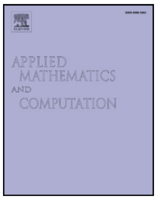
Applied Mathematics and Computation 293 (2017) 96–111

Contents lists available at ScienceDirect

Applied Mathematics and Computation

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



Implicit ODE solvers with good local error control for the transient analysis of Markov models

Víctor Suñé[a,*], Juan Antonio Carrasco[b]

[a] Departament d'Enginyeria Electrònica, Universitat Politècnica de Catalunya, C. Colom, 1 08222 Terrassa, Catalonia, Spain
[b] Departament d'Enginyeria Electrònica, Universitat Politècnica de Catalunya, Av. Diagonal, 647, planta 9 08028 Barcelona, Catalonia
Spain

ARTICLE INFO

Keywords:
Implicit ODE solvers
Continuous-time Markov chains
Transient analysis

ABSTRACT

Obtaining the transient probability distribution vector of a continuous-time Markov chain (CTMC) using an implicit ordinary differential equation (ODE) solver tends to be advantageous in terms of run-time computational cost when the product of the maximum output rate of the CTMC and the largest time of interest is large. In this paper, we show that when applied to the transient analysis of CTMCs, many implicit ODE solvers are such that the linear systems involved in their steps can be solved by using iterative methods with strict control of the 1-norm of the error. This allows the development of implementations of those ODE solvers for the transient analysis of CTMCs that can be more efficient and more accurate than more standard implementations.

1. Introduction

Consider a finite continuous-time Markov chain (CTMC) $X=\{X(t);t\geq 0\}$ with infinitesimal generator. The state space of X will be denoted by $\Omega=\{1,\dots,m\}$, its initial probability distribution (column) vector by $\mathbf{p}^0$, and the transpose of its infinitesimal generator by $\mathbf{Q}=(q_{i,j})_{1\leq i,j\leq m}$, i.e., $q_{i,\,j}$, $i\neq j$, will denote the transition rate from state j to state i and $|q_{i,i}|=-q_{i,i}=\sum_{\substack{j=1\\ j\neq i}}^{m} q_{j,i}$ will denote the output rate from state i. In this paper, we will be concerned with the computation of the transient probability distribution vector of X, $\mathbf{p}(t)=(P[X(t)=i])_{1\leq i\leq m}$, $t\geq 0$, when $\mathbf{Q}$ is large and sparse.

When $\mathbf{Q}$ is large, the only practical methods to compute $\mathbf{p}(t)$ are uniformization (also called randomization) [1,2], uniformization-based methods (see, e.g., [3–5]), the Krylov-based method described in [6], and formulating $\mathbf{p}(t)$ as the solution of an initial value problem (IVP) for the Kolmogorov system of ordinary differential equations (ODEs) and solving the IVP using an ODE solver. When $qt\gg 1$, where $q=\max_{1\leq i\leq m}|q_{i,i}|$, this latter alternative can be very attractive from the point of view of run-time computational cost. The IVP to be solved is

$$\begin{cases}\dfrac{\mathrm{d}\mathbf{p}(t)}{\mathrm{d}t}=\mathbf{Q}\,\mathbf{p}(t), \quad t\geq 0\,,\\ \mathbf{p}(0)=\mathbf{p}^0\,.\end{cases} \tag{1}$$

That IVP is stable because, using the Gershgorin circle theorem [7], which states that the eigenvalues of $\mathbf{Q}^{\mathrm{T}}$, which are the same as those of $\mathbf{Q}$, lie in the union of the m discs in the complex plane with centers $q_{i,\,i}\leq 0$ and radii $\sum_{\substack{j=1\\ j\neq i}}^{m}|q_{j,i}|=|q_{i,i}|$,

[*] Corresponding author.
E-mail addresses: victor.sunye@upc.edu (V. Suñé), juan.a.carrasco@upc.edu (J.A. Carrasco).

http://dx.doi.org/10.1016/j.amc.2016.08.009
0096-3003/© 2016 Elsevier Inc. All rights reserved.

it turns out that the eigenvalues of $\mathbf{Q}$ different from 0 have negative real part. As in [8], we will say that the IVP (1) is stiff if $qt \gg 1$. That such a stiffness criterion is reasonable is argued in [8] as follows. For each distinct eigenvalue λ_j of $\mathbf{Q}$, component i of $\mathbf{p}(t)$, $1 \leq i \leq m$, will include a factor of the form $\sum_{k=0}^{m_j-1} c_{i,j,k} t^k \mathrm{e}^{\lambda_j t}$, where m_j denotes the multiplicity of λ_j and the $c_{i,\,j,\,k}$ are appropriate constants. Consequently, when $\max_j |\lambda_j| t \gg 1$, we can expect $\mathbf{p}(t)$ to have components with large relative variation in the time interval $[0, t)$, making the IVP (1) stiff. But, again by the Gershgorin circle theorem, $\max_j |\lambda_j|$ is bounded from above by $\|\mathbf{Q}\|_1 = 2q$. Therefore, if $qt \gg 1$, it is reasonable to expect the IVP (1) to be stiff. For an ODE solver applied to the solution of the IVP (1) to be effective when the IVP is stiff, the ODE solver should be implicit and, preferably, A-stable [9]. In this paper, we will consider the use of implicit ODE solvers to solve the IVP (1).

Several papers have considered the use of implicit ODE solvers for solving the IVP (1) [8,10–16]. In [8], the performance of TR-BDF2 [17], an L-stable [18] linear two-step implicit ODE solver, was analyzed. The conclusions were that the run-time computational cost of TR-BDF2 was lower than the run-time computational cost of both uniformization and the explicit ODE solver RKF45 [19], but that, being only a second-order method, it required a small step size to achieve high accuracies. In [10], TR-BDF2 and a third-order L-stable implicit Runge–Kutta ODE solver (IRK3) were compared between them and with several uniformization variants. The conclusions were that both TR-BDF2 and IRK3 were preferable to the uniformization variants and that when the required accuracy was high, the run-time computational cost of IRK3 was lower than that of TR-BDF2. The performances of TR-BDF2 and IRK3 were analyzed in [11] for the case in which X is nearly completely decomposable [20], with the conclusion that those implicit ODE solvers should be implemented exploiting the near complete decomposability of X. In [12], the performance of IRK3 was analyzed when X is acyclic so that $\mathbf{Q}$ can be put into lower triangular form and the linear system involved in each step of the method can be solved very efficiently using direct methods. The conclusions were that IRK3 could be highly accurate and that its run-time computational cost was lower than the run-time computational cost of a uniformization variant and lower than the run-time computational cost of an improved version of an specific method for the transient analysis of acyclic CTMCs [21]. In [13], it was proposed to combine RKF45 and either TR-BDF2 or IRK3 in such a way that RKF45 is used up to some intermediate time and the other method is used from that point on. The result was a significant reduction in run-time computational cost compared with any of the ODE solvers used in isolation. In [14], the performance of several ODE solvers was analyzed (implicit Euler [22], trapezoidal rule [22], 2-stage Radau IIA [18,23], 2-stage Gauss [24], a singly diagonally implicit Runge–Kutta ODE solver of order two with two stages, and another one of order three with two stages) with the conclusion that the trapezoidal rule implemented using extrapolation, which gives a fourth-order implicit method, was a good alternative. In [15], it was proposed to combine uniformization and TR-BDF2 in such a way that uniformization is used up to some intermediate time and TR-BDF2 is used from that point on with significant reductions in run-time computational cost compared with TRB-BDF2 and uniformization. Finally, in [16] some of the ODE solvers in the VODPK package [25] were compared with uniformization and with the Krylov-based method described in [6], with the conclusion that for $\|\mathbf{Q}\|t > 500$, the Krylov-based method and a variant of the implicit ODE solver based on backward differentiation formulae (BDFs) with variable coefficients available in the package had a run-time computational cost lower than that of uniformization.

Applied to the IVP (1), an ODE solver produces approximations $\mathbf{p}_n$, $n = 0, 1, \ldots,$ to $\mathbf{p}(t_n)$ at a set of time points t_n starting from $t_0 = 0$ and $\mathbf{p}_0 = \mathbf{p}^0$. Step $n \geq 1$ spans the solution from $t = t_{n-1}$ to $t = t_n$ and has step size $h_n = t_n - t_{n-1}$. If the ODE solver is implicit, computing the approximation $\mathbf{p}_n$, $n = 1, 2, \ldots,$ requires solving one or more linear systems of equations with matrices related to $\mathbf{Q}$. In the case of the implicit ODE solvers considered in [8,10–16], the linear systems were solved by using iterative methods including Gauss–Seidel (GS) [26], a mixture of Jacobi [26] and GS, successive overrelaxation (SOR) [26], and a variant of restarted GMRES [27]. The convergence of some of those methods was only analyzed in [14], where it was noted that the Jacobi method was guaranteed to converge for some of the implicit ODE solvers considered, in [8,10–16]. At the best of the authors' knowledge, the impact of the errors with which the linear systems are solved has never been explicitly analyzed. It is clear, however, that if those errors are large, they can affect adversely not only the accuracy but also the run-time computational cost of the ODE solvers. Roughly speaking, the reason is that those errors may introduce significant spurious components in the computed solution associated with eigenvalues of $\mathbf{Q}$ of large absolute value, which are the ones that will limit most the step size and, then, to reduce to an acceptable level the impact of those spurious components in the error of the computed solution, the ODE solver may be forced to take steps smaller than the ones it would take if those errors were absent. The errors introduced in the solution of the linear systems will be referred to as step approximation errors. To clarify, the local error at step n of the ODE solver, $\mathbf{e}_n = \mathbf{p}_n - \mathrm{e}^{\mathbf{Q}h_n}\mathbf{p}_{n-1}$, has three components. The first component, called here the inherent local error, reflects the inherent error of the approximation formula underlying the ODE solver and is $\mathbf{i}_n = \mathbf{p}_n^+ - \mathrm{e}^{\mathbf{Q}h_n}\mathbf{p}_{n-1}$, where $\mathbf{p}_n^+$ is $\mathbf{p}_n$ computed exactly. The second component, which is the one that we call step approximation error, reflects the error introduced by solving the linear systems using iterative and, hence, inexact methods. Formally, that error is $\mathbf{s}_n = \widehat{\mathbf{p}}_n - \mathbf{p}_n^+$, where $\widehat{\mathbf{p}}_n$ is $\mathbf{p}_n$ computed by solving the linear systems using iterative methods, ignoring the impact of round-off errors. The third component of the local error collects the impact of round-off errors on $\mathbf{p}_n$.

The 1-norm is a convenient norm for measuring and controlling local error components for the IVP (1). The reason is that $(\mathrm{e}^{\mathbf{Q}\tau})^{\mathrm{T}} = \mathrm{e}^{\mathbf{Q}^{\mathrm{T}}\tau}$, $\tau \geq 0$, being a stochastic matrix, we have $\|\mathrm{e}^{\mathbf{Q}\tau}\|_1 = 1$, $\tau \geq 0$, implying that, measured in the 1-norm, local errors will not be amplified as they propagate through the solution to define the global error. Therefore, according to the discussion in the previous paragraph, it is convenient that $\|\mathbf{s}_n\|_1$, $n = 0, 1, \ldots,$ be small enough. But, $\|\mathbf{s}_n\|_1$ will be determined by the stopping criterion of the iterative method with which the involved linear systems are solved and unless