Applied Mathematics and Computation 289 (2016) 353–370

Contents lists available at ScienceDirect

Applied Mathematics and Computation

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

Improving the stochastic direct simulation method with applications to evolution partial differential equations

Flavius Guiaş[a,*], Pavel Eremeev[b]

[a] Dortmund University of Applied Sciences and Arts, Sonnenstr. 96, 44139 Dortmund, Germany
[b] Faculty of Mathematics, Dortmund University of Technology, Vogelpothsweg 87, 44221 Dortmund, Germany

ARTICLE INFO

MSC:
65C20
68U20
65M06
65M20
65M99
60J75
65Y99

PACS:
82.20.Wt
87.10.Mn

Keywords:
Stochastic simulation method
Markov jump processes
Method of lines
Reaction–diffusion
Nonlinear diffusion
Radiation diffusion

ABSTRACT

The stochastic direct simulation method is a numerical scheme for approximating the solutions of ordinary differential equations by path simulations of certain associated Markov jump processes. Its particular features make it suitable especially when applied to ODE systems originating from the spatial discretization of PDEs. The present paper provides further improvements to this basic method, which are based on the predictor–corrector principle. They are made possible by the fact that in its context a full path of the jump process is computed. With this full set of data one can perform either Picard iterations, Runge–Kutta steps, or a combination, with the goal of increasing the order of convergence. The improved method is applied to standard test problems such as a reaction–diffusion equation modeling a combustion process in 1D and 2D as well as to the radiation–diffusion equations, a system of two partial differential equations in two space dimensions which is very demanding from the computational point of view. Further optimization aspects which are also discussed in this paper are related to the efficient implementation of sampling algorithms based on Huffman trees.

1. Introduction

Works such as [11] and [4] have demonstrated the possibility of approximating the solutions of n-dimensional systems of ordinary differential equations $\dot{X} = F(X)$, $F = (F_i)_{i=1}^n$ by suitable Markov jump processes. The characterization of Markov processes by their infinitesimal generators and the estimation of the martingale terms (trendless stochastic perturbation of the average deterministic dynamics) by the techniques from these papers yields a central limit theorem which gives the theoretical background for the approximation result. This approximation property can be exploited in order to develop numerical schemes for ODE systems by simulating the paths of these Markov processes.

A special choice of jump process is that in which only one component is changed at a time, by an increment of $\pm 1/N$, where N is a resolution parameter corresponding to the weight of a 'numerical particle'. This procedure can be implemented by the following scheme, referred to as the *stochastic simulation method* [5,6] or the *direct ODE simulation algorithm* (dODE) [8]:

* Corresponding author. Tel.: +492319112260.
E-mail addresses: flavius.guias@fh-dortmund.de (F. Guiaş), pavel.eremeev@tu-dortmund.de (P. Eremeev).

http://dx.doi.org/10.1016/j.amc.2016.05.033
0096-3003/

1. Given the state $x_N = x_N(t)$ of the Markov process at time t:
2. Choose a component i with probability proportional to $|F_i(x_N)|$.
3. The waiting time is given by $\delta t = -\log U/\lambda$, where U is a uniformly distributed RV on (0, 1) and $\lambda = N\sum_{j=1}^{n}|F_j(x_N)|$. The time step δt is exponentially distributed with parameter λ.
4. Update the value of the time variable: $t = t + \delta t$.
5. Update the value of the sampled component: $x_{N,i} \mapsto x_{N,i} + \frac{1}{N} sign(F_i(x_N))$.
6. Update the values $F_j(x_N)$ for all j for which $F_j(x_N)$ depends on the sampled component i.
7. GOTO 1.

One can note that the random time steps δt are automatically adapted, and that after incrementing the time by this quantity, one updates the value of only one of the components of the system. The component i which has to be updated is chosen at random, with probability proportional to $|F_i(x_N)|$, i.e., with the absolute value of the right hand side of the corresponding equation. Note that the time steps Δt used by deterministic schemes are typically larger and that in this case one updates all components at once at the given time step.

The first to use the above stochastic approach in order to approximate ODEs modeling coupled chemical reactions was Gillespie in [5] and several follow-up papers summarized in [6]. The main improvement, introduced also by Gillespie, was the so-called *tau-leap* method, in which one computes basically the number of jumps of each type in a given (larger) time-interval and then performs all transitions at once. The convergence of this method was rigorously analyzed by other authors in [1].

Taking a look at the (dODE) scheme, it is clear that for large system dimensions n, this method is computationally affordable only if in step 6 one has to update $O(1)$ components. This condition is fulfilled in the case that the ODE system is a spatially discretized version of a partial differential equation. The paper [8] analyses the applicability of this scheme to several one-dimensional partial differential equations. In approximating diffusive dynamics this scheme turned out to be faster and to have smaller stochastic fluctuations than other stochastic simulations schemes approximating the same dynamics: random walk or flux simulation. Moreover, in contrast to explicit deterministic methods, the (dODE) method turned out to be stable also on nonuniform grids. This feature allowed the use of adapted grids or even moving grids for diffusion–transport dynamics described, for example, by the Burgers equation. Finally, the method turned out to be able to approximate also a free-boundary problem, namely the Black–Scholes equation for American put options.

A crucial element which influences the efficiency of the (dODE) scheme is the sampling step 2. The paper [8] was concerned mainly with the approximation aspects of the method and less with an optimal implementation which would enhance its performance. This was the main reason which restricted the numerical investigations only to one-dimensional PDEs.

The present paper presents a further development of the method discussed above. By taking a close look at the (dODE) algorithm, one realizes that it performs a complete path simulation of the n-dimensional Markov jump process, a feature which is not fully exploited. The user of any numerical method is usually interested in the values at given times, separated, say, by a constant increment Δt. Standard deterministic schemes (explicit or implicit) compute the value $X(t+\Delta t)$ by knowing the value of $X(t)$ and, in the case of multistep methods, also other values at past moments from the discrete vector of time steps. However, the (dODE) method, in order to compute the value $X(t+\Delta t)$, has to perform the simulation of a full path of a Markov jump process between $X(t)$ and $X(t+\Delta t)$ in order to approximate the value at the end of this time interval. This feature can be used to improve the order of convergence in at least two ways, based on the predictor–corrector principle. In both situations we evaluate the function F corresponding to the right hand side of the ODE system at the paths simulated by (dODE), which serve as predictor. Then, by computing the integrals of these quantities (integrals of step functions) we can obtain improved approximations.

The first approach consists in performing a Picard iteration:

$$x_N(t+\Delta t) = x_N(t) + \int_t^{t+\Delta t} F(\tilde{x}_N(s))ds \tag{1}$$

where $\tilde{x}_N(s)$ for $s \in [t, t+\Delta t]$ is the path simulated employing (dODE) with initial value $x_N(t)$. The computation of the integral is possible, since $F(\tilde{x}_N(s))$ is an n-dimensional, vector valued step function, which in fact is known. For comparison, note that the explicit Euler method reads

$$x(t+\Delta t) = x(t) + F(x(t))\Delta t = x(t) + \int_t^{t+\Delta t} F(x(t))ds. \tag{2}$$

It can be seen that here one integrates only the (constant) function with value given by $F(x(t))$, while in the new method we integrate in time a step function which is completely known and is apriori a better approximation of the solution of the system of ODEs over the whole interval of integration $[t, t+\Delta t]$ than the constant value used in the Euler scheme.

The second approach consists in a Runge–Kutta step:

$$x_N(t+\Delta t) = x_N(t) + \int_t^{t+\Delta t} \tilde{P}(s)ds \tag{3}$$