Model Checking Markov Chains using Krylov Subspace Methods: An Experience Report*

Falko Dulat, Joost-Pieter Katoen, and Viet Yen Nguyen

Software Modeling and Verification Group
RWTH Aachen University, Germany
{ katoen, nguyen }@cs.rwth-aachen.de

Abstract. The predominant technique for computing the transient distribution of a Continuous Time Markov Chain (CTMC) exploits uniformization, which is known to be stable and efficient for non-stiff to mildly-stiff CTMCs. On stiff CTMCs however, uniformization suffers from severe performance degradation. In this paper, we report on our observations and analysis of an alternative technique using Krylov subspaces. We implemented a Krylov-based extension to MRMC (Markov Reward Model Checker) and conducted extensive experiments on five case studies from different application domains. The results reveal that the Krylov-based technique is an order of magnitude faster on stiff CTMCs.

1 Introduction

Stiff CTMCs are found in many domains, among which systems biology, where the reaction rates of molecules may vary greatly, and mission critical systems engineering, where failures occur frequently (like sensor glitches) or sporadically (like complete sensor failure). The transient distribution of CTMCs —what is the probability to be in a state at time t?— is a prominent measure of interest, and is fundamental to a range of measures of interest such as time-bounded reachability properties [2]. Its computation is a well-studied topic and a survey of applicable techniques is discussed by De Souza e Silva and Gail [7]. One wide-spread method is Jensen's uniformization [16] which is known for its good numerical stability and is implemented as the default method for transient analysis in various — if not all— Markov analysis tools. Its performance degrades however on stiff models, which, given its many definitions in literature, we simply refer to as the degree of difference between the smallest and largest rates in the CTMC. Other methods like Runge-Kutta solvers require small discretization values on stiff models, thereby suffering from similar performance problems. On top of these problems, potential numerical instability, not uncommon with stiff models, needs to be dealt with as well.

In this paper we reintroduce a Krylov-based method for computing the transient of a CTMC. It is briefly mentioned in Moler and Van Loan's discourse [21] on 19 methods for the matrix exponential as a novel 20^{th} method and in

* Funded by ESA/ESTEC under Contract No. 21171/07/NL/JD

De Souza e Silva and Gail's survey [7] as a possible method for computing the transient of CTMC. Despite these references and their success for many matrix-related computations in different fields of science and engineering, Krylov-based methods received scant attention in the field of probabilistic analysis. We believe this is due to three reasons, namely (i) to our knowledge, experiments with a Krylov-based method have been only conducted on small academic examples [25] or without regard to stiffness versus non-stiffness (ii) due to the lack of the former, nobody has identified the class of CTMCs for which Krylov-based methods excel and (iii) the good applicability of Krylov-based methods to the transient have, to our knowledge, not been explained theoretically. This report addresses, among things, these issues:

1. We apply a Krylov-based method for computing the transient distribution of CTMCs to model check time-bounded reachability properties expressed in Continuous Stochastic Logic (CSL) [2] (see Section 4.1).
2. We extensively compare the implemented Krylov-based method to the existing uniformization-based method on five case studies from the literature comprising various application domains (see Section 4.4).
3. We identify that computing the transient distribution is (much) faster with Krylov-subspace methods for a particular class of models, namely stiff CTMCs (see Section 4.4).
4. We provide an explanation of the good approximation properties of the Krylov-based matrix exponential using Schwerdtfeger's formula [22] (see Section 3.2).

The overall aim of this paper is to reintroduce Krylov-based methods to the probabilistic community as the preferable method for analyzing *stiff* CTMCs and to substantiate this by means of an experience report.

Organization of the paper. Section 2 summarizes CTMCs, uniformization, and its usage for CSL model checking. Section 3 provides the basics of Krylov subspace techniques, introduces Schwerdtfeger's formula, and characterizes an error estimate. Section 4 describes our experimental set-up, the five case studies, and provides all our results. Section 5 discusses some related work, whereas Section 6 provides a discussion and pointers to future work. Section 7 concludes the paper.

2 CSL Model Checking

This section introduces the basic concepts of model checking CTMCs using Continuous Stochastic Logic (CSL). It is only used as a stepping stone towards the remainder part of the paper. We refer to [2] for an elaborate treatment on this topic.

A labeled CTMC is a tuple $(S, \mathbf{Q}, L)$ where S is a finite set of states, $L : S \to 2^{AP}$ is a labeling function and $\mathbf{Q} : S \times S \to \mathbb{R}$ is a generator matrix. Each diagonal element $q_{s,s} \in \mathbf{Q}$ is defined as $q_{s,s} = -\sum_{s' \in S, s \neq s'} q_{s,s'}$, and all remaining elements $q_{s,s'}$ have a rate ≥ 0. Intuitively, a transition from s to s'

(with $s \neq s'$) is triggered within t time units by probability $1 - e^{-q_{s,s'}t}$. In other words, the occurrence of a transition is exponentially distributed. The rate of staying in a state s is described by the diagonal elements, namely $|q_{s,s}|$.

The *transient* distribution, which is further referred to in this paper as the transient, of a CTMC, denoted by $\pi(t)$, is the vector of probabilities being in states $s \in S$ at a time t given an initial distribution $\pi(0)$. It is characterized by Kolmogorov's forward differential equation $\frac{d}{dt}\pi(t) = \mathbf{Q} \cdot \pi(t)$, whose solution, given an initial distribution $\pi(0)$, is the following:

$$\pi(t) = e^{\mathbf{Q}t} \cdot \pi(0) \tag{1}$$

There are numerous numerical techniques to compute $\pi(t)$, of which Jensen's uniformization algorithm [16] is widely used.

Uniformization considers a uniformization rate $\Lambda \geq max_{i \in S}|q_{i,i}|$ so that the generator matrix can be rewritten as $\mathbf{Q} = \Lambda \cdot (\mathbf{P} - I)$. The matrix $\mathbf{P}$ is a stochastic matrix of the *uniformized CTMC*, and I is the identity matrix. When this rewritten $\mathbf{Q}$ is substituted in Equation (1), we get $\pi(t) = e^{\Lambda(\mathbf{P}-I)t} \cdot \pi(0)$. This equation can be rewritten and the matrix exponential can be expanded according to the Taylor-MacLaurin series, after which one gets:

$$\pi(t) = (\sum_{n=0}^{\infty} e^{-\Lambda t} \frac{(\Lambda t)^n}{n!} \mathbf{P}^n) \cdot \pi(0) \tag{2}$$

The part $\sum_{n=0}^{\infty} e^{-\Lambda t} \frac{(\Lambda t)^n}{n!}$ is the Poisson density function and it converges to 1. A numerically stable technique for computing it is by Fox-Glynn's method [9]. When an error bound $\epsilon > 0$ is given, the sum of Equation (2) can be truncated. The error bound ϵ can be used to determine the left- and right series truncation points $\mathcal{L}_\epsilon$ and $\mathcal{R}_\epsilon$, such that $\sum_{\mathcal{L}_\epsilon}^{\mathcal{R}_\epsilon} e^{-\Lambda t} \frac{(\Lambda t)^n}{n!} \geq 1 - \epsilon$. The left and right truncation points tend to be in the order of $O(\Lambda t)$. Large Λ's are common for stiff CTMCs and if this is also combined with a large t, the number of terms needed by uniformization to compute the transient is large.

The transient is fundamental to analyze labeled CTMCs with properties expressed in CSL, which describes a measure of interest in terms of satisfiable states and paths. It is also at the heart of more recent verification techniques that check a CTMC against a timed automaton specification [6]. For the scope of this paper, the interesting CSL properties are of the form $\mathcal{P}_{\bowtie p}(\Diamond^{[t_1,t_2]} \Psi)$. Intuitively, it means that the set of paths that eventually reach a state satisfying Ψ has a probability measure meeting $\bowtie p$ (where $\bowtie \in \{<, >, \leq, \geq, =\}$) within the real-valued timebounds t_1 to t_2. Ψ is a CSL formula (in all our examples a boolean expression) over the set of atomic propositions AP used in the labeled CTMC. To evaluate these kind of CSL properties, one computes the transient on a modified labeled CTMC(s) and compares the transient probabilities with the bound $\bowtie p$.

3 Krylov Subspace Methods

In the remainder of this paper, we use $A = \mathbf{Q} \cdot t$ and $v = \pi(0)$, to keep the notation similar to the literature of Krylov-subspace methods [23] while maintaining a connection to the matrix exponential in Equation (1).

The principal idea of Krylov subspace methods is to approximate the original sparse matrix A by a matrix H_m of much smaller dimension m. This works because H_m preserves an important property of A: its extreme eigenvalues. We will show using Schwerdtfeger's formula [14] that due to the extreme eigenvalue preservation, $e^A v$ can be effectively approximated by operations on H_m.

3.1 Mathematical Formulation

A naive approach for computing $e^A v$ is by using the Taylor-MacLaurin series expansion:

$$e^A v = \sum_{i=0}^{\infty} \frac{A^i}{i!} v = Iv + Av + \frac{1}{2} A^2 v + \dots$$

The matrix powers make it evident that this approach is highly numerically unstable. Fortunately, numerous stable techniques have been developed by the numerical linear algebra community. A powerful technique central in this paper, Krylov-based methods, exploits the sparseness of the matrix. This property typically holds for infinitesimal generators. Several researchers [23, 13] have developed and studied Krylov-based methods to the matrix exponential, where the principal idea is to approximate $e^A v$ by an element in the m-order Krylov subspace, defined as

$$K_m(A, v) = span\{v, Av, A^2 v, \dots, A^{m-1} v\}$$

where *span* denotes the usual linear span of a set of vectors. The precision of the approximation is controlled by the natural m. A lower m leads to a coarser approximation while a higher m increases precision at the expense of increased memory and computation time.

The approximation to $e^A v$ starts with the Arnoldi iteration, shown in Figure 1. In this figure, the dot product of two vectors w and v_i is denoted as (w, v_i) and the Euclidean norm of a vector w is denoted as $||w||_2$. The iteration produces a sequence of orthonormal Arnoldi vectors v_1 through v_m, which as a matrix V_m forms the orthonormal basis of the Krylov subspace K_m. It also produces the matrix H_m from the coefficients $h_{i,j}$. That matrix is the linear projection of A onto subspace K_m and is of upper Hessenberg

$v_1 = v/||v||_2$
for $j = 1, 2, \dots, m$ **do**
  $w = Av_j$
  for $i = 1, 2, \dots, j$ **do**
    $h_{i,j} = (w, v_i)$
    $w = w - h_{i,j} v_i$
  end for
  $h_{j+1,j} = ||w||_2$
  $v_{j+1} = w/h_{j+1,j}$
end for

Fig. 1. Arnoldi iteration

form, i.e. H_m is nearly triangular due to the non-zero entries in first subdiagonal. Thus what the Arnoldi iteration does is a Hessenberg decomposition of A, resulting in the following relation:

$$A \approx V_m H_m V_m^T$$

From this decomposition, the computation of $e^A v$ can be derived by operations on the smaller H_m. In this derivation, we use e_n, which is the n^{th} vector of I:

$$\begin{aligned}
e^A &\approx e^{V_m H_m V_m^T} && \textit{(by application of exponential)} \\
e^A &\approx I + V_m H_m V_m^T + \frac{1}{2}(V_m H_m V_m^T)^2 + \ldots && \textit{(by series expansion)} \\
e^A &\approx V_m (I + H_m + \frac{1}{2} H_m^2 + \ldots) V_m^T && \textit{(by } I = V_m^T V_m = V_m V_m^T\textit{)} \\
e^A &\approx V_m e^{H_m} V_m^T && \textit{(by series de-expansion)} \\
e^A V_m &\approx V_m e^{H_m} && \textit{(by multiplication with } V_m\textit{)} \\
e^A v_1 &\approx V_m e^{H_m} e_1 && \textit{(by } v_1 = V_m e_1\textit{)} \\
e^A v &\approx V_m e^{H_m} e_1 ||v||_2 && \textit{(by } v_1 = v/||v||_2\textit{)}
\end{aligned}$$

The last equation means that one can approximate the exponential over matrix A by computing the exponential over the much smaller H_m using stable dense methods (like Padé approximation) and project the result back to the original space using matrix V_m.

There are several advantages to this approach. First, the method can be performed iteratively. If the precision does not suffice for a particular subspace dimension m, this can be increased and the Arnoldi iteration can resume with the existing matrices V_m and H_m and iteratively extend them until a satisfactory precision has been reached. The second advantage is the numerical robustness. During the Arnoldi iteration, only multiplication, addition, division and subtraction is performed on normalized vectors. The exponential over H_m is stable when Padé approximation is combined with scaling and squaring, as was established for example by Ward [28].

3.2 Schwerdtfeger's Formula

The approximation of $e^A v$ by $V_m e^{H_m} e_1 ||v||_2$ works particularly well, despite H_m being of much lower dimension than A. Attempts to explain this behavior have led to advances in determining stricter error bounds [23, 13]. Instead of taking that direction, we shall explain it by an analysis in terms of eigenvalues of A and H_m.

A great deal of study has been conducted in relation of the eigenvalues of H_m to those of A. It is now well-accepted that H_m's eigenvalues, referred to as Ritz values, strongly correspond to the extreme eigenvalues of A [26]. Those are

the eigenvalues near the edge of A's spectrum. We will show that those are the eigenvalues of interest for the matrix exponential.

Any analytical function over a matrix A, like the exponential, can also be described in terms of the eigenvalues of A. Several theorems for this exist and Rinehart has shown that they are derivable to each other [22]. Here we choose Schwerdtfeger's formula because its notation fits well in this context. When it is applied to the exponential, the following formula holds

$$e^A = \sum_{j=1}^{t} A_j \sum_{k=0}^{s_j-1} \frac{e^{\mu_j}}{k!}(A - \mu_j I)^k \tag{3}$$

where $\mu_1, \ldots, \mu_t$ are the distinct eigenvalues of A and $s_1, \ldots, s_t$ are the corresponding multiplicities. The term A_j is the Frobenius covariant [14] associated with eigenvalue μ_j. It is computed using the corresponding left eigenvectors $x_1, \ldots, x_{s_i}$ and right eigenvectors $y_1, \ldots, y_{s_i}$ via summation: $A_j = \sum_{k=1}^{s_i} x_k y_k$.

The term e^{μ_j} in Equation (3) exponentially converges to zero for small μ_j. This novel insight explains the good approximation of the Krylov-based matrix exponential: *only the largest eigenvalues, preserved by H_m, are dominant for the matrix exponential.* This observation coincides with a result by Garren and Smith [10], who concluded that the second largest eigenvalue (the largest eigenvalue is always one) is a good estimator for the convergence to the steady state. Equation (3) backs this result, indicating that the second largest eigenvalue is the most dominant for the transient behavior, thus also for the steady state.

3.3 Error Estimates

Krylov-based methods are approximations and those come with a certain loss of information. The study of the error induced by Krylov-based methods is an extensively fast-moving field. Yet the current a-priori error bounds are known to be overly conservative [13, 23] for linear applications of Krylov-based methods, let alone for Krylov-based matrix exponentials. For this reason, Saad studied a-posteriori error estimates [23]. They are based on truncation of the real error $e^A v - V_m e^{H_m} e_1 ||v||_2$, which is the following:

$$h_{m+1,m} \sum_{k=1}^{\infty} e_m^T \phi_k(H_m) e_1 A^{k-1} v_{m+1} \tag{4}$$

The function ϕ_i is defined by the recurrence relation

$$\begin{aligned} \phi_0(z) &= e^z \\ \phi_{i+1}(0) &= 1 \\ \phi_{i+1}(z) &= \frac{\phi_i(z) - \phi_i(0)}{z} \end{aligned}$$

Note that $\phi_{i+1}(0) = 1$ is defined by continuity, making the function ϕ well defined and analytic for all z. Based on the series of Equation (4), Saad proposes several

error *estimates* because sharp error bounds are too conservative. All estimates are under-approximations of the real error because they are based on norms of the series's first terms.

An exception to this is Saad's second estimate, which is described as a *rough* estimate. It is defined as the first term of the series in Equation (4) with $\phi_1(H_m)$ replaced by e^{H_m}, because the latter is cheaper to compute (and already computed). The resulting error estimate is the following:

$$h_{m+1,m} \left| e_m^T e^{H_m} ||v||_2 e_1 v_{m+1} \right| \tag{5}$$

Saad provides little argumentation why it is safe to approximate $\phi_1(H_m)$ by e^{H_m}. The latter is actually always bigger than the former, which we shall prove as follows. The first recursion of $\phi_i(z)$ can be rewritten as the following series (by step-wise derivation from left to right):

$$\phi_1(z) = \frac{e^z - 1}{z} = \frac{1}{z} \sum_{k=1}^{\infty} \frac{z^k}{k!} = \sum_{k=1}^{\infty} \frac{z^{k-1}}{k!} = \sum_{k=0}^{\infty} \frac{z^k}{(k+1)!}$$

It is not difficult to see that the right-most equation is always smaller than the Taylor-MacLaurin series of e^z. The experimental data from Saad's study suggest that Equation (5) is a bounded over-approximation of the real error, but this result is left unproven. Nevertheless, the study shows it is empirically a good estimate and for this reason it was used as the error estimate in our experimental evaluation.

4 Experiments

To compare the Krylov-based computation of the CTMC transient distribution against the uniformization-based method, we implemented the former in the Markov Reward Model Checker (MRMC) [18]. Uniformization is set as the default numerical engine of MRMC. We made a selection of case studies from the literature describing models from system biology, queuing networks and communication protocols and ran MRMC for different configurations of each case study for comparison.

4.1 Implementation

The Krylov-based method was implemented as an extension to MRMC by intercepting the invocations to uniformization. It reuses the already implemented Harwell-Boeing sparse matrix data structure [8] to store the infinitesimal generator matrix. The Krylov project matrix V_m and the Hessenberg matrix H_m are dense and were stored using the existing matrix data structures from the GNU Scientific Library (GNU GSL).

As there is no effective method (yet) to decide the perfect subspace size m given a particular error ϵ, the Krylov-based method was implemented as an iterative algorithm by repeatedly incrementing m until the desired error level is reached (see Section 3).

4.2 Experimental Setup

All experiments were run on a cluster of twelve identical nodes. Each node is equipped with a 2.33 GHz processor and 16GB RAM. The loaded operating system is 64-bits OpenSuSE 10.3. The cluster is only used for distributing the isolated runs over the nodes to speed up the overall experiment. For all case studies, three different configurations were run and an error level of 10^{-6} was used.

UNI These are runs with MRMC's default numerical engine, uniformization, enabled and steady-state detection [18] disabled.

UNI-S These runs are similar to the previous, but with steady-state detection enabled.

KRY These are runs with the iterative Krylov-based transient implementation as described in the previous section.

4.3 Case Studies

A careful selection of case studies from literature was made to comprise different modeling domains, different model sizes and different degrees of stiffness.

CSPS A cylic server polling system that consists of $N = 5$ stations. The model was originally described by Ibe and Trivedi [15]. The measure of interest is the probability that given an upper timebound, the second station will eventually start serving. This expressed in CSL as $\mathcal{P}_{=?}(\Diamond^{[0,t]}\, \mathit{full})$.

TQN A tandem queueing network with capacity $c = 20$ described by Hermans et al. [12]. The measure of interest is the probability that the first queue sc will become full within t time units. This is expressed in CSL as $\mathcal{P}_{=?}(\Diamond^{[0,t]}\, sc = 20)$.

PTP A simple peer-to-peer file sharing protocol described by Kwiatkowska et al. [19]. The swarm consists of one client that already has all $K = 5$ blocks of the file and $N = 2$ other clients that have obtained no blocks so far. The measure of interest is whether all N clients have obtained all K blocks by time t. This is expressed in CSL as $\mathcal{P}_{=?}(\Diamond^{[0,t]}\, \mathit{done})$.

ER An enzymatic reaction model by Busch et al. [4]. It describes the enzyme-catalyzed conversion of a molecular substrate species. The measure of interest is the probability that four units of the product molecule species Pr are eventually produced within t time units. This expressed in CSL as $\mathcal{P}_{=?}(\Diamond^{[0,t]}\, Pr = 4)$.

WGC A wireless group communication protocol analyzed by Massink et al. [20]. It is a variant of a subset of the IEEE 802.11 standard describing a subnet consisting of $N = 4$ wireless stations and an access point. The number of consecutive losses of a message transmitted through the network is described by the omission degree. The higher the omission degree, the bigger the state space. In our runs, we took $OD = 32$, becoming the largest model in the selection. The measure of interest is the probability that a message sent out by the access point is not received by any station within a given timeframe t. This is expressed in CSL as $\mathcal{P}_{=?}(\Diamond^{[0,t]}\, \mathit{fail})$.

Model	States	Transitions	Stiffness
CSPS	3072	14848	1600
TQN	861	2859	400
PTP	1024	5121	0.5
ER	4011	11431	4000000
WGC	1329669	9624713	6164

Table 1. Model properties of the case studies.

From the above case studies, the first three models are part of PRISM's repository of case studies. The WGC and ER models are not part of the official PRISM repository, but are expressed in PRISM and afterwards automatically converted to MRMC's file format using PRISM's built-in converter. An overview of the models metrics can be found in Table 1. The stiffness is defined as the ratio of the largest rate to the smallest rate in the CTMC.

4.4 Results

The results of the runs for all three configurations are described in Table 2 and Table 3. The timebound column describes the different upper timebounds used in the CSL property. The #terms column describes the number of terms in the series needed for uniformization to meet the error level 10^{-6}. The column m describes the Krylov subspace dimension needed to meet the error level 10^{-6}. The memory column in Table 2 is the peak memory consumption measured using Linux's processes interface. The time column is the running time for a particular configuration. The probability column shows the computed probabilities by both algorithms (with an error margin of 10^{-6}). The probabilities of the two uniformization runs (with and without steady-state detection) are equal to these probabilities within the error level of 10^{-6}. This applies to all our case studies.

Non-Stiff Models Considering the stiffness ratios in Table 1, we classified the models PTP, TQN and CSPS as non-stiff. These models have been well-studied using uniformization-based Markov analysis tools. The results of these case studies are outlined in Table 2. It shows that the Krylov algorithm is generally slower than the uniformization-based algorithm for non-stiff models. This observation highlights a class of models for which uniformization is known to work well: non-stiff to mildly-stiff sparse models. The uniformization rates needed for these models are small and thus the number of terms needed by uniformization is small. Note that the increase of the upper timebound directly correlates with the increase in number of terms. Also, the number of terms of **UNI-S** is bigger than that of **UNI**. This is due to the steady state detection, which requires tighter left and right truncation points for determining the steady state correctly [17].

Model	Time-bound	Terms		m	Memory [KB]			Time [ms]			Probability
		UNI	UNI-S	KRY	UNI	UNI-S	KRY	UNI	UNI-S	KRY	
CSPS	10	545	653	109	2944	2992	7256	190	360	2677	0.6524983
	20	769	922	132	2944	2992	9072	340	680	5633	0.8982785
	30	941	1129	147	2940	2992	10104	490	980	9158	0.9708183
	40	1086	1303	155	2944	2992	10656	640	1280	11249	0.9916387
	50	1214	1456	157	2940	2988	10928	780	1580	11933	0.9976044
	60	1330	1595	162	2944	2992	11352	940	1860	14133	0.9993137
	70	1436	1722	162	2940	2992	11348	1070	2170	13992	0.9998034
	80	1535	1841	162	2944	2988	11352	1230	2470	13811	0.9999437
	90	1627	1952	162	2944	2992	11352	1380	2760	13651	0.9999839
	100	1715	2058	162	2940	2992	11352	1530	3060	14038	0.9999954
TQN	0.02	144	173	12	92	96	92	0	0	5	0
	0.07	149	178	21	96	92	96	0	0	16	1.7e-06
	0.12	153	182	26	92	96	96	0	0	25	0.0019782
	0.17	157	186	29	96	92	96	0	0	32	0.0550075
	0.22	161	190	31	96	92	92	0	0	39	0.2875958
	0.27	166	195	32	92	96	92	0	0	41	0.6267612
	0.32	170	199	34	96	92	96	0	0	51	0.8643245
	0.37	173	203	35	96	92	92	0	0	52	0.9638449
	0.42	175	208	35	96	96	92	10	0	52	0.992505
	0.47	177	212	39	96	96	96	0	0	67	0.9987298
PTP	1	163	192	20	96	96	92	10	10	18	0.3892596
	2	177	212	23	96	96	92	10	20	23	0.9055015
	3	184	220	25	96	96	96	10	20	28	0.987485
	4	190	228	25	92	96	92	10	20	28	0.9983193
	5	195	234	26	92	96	96	20	20	31	0.9997729
	6	200	240	26	92	96	96	10	20	30	0.9999693
	7	204	245	26	96	96	96	20	20	31	0.9999958
	8	208	250	26	96	92	92	20	20	32	0.9999994
	9	212	254	27	96	96	96	20	20	33	0.9999999
	10	216	259	28	96	96	92	20	20	36	1

Table 2. Verification times and memory consumption on the non-stiff models.

Besides uniformization's well explainable performance characteristics for non-stiff to mildly stiff models, the Krylov-based method has a higher constant cost due to the Arnoldi iteration which computes a dense projection and Hessenberg matrix. Furthermore, despite the small size of the non-stiff models, a relatively large —though absolutely measured small— subspace dimension is needed to meet the desired error level.

Stiff Models The models ER and WGC are considered to be stiff. The results for these case studies are outlined in Table 3. Note that the probabilities for the WGC case study are all zero. This is expected behaviour since we chose a high omission degree (32) in order to increase the state space size. High omission

Model	Time-bound	Terms		m	Time [ms]			Probability
		UNI	UNI-S	KRY	UNI	UNI-S	KRY	
ER	100	7638	9166	51	490	930	372	0.1408622
	200	10801	12960	54	950	1840	442	0.5621672
	300	13227	15872	55	1400	2740	470	0.8457958
	400	15273	18326	55	1880	3640	454	0.9563306
	500	17075	20489	56	2340	4550	455	0.9892358
	600	18704	22444	56	2770	5420	484	0.9975874
	700	20203	24243	56	3260	6340	488	0.9994953
	800	21597	25916	56	3720	7220	492	0.9998998
	900	22907	27488	59	4170	8150	556	0.9999809
	1000	24146	28975	60	4640	9030	581	0.9999965
WGC	10000	578	693	33	171490	342550	109549	0
	20000	816	979	34	325930	650460	116335	0
	30000	999	1197	35	477890	953350	123042	0
	40000	1153	1382	35	628630	1253070	123357	0
	50000	1288	1545	36	778280	1551270	130295	0
	60000	1411	1692	36	927990	1848150	130355	0
	70000	1523	1827	36	1076650	2144230	130271	0
	80000	1629	1953	36	1225090	2439050	134827	0
	90000	1727	2071	36	1372840	2734100	130291	0
	100000	1820	2184	37	1519720	3026840	145825	0

Table 3. Verification times on the stiff models.

degrees significantly reduce the probability that the message is not received (cf. [30]).

The results show that for these models, the Krylov-based method is an order of magnitude faster than uniformization. This can be seen in Figures 2 and 3 which plot verification times (in ms) against the time bound of the CSL property. When uniformization is performed with steady-state detection, Krylov's performance gain over uniformization even increases. The figures show that the running times of uniformization (with and without steady-state detection) are obviously linear. The running times for the Krylov runs appear to be constant. A linear regression however showed that the slope of Krylov's running times are also linear, though with a very slow slope, whereas the slopes of uniformization are significantly higher.

These performance characteristics are explainable akin to the non-stiff models. Uniformization is sensitive to the uniformization rate and the upper timebound. The high stiffness is the direct cause for the former and causes uniformization to compute a significant amount of terms in order to satisfy the desired error level. Larger upper timebounds additionally increase that amount of terms. The Krylov-based method does not suffer much from the stiffness, as the infinitesimal generator matrix can be approximated accurately by a small Hessenberg matrix, and thus the Krylov technique terminates quickly. This compensates for the relatively high costs of the Arnoldi iteration.

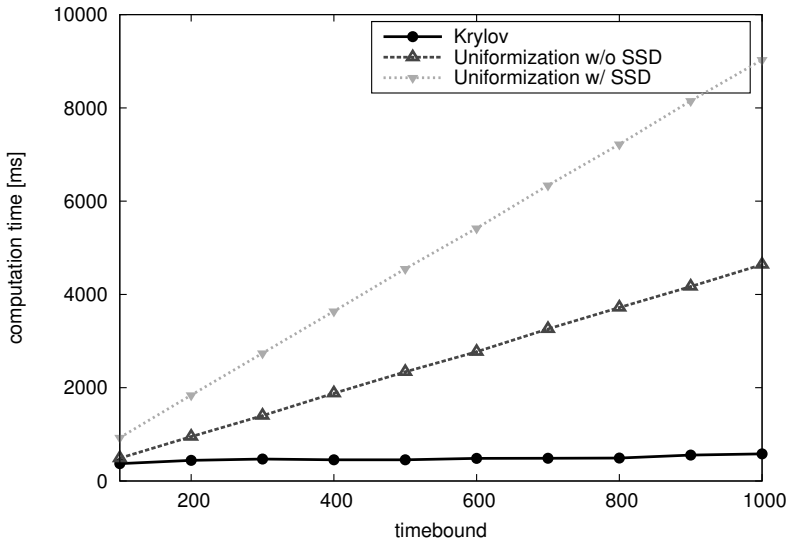

Fig. 2. Verification times of $\mathcal{P}_{=?}(\Diamond^{[0,t]}\, Pr = 4)$ with increasing timebounds t on the ER model.

Peak Memory Consumption Table 2 has a column of the memory consumption on non-stiff models. Due to space issues, the memory column is not present in Table 3. However summarized, for the stiff ER model, the memory consumption of uniformization was constant on 3 MB, of uniformization with steady state detection enabled was constant on 3.1 MB and of Krylov was between 5.5 to 7.0 MB. For the WGC model, the memory consumption of uniformization was constant on 384 MB, of uniformization with steady state detection was around 402 MB and of Krylov was between 1059 to 1141 MB.

These numbers indicate that uniformization has a clear advantage over the Krylov-based method when it comes to peak memory consumption. This can be explained to the storage of the dense projection matrix which is of size $m \times dim(A)$. Krylov's memory consumption increases for larger timebounds, although the increase is slow.

5 Related Work

Adaptive Uniformization [27] is an alternative method to handle stiff CTMCs. It essentially reduces the state space of a CTMC by slicing away "in-active" states and keeping the active states. The latter is defined as the states that are reachable within a predefined number, n say, of steps in the uniformized matrix. The usefulness of the method depends on the chosen n and the model itself. This is in contrast to the Krylov-based approach, which does not need additional input parameters. The advantage of adaptive uniformization, however, is that

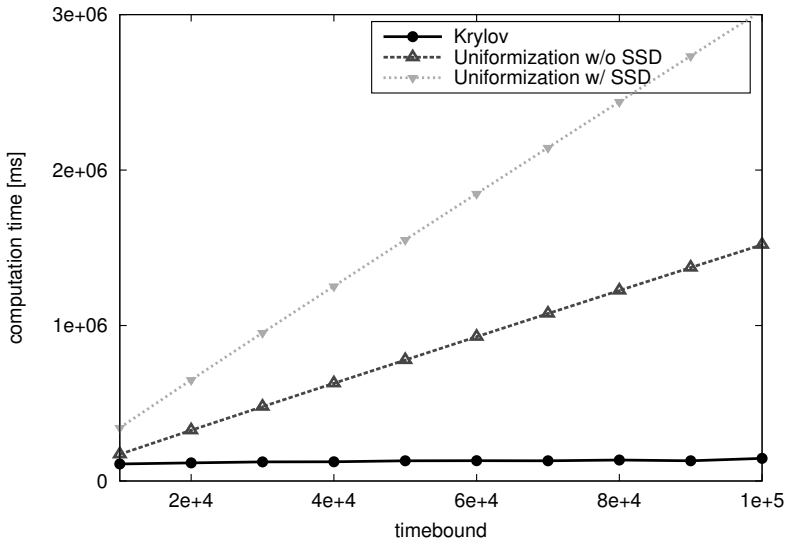

Fig. 3. Verification times of $\mathcal{P}_{=?}(\Diamond^{[0,t]}\, \mathit{fail})$ with increasing timebounds t on the WGC model.

an a priori error estimate can be given, as for standard uniformization. Adaptive uniformization has recently been combined with abstraction techniques [11].

ODE Solvers can be used to compute the transient by considering the generator matrix as a system of ordinary differential equations. Numerous techniques and optimizations have been developed to handle stiff ODEs, which is a well-known phenomenon in various scientific disciplines. The available ODE solver libraries are generally extensively tested and understood. The drawback is that ODE solvers are typically developed for a general class of problems and thus not optimized for the Markovian case. Nevertheless, we briefly experimented with Intel's ODE solver package by interfacing it with MRMC. Unfortunately the package did not expose sufficient information about the induced error, rendering its application for the transient unsatisfied.

Uniformization Power [1] is an optimization over Jensen's uniformization to increase the numerical stability and performance. It essentially performs scaling and squaring by subdividing the time interval to $t' = \frac{t}{2^n}$ and modifying equation 2 to calculate $\pi(t)$ as $\pi(t) = \mathbf{\Pi}^n$, with $\mathbf{\Pi} = (\sum_{k=0}^{\infty} e^{-\Lambda t'} \frac{(\Lambda t')^k}{k!} \mathbf{P}^k) \cdot \pi(0)$. The advantage is that it requires much smaller steps for large Λ than standard uniformization. The drawback is that matrix $\mathbf{\Pi}$ is dense, and thus the amount of memory required is excessive.

A Priori Error Bounds, especially when they are sharp, are more desirable over the a posteriori error estimate used in our experiments, because they give an a priori indication of the expected running time. Initially we experimented with several a priori error bounds developed by Hochbruck et al. [13], but found

these error bounds too conservative when compared to Saad's a posteriori error estimates. Hochbruck et al.'s bounds are also expensive to compute because they are based on the numerical range of the input matrix. The study of improved a priori error bounds for Krylov-based matrix exponentials is however an active field and advancements from there are directly applicable to the computation of the transient.

Expokit is a toolkit developed by Sidje [24] to compute matrix exponentials. It also comes with an optimized version for computing the transient of Markov chains. The error bounds used are based on the assumption that the elements of the transient vector sums up to 1. This however does not hold for CSL model checking because the initial vectors are interpreted differently, causing the elements not necessarily sum up to one. The toolkit also comes with a time-stepping scheme of the matrix exponential which subdivides the time bound and computes the transient in steps. Our implementation in MRMC is inspired by Expokit's MatLab code.

Regenerative Randomization with Laplace Transform Inversion is a technique for transient analysis of Markov reward model by Carrasco [5]. Here, the truncated transformed model obtained in this regenerative method is solved using a Laplace transform inversion algorithm instead of standard uniformization. The main difficulty in this technique is to find appropriate regeneration points. This is doable for certain classes of models, such as failure-repair models, but in general is a non-trivial issue. For stiff models with absorbing states (as for time-bounded reachability properties), this technique outperforms standard uniformization when the model is not very large.

6 Future Work

The field of Krylov-based subspaces is relatively young, and though its applications are spreading fast, there are several gaps open for study.

An open question is the relation of stiffness and the eigenvalues spectrum of a CTMC. Our debugging observations hint to a correlation between the stiffness and the way eigenvalues are spread in their spectrum. For the non-stiff models in the selected case studies, the eigenvalues are homogeneously spread across the spectrum. There is however no theoretical evidence to claim that this generally holds. There is a report however that a high clustering of eigenvalues is beneficial for the Krylov-based method [29] and this would back our experimental data. Further study is required to fully understand this.

An interesting direction of future work is improving the error bounds. The current a priori error bounds for Krylov-based matrix exponentials are conservative and expensive to compute. It would be desirable to have a cheap, yet reliable bound, enabling us to automatically determine the required subspace dimension in advance. In the mean time, a posteriori error estimates are the best choice for practical applications. In this context, it is an open question whether the currently used error estimate (see Equation 5) is a bounded over-approximation of the real error (see Equation 4).

The last point regards the trend towards parallelized architectures. The Krylov-based method is highly amenable to such architectures because it is mostly based on matrix-vector multiplications. Data-parallelism in graphics cards can be exploited to achieve a significant performance gain. This gain would be especially visible for large models. For DTMCs, such architectures have already been exploited with positive results [3].

7 Conclusions

This paper is an experimental report on the use of Krylov-based techniques for probabilistic model checking. We showed using Schwerdtfeger's formula how the Krylov-based method is well suited for computing the transient distribution. We thus implemented a Krylov-based method for model checking CTMCs as an alternative to uniformization in MRMC. The experimental results on a selection of five case studies from literature revealed that the Krylov-based implementation is an order of magnitude faster than uniformization on stiff models. This comes at the cost of increased memory consumption. If running time is the bottleneck, and if the model is stiff, our observations indicate that for time-bounded reachability properties a Krylov-based method is preferable over the commonly used uniformization.

References

1. H. Abdallah and R. Marie. The uniformized power method for transient solutions of Markov processes. *Computers & Operations Research*, 20(5):515 – 526, 1993.
2. C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen. Model-checking algorithms for continuous-time Markov chains. *IEEE Trans. on Softw. Eng.*, 29(6):524–541, 2003.
3. D. Bosnacki, S. Edelkamp, and D. Sulewski. Efficient probabilistic model checking on general purpose graphics processors. In *Proc. 16th Int. SPIN Workshop*, volume 5578 of *LNCS*, pages 32–49. Springer, 2009.
4. H. Busch, W. Sandmann, and V. Wolf. A numerical aggregation algorithm for the enzyme-catalyzed substrate conversion. In *Computational Methods in Systems Biology*, volume 4210 of *LNCS*, pages 298–311. Springer Verlag, 2006.
5. J.A. Carrasco. Transient analysis of rewarded continuous time Markov models by regenerative randomization with laplace transform inversion. *The Conputer Journal*, 46(1):84–99, 2003.
6. T. Chen, T. Han, J.-P. Katoen, and A. Mereacre. Quantitative model checking of continuous-time Markov chains against timed automata specification. In *LICS*, pages 309–318, 2009.
7. E. de Souza e Silva and H. R. Gail. Transient solutions for Markov chains. In W. Grassmann, editor, *Computational Probability*, pages 43–81. Kluwer Academic Publishers, 2000.
8. I. Duff, R. Grimes, and J. Lewis. User's guide for the Harwell-Boeing sparse matrix collection. Technical Report TR/PA/92/86, CERFACS, 1992.
9. B.L. Fox and P.W. Glynn. Computing Poisson probabilities. *Commun. ACM*, 31(4):440–445, 1988.

10. S.T. Garren and R.L. Smith. Estimating the second largest eigenvalue of a Markov transition matrix. *Bernoulli*, 6(2):215–242, 2000.
11. T.A. Henzinger, M. Mateescu, and V. Wolf. Sliding window abstraction for infinite Markov chains. In *Computer-Aided Verification*, volume 5643 of *LNCS*, pages 337–352. Springer, 2009.
12. H. Hermanns, J. Meyer-Kayser, and M. Siegle. Multi terminal binary decision diagrams to represent and analyse continuous time Markov chains. In B. Plateau, W. Stewart, and M. Silva, editors, *Proc. 3rd International Workshop on Numerical Solution of Markov Chains (NSMC'99)*, pages 188–207. Prensas Universitarias de Zaragoza, 1999.
13. M. Hochbruck and C. Lubich. On Krylov subspace approximations to the matrix exponential operator. *SIAM on Numerical Analysis*, 34(5):1911–1925, 1997.
14. R.A. Horn and C.R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1986.
15. O. Ibe and K. Trivedi. Stochastic Petri net models of polling systems. *IEEE Journal on Selected Areas in Communications*, 8(9):1649–1657, 1990.
16. A. Jensen. Markoff chains as an aid in the study of markoff processes. In *Skand. Aktuarietidskrift*, volume 36, pages 87–91, 1953.
17. J.-P. Katoen and I.S. Zapreev. Safe on-the-fly steady-state detection for time-bounded reachability. In *QEST*, pages 301–310, Los Alamitos, 2006. IEEE CS.
18. J.-P. Katoen, I.S. Zapreev, E.M. Hahn, H. Hermanns, and D.N. Jansen. The ins and outs of the probabilistic model checker MRMC. In *Quantitative Evaluation of Systems*, pages 167–176. IEEE CS Press, 2009.
19. M. Kwiatkowska, G. Norman, and D. Parker. Symmetry reduction for probabilistic model checking. In *Computer Aided Verification*, volume 4144 of *LNCS*, pages 234–248. Springer Verlag, 2006.
20. M. Massink, J.-P. Katoen, and D. Latella. Model checking dependability attributes of wireless group communication. In *Dependable Systems and Networks (DSN)*, pages 711–720. IEEE CS Press, 2004.
21. C. Moler and C. Van Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Review*, 45(1):3–49, 2003.
22. R.F. Rinehart. The equivalence of definitions of a matrix function. *The American Mathematical Monthly*, 62(6):395–414, 1955.
23. Y. Saad. Analysis of some Krylov subspace approximations to the matrix exponential operator. *SIAM Journal on Numerical Analysis*, 29(1):209–228, 1992.
24. R. Sidje. Expokit: a software package for computing matrix exponentials. *ACM Trans. Math. Softw.*, 24(1):130–156, 1998.
25. R. Sidje and W.J. Stewart. A survey of methods for computing large sparse matrix exponentials arising in Markov chains. In *Markov Chains, Computational Statistics and Data Analysis 29*, pages 345–368, 1996.
26. L.N. Trefethen and D. Bau III. *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, 1997.
27. A. van Moorsel and W. Sanders. Adaptive uniformization. *Communications in Statistics - Stochastic Models*, 10(3):619–648, 1994.
28. R.C. Ward. Numerical computation of the matrix exponential with accuracy estimate. *SIAM Journal on Numerical Analysis*, 14(4):600–610, 1977.
29. R. Weiss. Error-minimizing Krylov subspace methods. *SIAM Journal on Scientific Computing*, 15(3):511–527, 1994.
30. I.S. Zapreev. *Model Checking Markov Chains: Techniques and Tools*. PhD thesis, Univ. of Twente, 2008.