the next step, and we can exploit this fact to improve efficiency by dropping states that become unlikely. The early version of our method [8] pruned states with probability below a prescribed threshold droptol. This may, however, remove states that have yet accumulated enough probabilities in the current time but soon turn out to be significant. Aggressively discarding these states increases the chance of failure at the following steps, forcing the algorithm to recover the just-dropped states. Therefore, we need to take into account the rate of probability flow

between states as done in [13]. Recall that the derivative of the FSP-projected solution at time  $t_k$  is given by

$$\dot{\mathbf{p}}_k = \mathbf{A}_{J_k} \mathbf{p}_k.$$

The positive entries of  $\dot{\mathbf{p}}_k$  tell us which states will receive more probability mass, and likewise, the negative entries indicate states that will lose probability mass. We only remove states with both small probabilities *and* small derivatives. Let  $\text{droptol}$  be the drop tolerance for the probability and  $\text{droptol}'$  the tolerance for the derivative, we remove states  $i$  in the projection that satisfy both  $(\mathbf{p}_k)_i < \text{droptol}$  and  $(\dot{\mathbf{p}}_k)_i < \text{droptol}'$ .

Finally, we mention that the value of  $\text{droptol}$  needs to be chosen small enough so that the entries of the truncated vector  $\mathbf{p}_k$  sum close to 1, as mandated by the post-check (8). More specifically, define the set of candidate entries to be dropped as

$$L = \{i : (\mathbf{p}_k)_i < \text{droptol}\},$$

we require that

$$\mathbf{1}^T \mathbf{p}_k - \sum_{i \in L} (\mathbf{p}_k)_i \geq 1 - \varepsilon_{FSP} \frac{t_k}{t_f}.$$

If this fails then the value  $\text{droptol}$  is too large and our code automatically reduce this threshold by a factor of 10 to avoid removing important states.

#### D. Incomplete orthogonalization

Our previous work [8] uses Arnoldi's full orthogonalization process (FOP) to generate the matrices  $\mathbf{V}_{m+1}$  and  $\bar{\mathbf{H}}_m$ . But FOP has the main drawback that the long recurrences in the modified Gram-Schmidt sweeps cost  $O(m^2n)$  and thus become prohibitive for large  $n$ . In this sequel we use an incomplete orthogonalization process (IOP) outlined in Alg. 1. It is an alternative procedure that orthogonalizes each basis vector  $\mathbf{v}_j$  in  $\mathbf{V}_{m+1}$  only to the preceding  $q$  ones  $\mathbf{v}_{\max(1, j-q)}, \dots, \mathbf{v}_{j-1}$  where  $1 \leq q \leq m$ . Setting  $q := m$  recovers the original Arnoldi process. IOP's cost of  $O(qmn)$  is much smaller than FOP's cost of  $O(m^2n)$  when  $n$  is large and  $q \ll m$  (we took  $q = 2$  in our experiments). With the matrices  $\mathbf{V}_{m+1}$  and  $\bar{\mathbf{H}}_m$  now formed by IOP, we still invoke (6) for the approximation of the matrix exponential, and the error estimation is done in the same way as already implemented for FOP in Expokit [12, Alg. 3.2].

---

#### Algorithm 1 IOP( $m, q$ )

---

**Input:** Matrix  $\mathbf{A}$  and vector  $\mathbf{v}$ , Krylov dimension  $m$ , orthogonalization length  $q$ .

**Output:** Krylov basis  $\mathbf{V}_m$  and banded matrix  $\mathbf{H}_m$ .

```

 $\mathbf{v}_1 := \mathbf{v} / \|\mathbf{v}\|_2$ 
for  $j := 1 : m$  do
   $\mathbf{p} := \mathbf{A} \mathbf{v}_j$ ;
  for  $i := \max(1, j - q + 1) : j$  do
     $h_{ij} := \mathbf{v}_i^T \mathbf{p}$ ;
     $\mathbf{p} := \mathbf{p} - h_{ij} \mathbf{v}_i$ ;
  end for
   $\mathbf{v}_{j+1} := \mathbf{p} / h_{j+1, j}$ ;
end for

```

---

Although FOP is preferable in solving linear systems and finding eigenvalues of general unsymmetric matrices

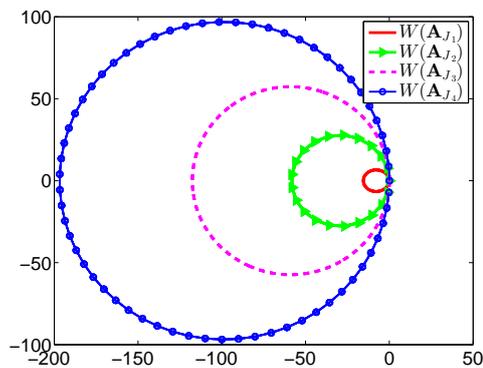


Fig. 1. The numerical range of the FSP matrices  $\mathbf{A}(J_k)$ ,  $k = 1, 2, 3, 4$  for a simple birth-death process with birth rate  $\gamma = 0.5$ , death rate  $\gamma = 0.2$ . The projections are  $J_1 = \{0, \dots, 5\} \cup \{25, \dots, 40\}$ ,  $J_2 = \{0, \dots, 150\}$ ,  $J_3 = \{100, \dots, 300\}$  and  $J_4 = \{0, \dots, 500\}$ . Larger numerical range typically implies slower convergence of the Krylov approximation. The numerical range is computed by the algorithm of Cowen and Harel (<https://www.math.iupui.edu/~ccowen/Downloads/33NumRange.html>).

due to its numerical robustness [14], the IOP yields better performance for our matrix exponential solver because the Krylov pairs  $\mathbf{V}_m$  and  $\mathbf{H}_m$  can be built and discarded quickly, even when  $m$  varies in a larger range, as the FSP projection  $J_k$  changes. The strategy to vary  $m$  adaptively is described next.

#### E. Adaptive Krylov dimension

The quality of the Krylov approximation to  $\exp(\tau \mathbf{A}_{J_k}) \mathbf{p}_k$  is closely related to the numerical range of  $\mathbf{A}_{J_k}$ . In particular, assuming that the numerical range is contained in a disk  $|z + \rho| < \rho$ , superlinear convergence in Krylov approximation is guaranteed for  $m \geq 2\rho\tau$  [15]. Changing the projection  $J_k$  alters the numerical range of  $\mathbf{A}_{J_k}$  at every step. Fig. 1 illustrates the effects of changing  $J$  on the numerical range  $W(\mathbf{A}_J)$  in a simple birth-death process. Keeping the same  $m$  across different projections may not be ideal and may cause the stepsize to become small to compensate the loss of accuracy induced by an inadequate  $m$  for a particular  $J_k$ , as occurred in the original Krylov-FSP [2] and our previous work [8], both of which used a fixed Krylov dimension  $m$ . This motivates adopting an adaptive strategy suggested by Niesen and Wright [16]. Let  $\text{Tol}$  be the desired tolerance for the approximation (6) and  $\tau_{k-1}, m_{k-1}, \tilde{\varepsilon}_k$  be respectively the stepsize, Krylov dimension, and estimated local error in successfully advancing from  $t_{k-1}$  to  $t_k$ . Following heuristics of ODE solvers, we seek the next stepsize  $\tau_k$  to advance from  $t_k$  to  $t_{k+1} = t_k + \tau_k$  in the suggested form

$$\tau_k^{\text{suggested}} := \gamma \tau_{k-1} (\omega_k)^{-1/s}, \quad (10)$$

where  $\omega_k$  is an error factor characterized by

$$\omega_k = \frac{\tilde{\varepsilon}_k}{\tau_{k-1} \cdot \text{Tol}}, \quad (11)$$

and  $\gamma$  is a safety factor (taken as 0.9 here), and  $s$  is the 'heuristic order' of the time-stepping scheme. As in [16], we let  $s = m/4$  for the first two suggestions, and if two consecutive stepsizes  $\tau_k^{(1)}$  and  $\tau_k^{(2)}$  suggested by (10) have been rejected in the same step with estimated errors  $\tilde{\varepsilon}_{k+1}^{(1)}$  and  $\tilde{\varepsilon}_{k+1}^{(2)}$  respectively, then we estimate the next suggestion

using

$$s \approx \frac{\log(\hat{\varepsilon}_{k+1}^{(2)}/\hat{\varepsilon}_{k+1}^{(1)})}{\log(\tau_k^{(2)}/\tau_k^{(1)})} - 1.$$

Alongside of this, we compute the prospective dimension

$$m_k^{suggested} := m_{k-1} + \frac{\log(\omega_k/\gamma)}{\log(\kappa)}, \quad (12)$$

where  $\omega_k$  is given by (11) as before. The value of  $\kappa$  is set to 2 for the first two suggestions (as in [16]). If there are two consecutive attempts in the same step using subspace dimensions  $m_k^{(1)}$  and  $m_k^{(2)}$  with estimated errors  $\hat{\varepsilon}_{k+1}^{(1)}$  and  $\hat{\varepsilon}_{k+1}^{(2)}$  respectively, then we use the estimate

$$\kappa \approx \left( \hat{\varepsilon}_{k+1}^{(1)} / \hat{\varepsilon}_{k+1}^{(2)} \right)^{1/(m_k^{(2)} - m_k^{(1)})}.$$

With candidates for stepsize and dimension at hand, we now weight their future costs. Advancing *one* step with a matrix  $\mathbf{A}_{J_k}$ , stepsize  $\tau$ , and Krylov dimension  $m$  using IOP with an orthogonalization of length  $q$  costs about

$$C_1(\tau, q, m, \mathbf{A}_{J_k}) = C_{IOP}(q, m, \mathbf{A}_{J_k}) + C_{Padé}(\tau, m) + C_{OVH},$$

where  $C_{IOP}(q, m, \mathbf{A}_{J_k})$  is the cost of the IOP (which includes the cost of the matrix-vector products with  $\mathbf{A}_{J_k}$ ),  $C_{Padé}(\tau, m)$  is the cost of the Padé technique for  $\exp(\tau \mathbf{H}_m)$  and  $C_{OVH}$  accounts for some overhead. We extrapolate from the cost of one step to estimating the future cost of integrating over the whole of  $[t_k, t_f]$  as  $C = \lceil \frac{t_f - t_k}{\tau} \rceil C_1$ . Now, if

$$C(\tau_k^{suggested}, q, m_{k-1}, \mathbf{A}_{J_k}) \leq C(\tau_{k-1}, q, m_k^{suggested}, \mathbf{A}_{J_k}),$$

meaning that the cost of varying the stepsize is smaller, then we choose to vary stepsize and retain the dimension (i.e. letting  $\tau_k := \tau_k^{suggested}$ ,  $m_k := m_{k-1}$ ). Otherwise, we retain the stepsize and vary the dimension (i.e., letting  $\tau_k := \tau_{k-1}$ ,  $m_k := m_k^{suggested}$ ).

#### IV. NUMERICAL EXAMPLES

We test our algorithm on a set of challenging CME problems taken from the literature. Our testing platform is a laptop running Ubuntu Linux, with 8 GB RAM and Intel Core i7 CPU. The Krylov tolerance is set at  $\text{Tol} := 10^{-8}$  for all three tests. The incomplete orthogonalization parameter is set to  $q := 2$ , and the Krylov dimension can vary dynamically from 10 to 100, except for the MAPK example where the maximum dimension is reduced to 30 due to memory constraint. Other algorithmic parameters are chosen differently for the specific problems explained below. We implemented our algorithm in FORTRAN, compiled by the GNU compiler.

##### A. Toggle switch

We revisit the toggle switch problem taken from [7]. There are two species interacting with each other through 4 reactions (Table I). We set the final time  $t_f = 100$ , FSP tolerance  $\varepsilon_{FSP} = 10^{-6}$ , drop tolerances starting at  $\text{droptol} = 10^{-10}$  and  $\text{droptol}' = 10^{-16}$ . Although [7] reports the need to use an FSP of  $2^{25}$  (over 30 million) states, our algorithm was able to solve the problem with a much smaller projection size that stays below 50 thousand states. This confirms the effectiveness of the SSA-driven scheme in

##### Algorithm 2 Adaptive Krylov-FSP.

**Input:** Initial projection  $J_0$ , final time  $t_f$ , stoichiometric vectors  $\nu_1, \dots, \nu_M$ , propensity functions  $\alpha_1, \dots, \alpha_M$ ; FSP tolerance  $\varepsilon_{FSP}$ , Krylov approximation tolerance  $\text{Tol}$ , drop tolerance values  $\text{droptol}$  and  $\text{droptol}'$ .

**Output:** The approximation  $\mathbf{p}_{FSP}(t_f)$  to the solution of the CME at time  $t_f$  within the FSP tolerance  $\varepsilon_{FSP}$ .

- 1: Initialize  $t := 0$ ,  $q := 2$ ,  $m := 10$  and the stepsize  $\tau$ .
- 2: Apply IOP( $m, q$ ) to generate  $\mathbf{V}_m$  and  $\mathbf{H}_m$ .
- 3: Estimate the Krylov error. If the error is above  $\tau \cdot \text{Tol}$ , adjust  $m$  and  $\tau$  and go back to step 2. Otherwise, compute the suggested values  $m_{suggested}$  and  $\tau_{suggested}$  for the next step.
- 4: Compute the tentative approximation

$$\mathbf{w} := \mathbf{V}_m \exp(\tau \mathbf{H}_m) \|\mathbf{v}\| \mathbf{e}_1$$

- 5: Check the FSP-like criteria

$$\mathbb{1}^T \mathbf{w} \geq 1 - \varepsilon_{FSP} \frac{t + \tau}{t_f}. \quad (13)$$

If fails, reduce  $\tau$  and go back to step 4 and set the flag  $iexpand := TRUE$ .

- 6: Set  $t := t + \tau$ . Set  $\mathbf{p}_{FSP}(t) := \mathbf{w}$ .
- 7: Prune states with probabilities below  $\text{droptol}$  and derivatives below  $\text{droptol}'$ . Reduce  $\text{droptol}$  if necessary.
- 8: If  $iexpand$  is  $TRUE$ , set  $\tau_{next} := \max(\tau, \tau_{suggested})$ . Use the SSA-guided procedure to expand the projection  $J$  to propagate the solution over the next time interval  $[t, t + \tau_{next}]$ .
- 9: If  $t < t_f$ , set  $\mathbf{v} := \mathbf{p}_{FSP}(t)$  and go back to step 2. Otherwise, export the approximation as well as the projection space.

tracking the most relevant states. Furthermore, our algorithm took only 200 seconds to solve the CME, as equilibrium was detected at  $t \approx 30$  by the *happy breakdown* feature of the Krylov approximation. Fig. 2 shows the history of problem size and accumulated CPU time for this example.

TABLE I  
REACTION CHANNELS OF THE TOGGLE SWITCH EXAMPLE. PARAMETERS AS IN [7], NAMELY  $\alpha_1 = 5000$ ,  $\alpha_2 = 1600$ ,  $\beta = 2.5$ ,  $\gamma = 1.5$ ,  $\delta_1 = \delta_2 = 1$ . ( $[X]$  IS THE NUMBER OF COPIES OF THE SPECIES X.)

	reaction	propensity
1.	$U \rightarrow \emptyset$	$\delta_1[U]$
2.	$\emptyset \rightarrow U$	$\frac{\alpha_1}{1+[V]^\beta}$
3.	$V \rightarrow \emptyset$	$\delta_2[V]$
4.	$\emptyset \rightarrow V$	$\frac{\alpha_2}{1+[U]^\gamma}$

##### B. Goutsias model

This test problem is adopted from a model of transcriptional regulation proposed by Goutsias [17] and is setup as in [18]. There are 6 chemical species  $M$ ,  $D$ ,  $RNA$ ,  $DNA$ ,  $DNA.D$ ,  $DNA.2D$ , that interact through 10 reaction channels (Table II), rate constants are set as in [18]. We integrate the CME for this model to  $t_f = 300s$  starting with

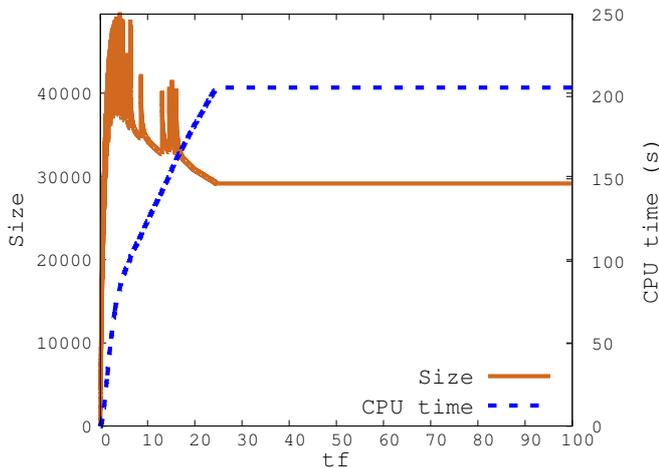


Fig. 2. Toggle switch. Projection size (solid) and accumulated CPU time (dotted) over the integration interval  $[0, 100]$ . Our algorithm detects equilibrium at  $t \approx 30$  and leapt straight to the end.

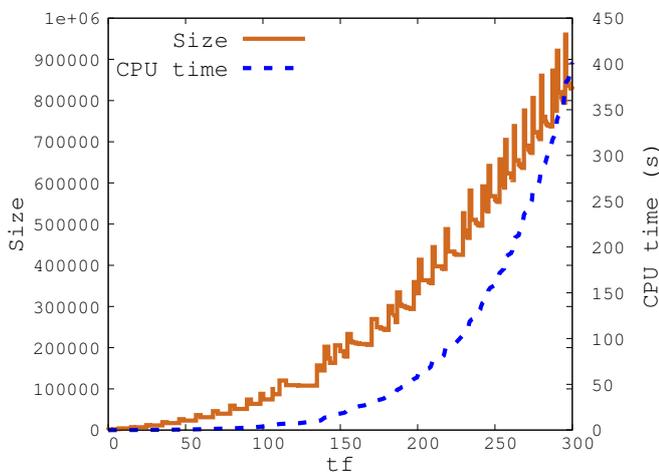


Fig. 3. Goutsias model. Projection size (solid) and accumulated CPU time (dotted) over the integration interval  $[0, 300]$ .

the initial state

$$\begin{aligned} \mathbf{x}_0 &= ([M], [D], [RNA], [DNA], [DNA.D], [DNA.2D]) \\ &= (2, 6, 0, 2, 0, 0). \end{aligned} \quad (14)$$

We set the FSP tolerance  $\varepsilon_{FSP} = 10^{-6}$ , drop tolerances starting from  $\text{droptol} = 10^{-10}$  and  $\text{droptol}' = 10^{-16}$ . To provide further validation for our result, we sampled 5 million trajectories using a direct implementation of the SSA in FORTRAN. From these we obtained the histogram for the marginal distribution of each species at  $t_f = 300$ . Fig. 4 shows the distribution of dimer approximated by SSA and our direct CME solver with good agreement. The same holds for all of the remaining species but we did not include the graphs here for brevity. Contrary to the toggle switch example, the support of the CME solution keeps growing in the Goutsias example. This provides a significant challenge to our approach. Nevertheless, our algorithm was able to finish in less than 400 seconds on the testing platform.

### C. Mitogen-activated Protein Kinase (MAPK) cascade

The MAPK cascade problem consists of 22 species and 30 reactions, originating from the chemical reaction network

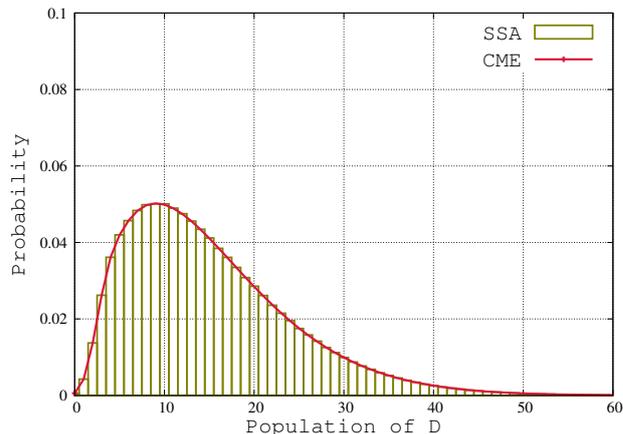


Fig. 4. Goutsias model. Marginal distribution of dimer at  $t_f = 300$  as computed by the SSA (bar) and our CME solver (line).

TABLE II  
REACTION CHANNELS IN GOUTSIAS MODEL OF REGULATED TRANSCRIPTION. THE PARAMETER  $A = 6.0221415 \times 10^{23}$  IS AVOGADRO'S NUMBER, AND  $V = 10^{-15}L$  IS THE SYSTEM VOLUME CHOSEN FOR THIS EXPERIMENT. ( $[X]$  IS THE NUMBER OF COPIES OF THE SPECIES  $X$ .)

reaction	propensity
1. $RNA \rightarrow RNA + M$	$c_1[RNA]$
2. $M \rightarrow \emptyset$	$c_2[M]$
3. $DNA.D \rightarrow RNA + DNA.D$	$c_3[DNA.D]$
4. $RNA \rightarrow \emptyset$	$c_4[RNA]$
5. $DNA + D \rightarrow DNA.D$	$c_5[DNA][D]$
6. $DNA.D \rightarrow DNA + D$	$c_6[DNA.D]$
7. $DNA.D + D \rightarrow DNA.2D$	$c_7[DNA.D][D]$
8. $DNA.2D \rightarrow DNA.D + D$	$c_8[DNA.2D]$
9. $M + M \rightarrow D$	$\frac{c_9}{2}[M]([M] - 1)$
10. $D \rightarrow M + M$	$c_{10}[D]$

studied in [19]. The CME for this system was set up and studied before in [20] with the total quasi-steady state assumption and we refer to that paper for the set of reactions and parameters. We revisit it here using our algorithm with  $\varepsilon_{FSP} = 10^{-1}$ , final time  $t_f = 10$ . We start with the seven key species  $[E_1], [E_2], [KKP'ase], [KP'ase], [KKK], [KK],$  and  $[K]$  set to 50 and other species at zero. Our method finished in less than 45 minutes. We plot the marginal distribution of the first four species in the cascade in Fig. 5. This is the first time the MAPK cascade problem is solved in this challenging settings by a direct CME approach without using advanced knowledge about the model as done in [20].

## V. CONCLUSION

We have improved our SSA-driven Krylov approximation scheme with an incomplete orthogonalization process, a strategy to vary the dimension of Krylov subspaces and a refined logic for dropping states that become unlikely throughout the integration. We tested our algorithm on increasingly difficult CME problems, namely the toggle switch example with two species, the Goutsias model with six species and the MAPK cascade with twenty two species.

## REFERENCES

- [1] B. Munsky and M. Khammash. The finite state projection algorithm for the solution of the chemical master equation. *J. Chem. Phys.*, 124(4):044104, 2006.

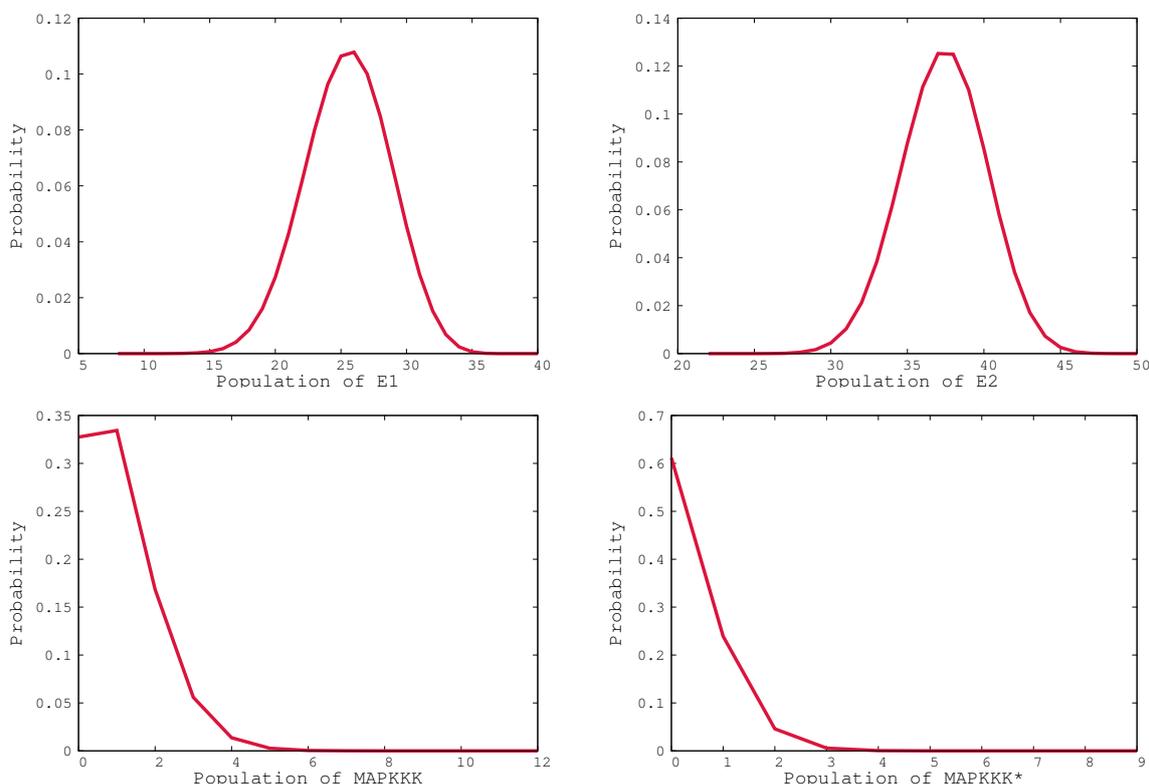


Fig. 5. MAPK example. Marginal distributions of the first four species in the cascade at time  $t_f = 10$ .

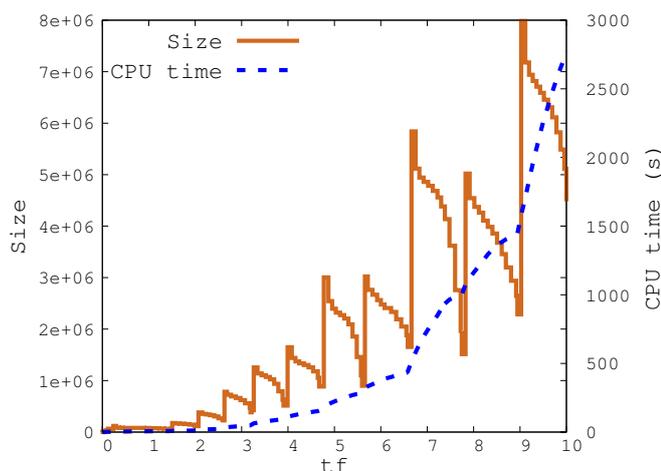


Fig. 6. MAPK example. Projection size (solid) and accumulated CPU time (dotted) over the integration interval  $[0, 10]$ .

[2] K. Burrage, M. Hegland, S. MacNamara, and R.B. Sidje. A Krylov-based finite state projection algorithm for solving the chemical master equation arising in the discrete modelling of biological systems. In A.N. Langville and W.J. Stewart, editors, *150<sup>th</sup> Markov Anniversary Meeting, Charleston, SC, USA*, pages 21–38. Bosc Books, 2006.

[3] B. Munsky and M. Khammash. A multiple time interval finite state projection algorithm for the solution to the chemical master equation. *J. Comp. Phys.*, 226(1):818–835, 2007.

[4] T. Jahnke and T. Udrescu. Solving chemical master equations by adaptive wavelet compression. *J. Comp. Phys.*, 229(16):5724–5741, 2010.

[5] V. Wolf, R. Goel, M. Mateescu, and T.A. Henzinger. Solving the chemical master equation using sliding windows. *BMC Sys. Bio.*, 4(1), 2010.

[6] S. Dolgov and B. Khoromskij. Simultaneous state-time approximation of the chemical master equation using tensor product formats. *Numer. Linear Algebra Appl.*, 22:197–219, 2013.

[7] V. Kazeev, M. Khammash, M. Nip, and C. Schwab. Direct solution

of the chemical master equation using quantized tensor trains. *PLoS Comp. Bio.*, 10(3), 2014.

[8] R.B. Sidje and H.D. Vo. Solving the chemical master equation by a fast adaptive finite state projection based on the stochastic simulation algorithm. *Math. Biosci.*, 269:10–16, 2015.

[9] Y. Cao, A. Terebus, and J. Liang. State Space Truncation with Quantified Errors for Accurate Solutions to Discrete Chemical Master Equation. *Bulletin Math. Biol.*, 78(4):617–661, 2016.

[10] G. Neuert, B. Munsky, R. Z. Tan, L. Teytelman, M. Khammash, and A. V. Oudenaarden. Systematic identification of signal-activated stochastic gene regulation. *Science*, 339(6119):584–587, 2013.

[11] B. Munsky, Z. Fox, and G. Neuert. Integrating single-molecule experiments and discrete stochastic models to understand heterogeneous gene transcription dynamics. *Methods*, 85:12–21, 2015.

[12] R.B. Sidje. Expokit: A software package for computing matrix exponentials. *ACM Trans. Math. Softw.*, 24(1):130–156, 1998.

[13] S. Ramaswamy, R. Lakerveld, P.I. Barton, and G. Stephanopoulos. Controlled formation of nanostructures with desired geometries: Part 3. Dynamic modeling and simulation of directed self-assembly of nanoparticles through adaptive finite state projection. *Ind. Eng. Chem. Res.*, 54(16):4371–4384, 2015.

[14] Z. Jia. On IOM(q): The incomplete orthogonalization method for large unsymmetric linear systems. *Numer. Linear Algebra Appl.*, 3(6):491–512, 1996.

[15] M. Hochbruck and C. Lubich. on Krylov Subspace Approximations To the Matrix Exponential Operator. *SIAM J. Numer. Anal.*, 34(5):1–16, 1997.

[16] J. Niesen and W.M. Wright. Algorithm 919: A Krylov subspace algorithm for evaluating the  $\phi$ -functions appearing in exponential integrators. *ACM Trans. Math. Softw.*, 38(3):22:1–22:19, 2012.

[17] J. Goutsias. Quasiequilibrium approximation of fast reaction kinetics in stochastic biochemical systems. *J. Chem. Phys.*, 122(18):184102, 2005.

[18] S. Macnamara, K. Burrage, and R.B. Sidje. Multiscale Modeling of Chemical Kinetics via Master Equation. *Multiscale Model. Simul.*, 6(4):1146–1168, 2008.

[19] C.Y. Huang and J. E. Ferrell Jr. Ultrasensitivity in the mitogen-activated protein kinase cascade. *Proc. Natl. Acad. Sci. USA*, 93(19):10078–10083, 1996.

[20] S. MacNamara, A.M. Bersani, K. Burrage, and R.B. Sidje. Stochastic chemical kinetics and the total quasi-steady-state assumption: Application to the stochastic simulation algorithm and chemical master equation. *J. Chem. Phys.*, 129(9), 2008.