A Fast Algorithm for High-Dimensional Markov Processes with Finite Sets of Transition Rates

Hans Ekkehard Plesser† and Dietmar Wendt‡

†Laboratory for Neural Modeling, FRP, RIKEN, 2-1 Hirosawa, Wako-shi, Saitama 351-01, Japan (plesser@yugiri.riken.go.jp)
‡Institut für Theoretische Physik, RWTH Aachen, 52056 Aachen, Germany (D.Wendt@physik.rwth-aachen.de)

Abstract

The discrete class algorithm presented in this paper is an efficient simulation tool for stochastic processes governed by a reasonably small set of transition rates. The algorithm is presented, its performance compared to prevailing methods and applications to epitaxial growth and neuronal models are sketched.

I. Introduction

Stochastic processes play a crucial role in many fields of science and technology and have received much attention ever since the ground-breaking work by Einstein, Smoluchowski and others at the beginning of the century [1, 2]. While many low-dimensional stochastic processes can be treated analytically [3, 4], this is no longer the case for spatially extended, high-dimensional systems, such as diffusion-limited reaction-diffusion systems, epitaxial growth, population dynamics or neuronal interactions [5, 6, 7, 8].

The unifying feature of all these systems is that their development in time is given by a master equation

$$\frac{\partial}{\partial t}\mathcal{P}\left(\mathbf{n}, t \,|\, \mathbf{n}^{(0)}\right) = \sum_{\mathbf{n}'} W_{\mathbf{n}'\to\mathbf{n}}\mathcal{P}\left(\mathbf{n}', t \,|\, \mathbf{n}^{(0)}\right) - W_{\mathbf{n}\to\mathbf{n}'}\mathcal{P}\left(\mathbf{n}, t \,|\, \mathbf{n}^{(0)}\right)$$

for the probability of the system to be in state $\mathbf{n}$ at time t if it was in state $\mathbf{n}^{(0)}$ at time $t^{(0)}$. Here, $W_{\mathbf{n}\to\mathbf{n}'}$ are transition rates and $\mathbf{n}$ is a vector in an m-dimensional discrete state space. Since the master equation can rarely be solved analytically, efficient simulation methods for the generation of trajectories obeying the equation are of tantamount importance.

In the next section, we present the highly efficient *discrete class algorithm* (DCA) for the simulation of systems governed by a reasonably small set of different transistion rates. In section 3 we apply the algorithm to a simple model of epitaxial growth and demonstrate the speed-up compared to prevailing methods. Finally, in section 4, we show how the DCA can be used to study large neural networks.

II. The Discrete Class Algorithm

The discrete class algorithm is an extension of the minimal process method [9] introduced by Gillespie [10]. This elegant algorithm proceeds from state $\mathbf{n}$ at t to $\mathbf{n}'$ at $t' = t + \tau$ as follows:

1. Calculate the total rate for leaving state $\mathbf{n}$: $W_{\mathbf{n}} = \sum_{\mathbf{n}'} W_{\mathbf{n}\to\mathbf{n}'}$.
2. Determine the (exponentially distributed) time step $\tau = -\ln \mathtt{rnd}(0,1]/W_{\mathbf{n}}$.
3. Choose a new state $\mathbf{n}'$ with probability $p_{\mathbf{n}}(\mathbf{n}') = W_{\mathbf{n}\to\mathbf{n}'}/W_{\mathbf{n}}$.

The efficiency of this algorithm hinges on the efficient implementation of step 3, i.e. the selection of the new state $\mathbf{n}'$ from the probabililty distribution $p_{\mathbf{n}}(\mathbf{n}')$ depending on the current state $\mathbf{n}$. For an m-dimensional problem, the number of states $\mathbf{n}'$ accessible from $\mathbf{n}$ ($W_{\mathbf{n}\to\mathbf{n}'} > 0$) will be $M \sim \mathrm{O}(m)$. To see this, consider a reaction-diffusion system with two particle species A and B and one chemical reaction $\mathrm{A} + \mathrm{B} \to \emptyset$, modelled on a grid of $L \times L$ cells. Then, the state space is $m = 2L^2$-dimensional and as long as all cells contain at least one A and one B particle each, at every time step one out of $M = (4+4+1)\cdot L^2$ possible events has to be chosen: From any one of the L^2 cells, either an A or a B particle diffuses to any one of the four nearest neighbors or a reaction occurs in it. Therefore, linear selection schemes requiring an effort of $\mathrm{O}(M)$ additions per time step are utterly unsuitable for large systems, as are rejection methods, which are efficient only if $p_{\mathbf{n}}(\mathbf{n}')$ is restricted to a small inter-

val [11]. Methods employing binary trees for step 3 still require an effort of $O(\log_2 M)$ per time step [12].

A sophisticated method using a logarithmic classification scheme for step 3 has been introduced a few years ago [13, 14]. This method yields a computational effort for step 3 independent of M, i.e. independent of the size of the system, provided the transition rates $W_{\mathbf{n}\to\mathbf{n}'}$ are independent of M (for a more detailed discussion see [11]). The algorithm involves some overhead though, and for systems spanning a very large range of transition rates ($> O(10^{14})$), the algorithm slows down slightly due to computational precautions required to avoid round-off errors.

The discrete class algorithm is similar in spirit to the logarithmic classes, but specifically aimed at systems with a reasonably small, discrete set of transition rates, i.e.

$$W_{\mathbf{n}\to\mathbf{n}'} \in \{r_1, \ldots, r_K\} , \quad K \lesssim 50 \ .$$

While this may seem a to be a strong restriction at first, models of epitaxial growth [15] fulfill these restrictions as well as some neuronal models.

The DCA implements step 3 of the minimal process method as follows. Each possible transition event $\mathbf{n} \to \mathbf{n}'$ is assigned to one of K classes according to its rate

$$D_\nu = \{\mathbf{n} \to \mathbf{n}' \,|\, W_{\mathbf{n}\to\mathbf{n}'} = r_\nu\} , \quad \nu \in 1, \ldots, K \quad .$$

Thus the rate of events in class D_ν is given by

$$R_\nu = \sum_{\mathbf{n}'\in D_\nu} W_{\mathbf{n}\to\mathbf{n}'} = \|D_\nu\| \, r_\nu \ ,$$

where $\|D_\nu\|$ is the number of events in D_ν, while the total rate of events is given by

$$W_{\mathbf{n}} = \sum_{\nu=1}^{K} R_\nu \ .$$

Furthermore, within a class each event occurs with equal probability. The selection step 3 is thus split into two substeps:

3a. Choose a class D_ν with probability $R_\nu / W_{\mathbf{n}}$ by linear selection, i.e. for a $\rho = \mathtt{rnd}[\,0, W_{\mathbf{n}})$ select that class ν for which

$$\sum_{i=1}^{\nu-1} R_i \le \rho < \sum_{i=1}^{\nu} R_i \ .$$

3b. Select the new state $\mathbf{n}'$ from class D_ν at random.

The linear selection in step 3a requires drawing a single, uniformly distributed random number and $O(K)$ additions, independent of the size of the system, while step 3b requires drawing another uniformly distributed random number. Thus, the efficiency of the selection algorithm does not depend on the size of the system under study. As the number of classes is assumed to be small, the total rate $W_{\mathbf{n}}$ can be calculated at every step, keeping numeric inaccuracies to a minimum.

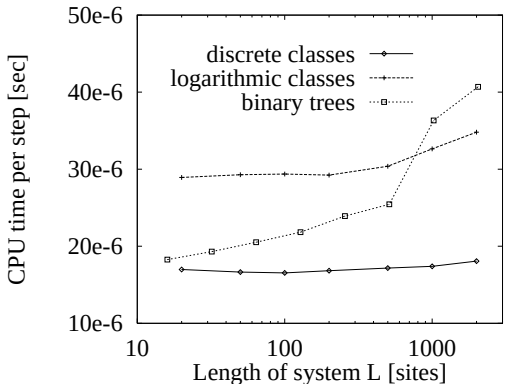

Figure 1: CPU time required per time step for different simulation algorithms. All simulations were performed on an SGI Indy workstation with 192 megabyte of memory.

III. Comparison with other methods

To demonstrate the performance of the DCA compared to the logarithmic classes [5, 11] and other state-of-the art methods such as binary trees [12], let us consider a simple model of epitaxial growth based on [6]:

- the substrate is an $L \times L$ lattice;
- each lattice site is occupied by one adatom or none;
- in an initial phase, $N < L \times L$ adatoms are deposited, which cannot evaporate;
- an adatom with all four next neighbor sites occupied cannot move;
- all other adatoms diffuse to next neighbor sites with rates

$$w_n = \tfrac{2k_BT}{h} e^{-E_S/k_BT} \, e^{-nE_N/k_BT} \ .$$

Here, n is the number of occupied next-neighbor sites, h, k_B are Planck's and Boltzmann's constants, T is temperature and E_S, E_N are material-dependent energies characterizing adatom-substrate and adatom-adatom interactions, respectively; both are on the order of 1eV. Thus, after the deposition phase, the system is determined by just five different rates which

Figure 2: Propagation of waves on the retina. Grey level indicates time since last firing, black being most recent.

are at $T = 600\text{K}$: $w_0 = 3.0 \cdot 10^2$, $w_1 = 1.2 \cdot 10^{-6}$, $w_2 = 4.8 \cdot 10^{-15}$, $w_3 = 1.9 \cdot 10^{-23}$ and $w_4 \equiv 0$.

Figure 1 shows the CPU time required per step for the simulation of this model for different lattice sizes using the discrete class, the logarithmic class and the binary tree algorithm. This demonstrates clearly the superiority of the discrete classes to the other algorithms in terms of absolute times as well as the size-independence of efficiency. The minuscule increase in CPU time for the DCA at very large lattices is due to cache effects, i.e. shortcomings of the hardware; for a detailed discussion, see [16].

IV. A Neuronal Model

A crucial problem in modeling the signal processing by neuronal networks is the enormous number of neurons involved even in simple tasks. Typically, though, only a small number of neurons will respond to any one stimulus presented e.g. to the eye or the ear. The DCA is well suited for the simulation of such largely "dormant" systems, since it automatically "focuses" on active regions of the system under study.

To demonstrate the applicability of the DCA to neuronal studies, we have formulated the following model, which is essentially a simplified type of Stein's model neuron [17].

- At $t = 0$, each neuron j has the resting membrane potential $v_j(0) = 0$.
- The membrane potential v_j is governed by the equation $\mathrm{d}v_j/\mathrm{d}t = f_s^{(j)}(t) + f_p^{(j)}(t)$.
- All input $f_s^{(j)}$, $f_p^{(j)}$ consists of delta-spikes, i.e. an input event at time T corresponds to the transition $v_j(T-) \to v_j(T+) = v_j(T-) + 1$.
- $f_p^{(j)}(t)$ is Poissonian noise with rate $1/\tau_p$.
- $f_s^{(j)}(t)$ is synaptic input from other neurons.
- As the potential reaches a threshold, $v_j(t) = \Theta$, neuron j fires a spike after an average waiting time τ_f, which is transmitted to all k_j neurons receiving input from j; neuron j is reset to an absolute refractory state.
- All input is ignored in the refractory state and the neuron returns to the resting state $v_j(t) = 0$ with rate $1/\tau_r$.

This model is obviously very well suited for the DCA, since it is governed by only three different rates: $1/\tau_p$, $1/\tau_f$ and $1/\tau_r$. In order to model spontaneous retinal waves as have been observed in newborn ferrets [18], we have simulated a grid of neurons with strongly localized synaptic connections. The network studied had 512×512 neurons and some $6.7 \cdot 10^6$ synapses, the threshold was set to $\Theta = 7$. The simulation was stopped after 1.5 million spikes had been generated, which required only 80 seconds of CPU time on an SGI Indy workstation. Figure 2 shows a typical state of activity.

V. Conclusions

The DCA algorithm presented here is a powerful tool for the study of a large class of stochastic systems and should foster research in these fields. The extension of the epitaxial model towards more complex phenomena is straightforward.

The neuron model presented above is most likely too simplistic to further our understanding of real neuronal systems, but we are presently working on a faithful implementation of Stein's model. Preliminary results indicate that leak currents and inhibitory inputs can be included. Inclusion of arbitrary synaptic weights, though, might necessitate recourse to the more generally applicable logarithmic class algorithm.

Note that the effective implementation of the DCA requires sophisticated data structures, similar to those described in [11]. Source code that can be integrated in simulation software via an easy to use interface is available from the authors upon request.

Acknowledgement

Hans E. Plesser acknowledges partial financial support by Studienstiftung des deutschen Volkes.

References

[1] A. Einstein, *Untersuchungen über die Theorie der 'Brownschen Bewegung'*, R. Fürth ed., Akademische Verlagsanstalt, Leipzig, 1922

[2] M. v. Smoluchowski, *Abhandlungen über die Brownsche Bewegung und verwandte Erscheinungen*, R. Fürth ed., Akademische Verlagsanstalt, Leizig, 1923

[3] C. W. Gardiner, *Handbook of Stochastic Methods for Physics, Chemistry and the Natural Sciences*, 2nd ed., Springer, Berlin, 1985

[4] N. G. van Kampen, *Stochastic Processes in Physics and Chemistry*, 2nd ed., North–Holland, Amsterdam, 1992

[5] D. Wendt, T. Fricke, J. Schnakenberg, Z. Phys. B **96**:541 (1995)

[6] C. Ratsch et al., Phys. Rev. Lett. **72**:3194 (1993)

[7] R. Engbert, F. R. Drepper, Chaos, Solitons & Fractals **4**:1147 (1994)

[8] T. Ohira, J. D. Cowan, Phys. Rev. E **48**:2259 (1993)

[9] S. Karlin, H. M. Taylor, *A First Course in Stochastic Processes*, 2nd ed., Academic Press, San Diego, 1975

[10] D. T. Gillespie, J. Comp. Phys. **22**:403 (1976)

[11] T. Fricke, D. Wendt, Int. J. Mod. Phys. C **6**:277 (1995)

[12] J. L. Blue, I. Beichl, F. Sullivan, Phys. Rev. E **51**:R867 (1995)

[13] T. Fricke, J. Schnakenberg, Z. Phys. B **83**:277 (1991)

[14] Y. Matias, J. S. Vitter, W.–C. Ni, Proceedings of the 4th Annual SIAM/ACM Symposium on Discrete Algorithms (SODA '93), Austin, TX, 1993

[15] P. A. Maksym, Semicond. Sci. Technol. **3**:594 (1988)

[16] H. E. Plesser, Untersuchungen über die Anwendbarkeit stochastischer Verfahren zur Lösung partieller Differentialgleichungen, diploma thesis, RWTH Aachen, 1995

[17] H. C. Tuckwell, *Stochastic Processes in the Neurosciences*, SIAM, Philadelphia, 1989

[18] M. B. Feller et al., Science **272**, 1182 (1996)