Fast simulations of stochastic dynamical systems

Juan A. Acebrón [a] , and Renato Spigler [b]

[a] *Departamento de Automática, Escuela Politécnica, Universidad de Alcalá,*
Crta. Madrid-Barcelona km. 31.600, 28871 Alcalá de Henares, Spain

[b] *Dipartimento di Matematica, Università di "Roma Tre", Largo S.L. Murialdo 1,*
00146 Rome, Italy

Abstract

A number of features relevant to stochastic dynamical systems, such as moments of solutions and first passage times, can be efficiently computed resorting to sequences of quasi-random numbers (low-discrepancy sequences). This method represents an alternative choice to the use of pseudorandom numbers, provided that a special care is paid to keep sufficiently uncorrelated as well as uniform the former sequences.

Key words: Stochastic differential equations, quasi-random numbers, quasi-Monte Carlo methods, first passage time, colored noise stochastic process
PACS: 65C05, 65C20, 65C30

---

1 Introduction

Analyzing random phenomena, for instance the effect of random perturbations on otherwise deterministic dynamical systems, is of paramount importance in a variety of problems, being rather ubiquitous. Within such difficult problems, an explicit analytical solution can be hardly found, thus pushing to the search for asymptotic or numerical methods. Effective numerical methods for integrating stochastic dynamical systems (that is systems of stochastic differential equations, SDEs) are based, as a rule, on the generation of suitable sequences of random numbers, which are often called "pseudorandom", to stress that ideal randomness cannot be reproduced on real computers. The pseudorandom numbers mimic the basic property of the random numbers of

---

Email addresses: `juan.acebron@uah.es` (Juan A. Acebrón),
`spigler@mat.uniroma3.it` (Renato Spigler).

Preprint submitted to Elsevier Science *17 December 2004*

being uncorrelated as well as statistically uniformly distributed. The latter feature is however satisfied only when infinitely many numbers are involved. Indeed, there is a nonzero probability to obtain any given number more than once. Moreover, the numerical error made using a sample of N pseudorandom numbers is of order of $N^{-1/2}$, which implies a rather slow convergence when $N \to \infty$.

An alternative choice to the pseudorandom numbers does exist, and is represented by the so-called "quasi-random" numbers. Sequences of such numbers, also called low-discrepancy sequences, were first introduced by J.G. van der Corput in 1935, popularized by H. Niederreiter [1], and exploited in some applications in physics and in financial mathematics, see [2,3]. These sequences are built using strictly deterministic uniformly distributed numbers, which are strongly correlated. In spite of such a correlation, these numbers have been successfully used in some problems, instead of their pseudorandom analog. When this can be done, the advantage is great, in that the inherent numerical error is in this case deterministic and of order of $N^{-1} \log^{d-1} N$, where d is the dimension of the random variables being simulated. In practice, it is observed that destroying somehow the correlation among the quasi-random numbers, the uniformity of their distribution is also affected. On the other hand, reducing such a correlation is mandatory in order to reproduce precisely the truly random nature of the phenomena being investigated.

2 Generalities on solving SDEs with quasi-random numbers

Consider, for the purpose of illustration, the van der Corput sequence. It works as follows: For every integer number out from a given set, write it in binary form, then write down its digits in the reverse order, and take the fractional binary number whose mantissa is given by such last sequence of binary digits. Finally, transform back the latter into a number in base 10. This will be a number between 0 and 1, and it can be proved that all numbers obtained in this way are uniformly distributed on the interval $[0, 1]$. For instance, the decimal number 1 is represented in base 2 by 01, reverting it we obtain 10, then 0.10, which in base 10 is 0.5. Similarly, from the integers 2 and 3 we obtain, symbolically: $2 \to 10 \to 01 \to 0.01 \to 0.25$; $3 \to 11 \to 11 \to 0.11 \to 0.75$. Therefore, the van der Corput sequence constructed from the integers $1, 2, 3$ is given by $0.5, 0.25, 0.75$. Any other numerical basis instead of 2 can be used provided that such a basis is a prime number. The ensuing sequences are called Halton sequences. Other sequences of quasi-random numbers have been derived in the literature, following different approaches.

Despite the success achieved by quasi-random numbers in evaluating multidimensional integrals, it seemed for some time that they would fail in solving numerically stochastic dynamical systems, such as

$$\dot{x} = f(x) + g(x)\xi(t), \quad x = x_0, \tag{1}$$

where $\xi(t)$ represents a gaussian white noise process,

$$< \xi >= 0, \quad < \xi(t)\xi(t') >= \delta(t - t'), \tag{2}$$

see [4]. Such a problem can be numerically solved upon time discretization by a variety of schemes [5]. In this paper, two such schemes have been used. The simplest one is the Euler scheme, given by [5,6],

$$x_{i+1} = x_i + f(x_i)\Delta t + \sqrt{\Delta t}\, g(x_i)\zeta_i, \tag{3}$$

where x_i represents an approximation to $x(t_i)$, $t_i = i\Delta t$, $i = 1, 2, \ldots, M$, which turns out to be of order 1/2 as a "strong method" (i.e., when one computes paths), and of order 1 as a "weak method" (when computing moments). Higher order methods do exist, for instance

$$\begin{aligned} x_{i+1} = x_i + f\Delta t + \sqrt{\Delta t}\, g\zeta_i + \frac{1}{2} g\, g'(\zeta_i^2 - 1)\Delta t \\ + \frac{1}{2}[f'\, g - f\, g'](\zeta_i + \frac{\eta_i}{\sqrt{3}})\Delta t^{3/2} \\ + \frac{1}{2}(f\, f' + \frac{1}{2} f'' g^2)\Delta t^2 + (f g' + \frac{1}{2} g'' g^2)\Delta t^{3/2}\zeta_i, \end{aligned} \tag{4}$$

where all quantities f, g, and their derivatives are evaluated at x_i, which is based on the stochastic Taylor formula [5]. It can be shown that, computing moments through such a formula, the error is of order 2. For this reason, it is often referred to as an "order 2.0 weak Taylor scheme", see [5]. In both equations, the ζ_i's and the η_i's denote independent random numbers, distributed according to a gaussian distribution with zero mean and unit variance. The moments of the solution to Eq. (1) can be obtained evaluating N times the sequence x_i for $i = 1, 2, \ldots, M$, and then taking averages over the ensemble of size N.

Clearly, in practice the ζ_i's and the η_i's are obtained generating pseudorandom or quasi-random sequences. Despite the negative results reported in [4], we can show that such a numerical integration can be accomplished correctly by a careful implementation of quasi-random sequences, and actually this can be done very efficiently.

The main problem encountered using sequences of quasi-random numbers being the correlation among them, it is natural to try to destroy it. This goal can be achieved "scrambling" somehow these numbers. A possible way to do it is provided by a reordering strategy, which consists of relabeling at each time step all positions x_i inside the ensemble according to their amplitude, e.g., by decreasing or increasing amplitudes. Such a procedure was earlier proposed in [7,8] to solve diffusion problems by particle simulations. In this paper, we adopted a similar strategy to solve numerically rather general systems of SDEs, hence generating paths of rather general stochastic processes.

While there are many ways to scramble a given sequence, reordering seems to be the optimal way. This is because reordering best respects uniformity of the sequence, and moreover it is algorithmically simple. It is easy to see that reordering amounts to premultiplying by a permutation matrix the vector containing all positions at each time step.

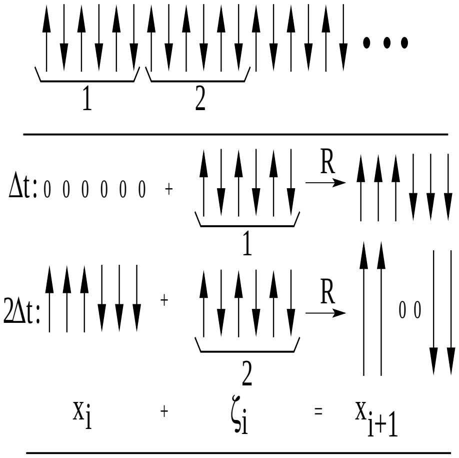

Fig. 1. Sketchy diagram illustrating the effect of random walk discretization of the one-dimensional brownian motion using the van der Corput sequence. Here the sample size is $N = 6$.

Numerical experiments show that often reordering is very effective, yet sometimes it fails. The point is that, while it seems to be necessary, it may not be enough. In fact, certain quasi-periodicities affect all sequences of quasi-random numbers, thus introducing unwanted regularities. Such regularities represent correlations among the numbers, whose overall effect is a departure from the truly random behavior that we are supposed to mimic. The periodicity inherent to all sequences of pseudorandom numbers is also unwanted, but others than in case of quasi-random numbers, such a periodicity is usually so large that it cannot be perceived in practice, see [9].

To make it clear what we mean by periodic or quasi-periodic behavior, consider the one-dimensional brownian motion equation, and try to solve it numerically by a random walk approximation, using the van der Corput sequence. The first few numbers are: $0.5, 0.25, 0.75, 0.125, 0.625, 0.375, \ldots$ To simulate one-dimensional random walks, what only matters is knowing whether one should move either to the left or to the right. This should occur with the same probability, and thus every quasi-random number of the sequence, lying in $[0, 0.5)$ implies movement to the left, while every number in $[0.5, 1)$ does it to the right.

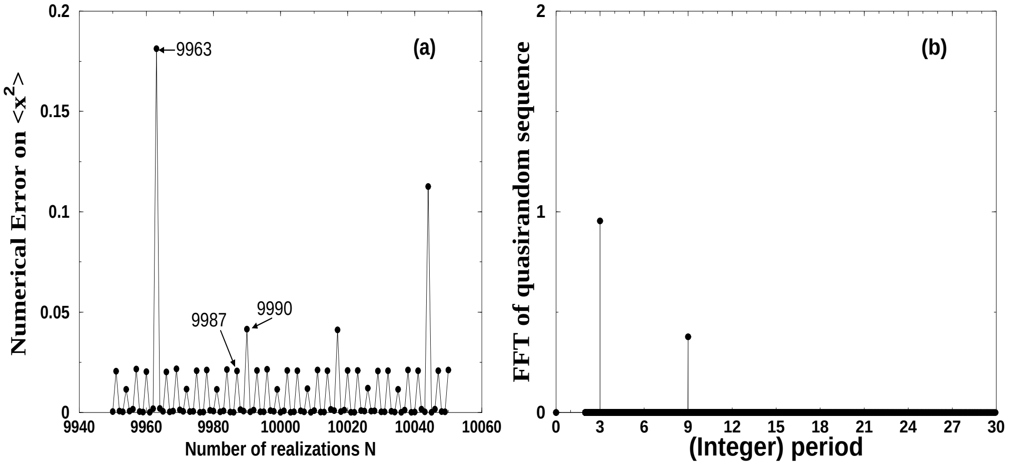

Fig. 2. (a) Numerical error made in computing the second moment of the 1D brownian motion at some fixed time. The brownian motion was approximated by a random walk; (b) peaks in the FFT of the Halton quasi-random number sequence with base $p = 3$. Parameters are $N = 1997$, $\Delta t = 0.001$.

On the top part of Fig. 1, a sequence of arrows denotes a sequence of left- and right-motions associated to the aforementioned random walk (those pointing upward, e.g., denote displacement to the right). For simplicity, suppose that only $N = 6$ realizations of the paths have been used. Then, in this case, the sequence of arrows should be used in sections of six, in that we need the first six arrows to assign the direction of the random walks at the first time step. The first six arrows are added to the initial positions (for simplicity taken all equal to zero). Reordering the so-obtained sequence according to increasing size, then yields the three first arrows upward and the last three downward (see Fig. 1 at time Δt). At the next time step (time $2\Delta t$), the latter sequence is subject again to random walk motion, requiring the next available string in the quasi-random number sequence. The output is then reordered as above. Note that, proceeding in this way, the first upward arrow (first realization) increases in size monotonically, while the last one similarly decreases. Therefore, this procedure cannot lead to imitate any random walk. Clearly, it is the evenness of the realization number, N, that spoils the procedure. In other words, a keen choice would require that N be not divisible by 2, which is the prime number underlying the van der Corput sequence. Similarly, when using other Halton-type sequences, characterized by the prime number p, one should avoid realization samples whose size is divisible by p. In Fig. 2(a), the numerical error made computing the second moment of the 1D brownian motion at some fixed time is shown, as a function of the number of realizations. Here a random walk approximation of the brownian motion has been made. To this purpose

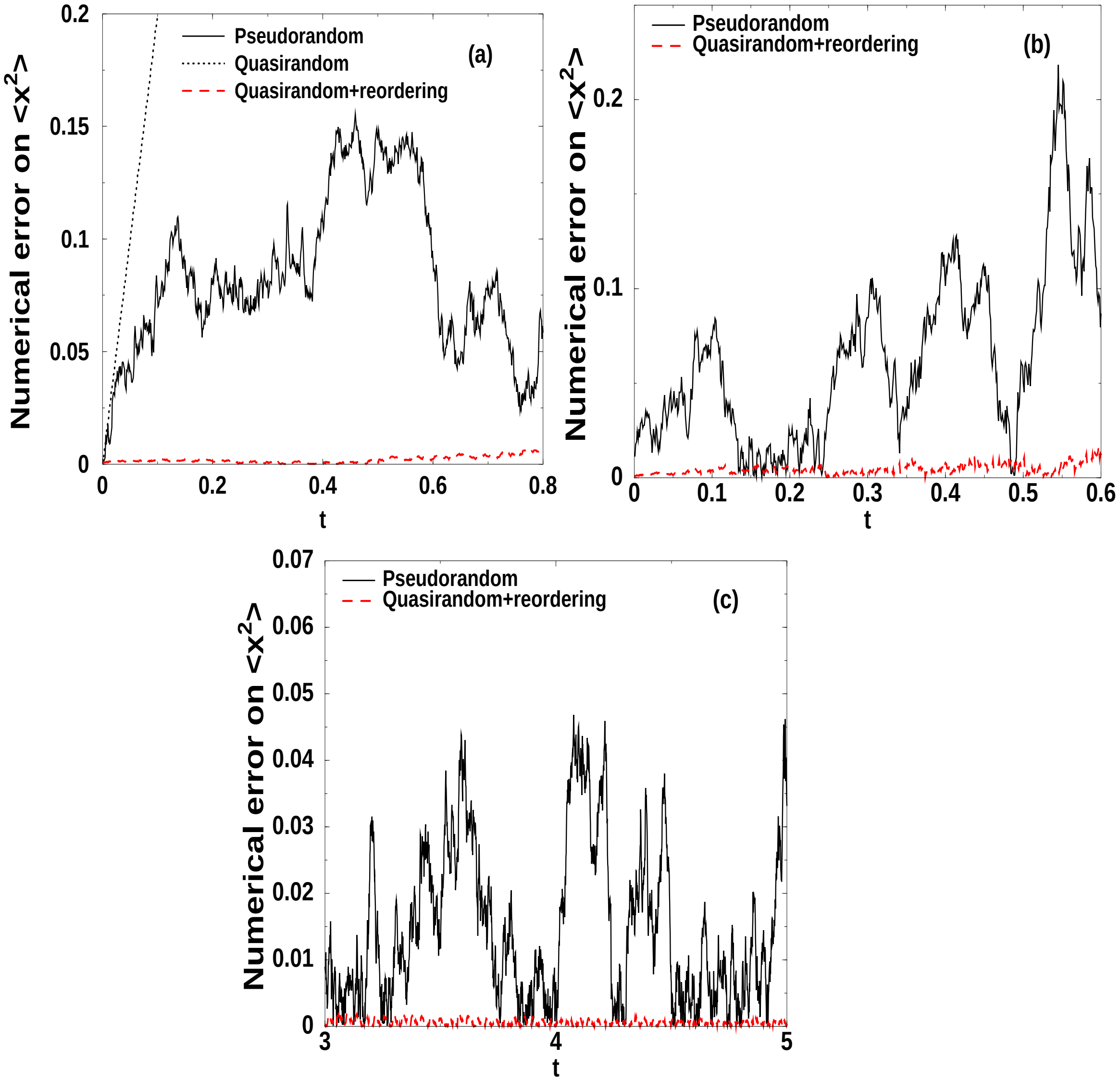

Fig. 3. One-dimensional problems: (a) brownian motion, (b) multiplicative noise, (c) bistable Duffing potential. Parameters are $N = 1997$, and $\Delta t = 0.001$. The numerical scheme adopted was Euler's.

we used the Halton sequence with $p = 3$ as the van der Corput sequence was used in Fig. 1, that is recording upward and downward arrows accordingly to the previous rule. Note that a number of peaks appear correspondingly to realization numbers N divisible by 3, and higher peaks appear when N is divisible by powers of 3. In Fig. 2(b), the FFT of such a sequence is shown. In this case the pattern in Fig. 1 becomes a little more involved, the sequence being quasi-periodic rather than merely periodic. However, the presence of sub-periods implies the same drawbacks observed using the van der Corput sequence. In Fig. 2(b), only the harmonics in the FFT are shown which occur at integer abscissas (periods), in that only these matter, in view of the divisibility of N.

For the purpose of illustration, consider the examples

$$\begin{aligned}&(a)\,\dot{x} = \xi, \quad < x^2 >= 4 + t\\&(b)\,\dot{x} = \varepsilon x + x\,\xi, \quad < x^2 >= 4\,e^{(1+2\varepsilon)t}\\&(c)\,\dot{x} = x - x^3 + \xi, \quad < x^2 >\approx 1.0418, \quad t \to \infty\end{aligned} \tag{5}$$

all with $x_0 = 2$, which correspond, respectively, to brownian motion, a case of multiplicative noise dynamics, and motion in a bistable Duffing potential subject to random perturbations. Here the Euler scheme was adopted to simulate the paths. In order to generate gaussian random numbers, a pseudorandom or a quasi-random number is picked up from a uniform distribution on $[0, 1]$, and then the inverse error function is applied to it. Other methods for generating gaussian numbers do exist, and have been succesfully used with pseudorandom number sequences; see for instance the well known Box-Muller method [5]. However, such a method neither shows any advantage nor performs better than the inverse error function method when quasi-random sequences are used [7].

In Fig. 3, the numerical error made in computing the second moment, $< x^2 >$, is plotted as a function of time. Such an error can be obtained since in cases (a) and (b) the analytical solution of it is known, while in case (c) only an asymptotic representation is known. For this reason only relatively large times were considered in Fig. 3(c). In Fig. 3(a), results obtained by means of quasi-random sequences without reordering are also shown, and they blow up rapidly, as expected.

Table 1
CPU time in seconds and relative error for example (a)

	Quasi-random		**Pseudorandom**	
N	Time	Error	Time	$\overline{Error^2}^{1/2}$
999	0.60	2.64×10^{-4}	0.44	3.13×10^{-2}
9999	6.97	4.51×10^{-5}	2.18	9.42×10^{-3}
99999	87.02	6.79×10^{-6}	32.10	2.76×10^{-3}

In Tables 1, 2, and 3, the second and fourth columns labeled by “Time” show the overall computational time (in seconds) spent in solving the stochastic differential equations given in examples (a), (b), and (c), respectively. In the second column, this has been done using quasi-random sequences, while in the fourth column pseudorandom numbers were used in a single run. In the third column, the relative error made using quasi-random numbers is shown,

Table 2
CPU time in seconds and relative error for example (b) with $\varepsilon = -1$

N	**Quasi-random**		**Pseudorandom**	
	Time	Error	Time	$\overline{Error^2}^{1/2}$
999	0.60	2.33×10^{-3}	0.44	5.79×10^{-2}
9999	7.00	9.40×10^{-4}	2.16	2.69×10^{-2}
99999	84.55	6.46×10^{-4}	33.89	7.06×10^{-3}

Table 3
CPU time in seconds and relative error for example (c)

N	**Quasi-random**		**Pseudorandom**	
	Time	Error	Time	$\overline{Error^2}^{1/2}$
999	6.02	5.27×10^{-3}	2.26	3.39×10^{-2}
9999	71.03	5.31×10^{-4}	32.01	8.94×10^{-3}
99999	860.33	1.07×10^{-4}	321.18	3.02×10^{-3}

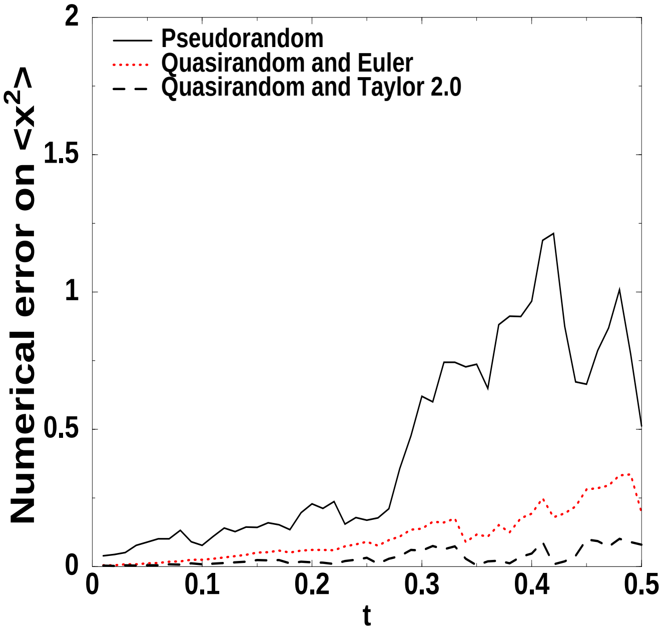

Fig. 4. Comparison between the error attained solving numerically example (b) with $\varepsilon = 1$, by means of an Euler scheme and a Taylor order 2.0 weak scheme. In both cases, quasi-random sequences of numbers with reordering were used, and pseudorandom numbers for the Euler case. Parameters are as in the previous figure, except now with $\Delta t = 0.01$.

while in the fifth column the root mean square of the relative error is shown when pseudorandom numbers were used. Such a root mean square error has been obtained running 100 different batches each consisting of N realizations.

Table 4
Relative error for example (b) with $\varepsilon = 1$ and two different numerical schemes

	Quasi-random		**Pseudorandom**	
N	Euler	Taylor order 2.0	Euler	Taylor order 2.0
999	1.75×10^{-2}	4.13×10^{-3}	6.68×10^{-2}	5.48×10^{-2}
9999	1.17×10^{-2}	1.53×10^{-3}	2.97×10^{-2}	1.76×10^{-2}
99999	1.24×10^{-2}	6.53×10^{-4}	2.01×10^{-2}	7.78×10^{-3}

Three different numbers of realizations have been used, labeled by N in the first column. Note that, for instance in Table 1, using pseudorandom numbers the CPU time necessary to achieve the smaller relative error (order 10^{-4}) is almost 53 times larger than that made using quasi-random numbers. Similar results can be seen in Table 2, and 3. Therefore, the advantage of using quasi-random over pseudorandom sequences is clear, results being spectacular in these cases. The differences in CPU time, when quasi-random and pseudorandom sequences were used, for a fixed N, are merely due to the reordering procedure. The latter is mandatory for a successful implementation of quasi-random sequences.

Quasi-random sequences of numbers can also be implemented sucessfully in numerical schemes of higher order. To illustrate the performance achieved doing so, we have solved numerically example (b) with $\varepsilon = 1$, for a much larger time-step, i.e. $\Delta t = 0.01$, and two different numerical schemes, the Euler method, eq. (3), and the order 2.0 Taylor method, eq. (4). In fig. 4, a comparison is made between the results obtained by such methods, solving numerically example (b) by means of pseudorandom and quasi-random sequences. The order 2.0 Taylor scheme in (4) involves two gaussian random numbers, which are obtained through two independent sequences of quasi-random numbers. These are picked up from Halton sequences corresponding to two different numerical bases, 2 and 3 respectively. In Table 4, the second and fourth columns display the relative error and the root mean square error made solving example (b) by the Euler method with quasi-random and with pseudorandom sequences, respectively. The third and fifth columns show the results of the numerical solution obtained by an order 2.0 Taylor scheme. Note that the higher-order method does indeed perform better than the Euler method, the larger the number of realizations, N, the better. In fact, two sources of error should be considered. The first is the statistical error, which is related to the finite size of the sample, N, and turns to be of order $N^{-1/2}$ for pseudorandom numbers, and of order $N^{-1} \log^{d-1} N$ (for a suitable "dimension" d) for quasi-random sequences [2]. The second is the truncation error, which can be reduced upon adopting higher-order schemes. Only when N is sufficiently large the effect of taking higher-order methods can be observed. Note that when $N = 99999$ and using quasi-random sequences, the error is of order 10^{-2} with the Euler

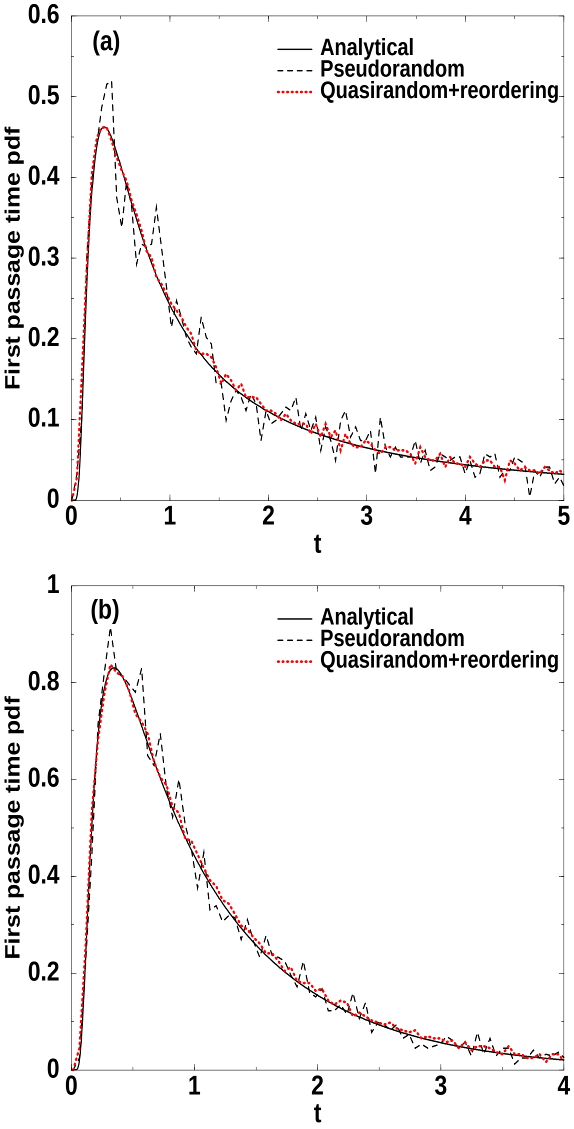

Fig. 5. First passage time probability density function: (a) brownian motion; (b) Ornstein-Uhlenbeck process. Parameters are as in Fig. 3.

method and of order 10^{-4} with the Taylor method, as expected.

2.2 Computing first passage times

Another important issue concerning physical systems affected by random perturbations is the evaluation of first passage times [10]. In Fig. 5, the probability density function of the first passage time is plotted as a function of time, correspondingly to the brownian motion (example (a)) and to the Ornstein-Uhlenbeck dynamical system

$$\dot{x} = -x + 1 + \xi, \quad x_0 = 0. \tag{6}$$

To obtain such plots, the equations above have been solved numerically for a number of paths, and the first time when each path crossed the level $x = 1$ has been recorded. Then, the probability density has been obtained as an histogram. Again, the analytical solutions were available in these cases, see [11]. In solving such problems, a rather small time step has been chosen, because otherwise the numerical error made in capturing precisely the first exit time may dominate the overall computation, see [12–14]. Results obtained by quasi-random numbers outperform those achieved by pseudorandom numbers, unless times become too large. This deterioration is due to two reasons. One is that reordering becomes less and less effective, because the involved permutation matrices tend to the identity matrix. The other is that there are fewer realizations left that can be exploited when time becomes exceedingly large.

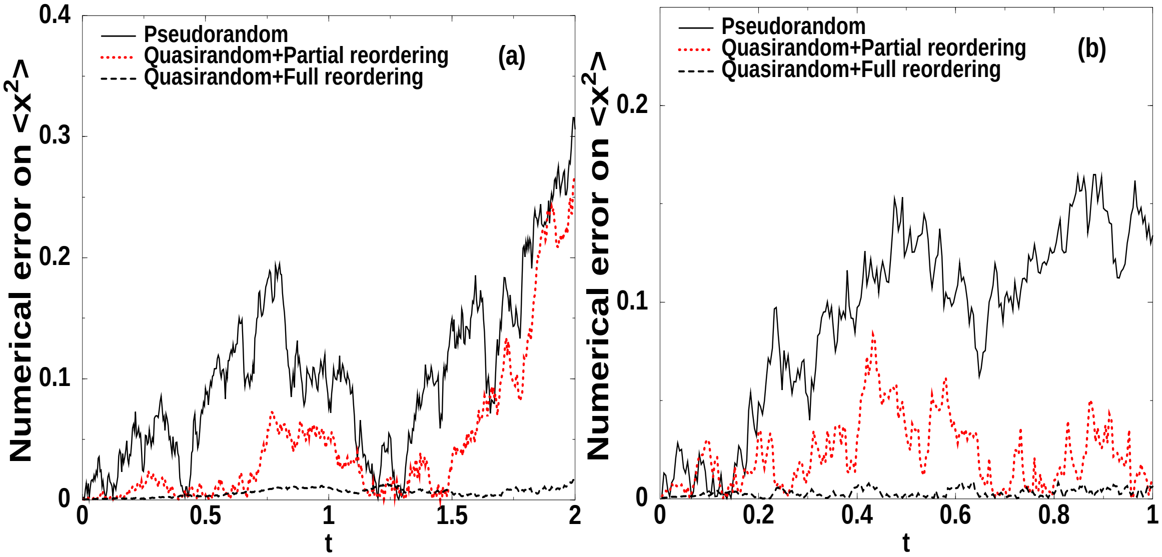

Fig. 6. Two-dimensional problems: (a) 2D brownian motion; (b) 2D multiplicative noise. Parameters are as in Fig. 3.

2.3 Multi-dimensional SDEs

In principle, quasi-random numbers sequences can be used to integrate numerically multidimensional stochastic dynamics as well. Consider two examples given by the two-dimensional brownian motion and by a two-dimensional multiplicative noise system,

$$(a)\begin{cases}\dot{x}=\xi_1\\ \dot{y}=\xi_2,\end{cases}\quad ,\quad <x^2>=<y^2>=4+t \tag{7}$$

$$(b)\begin{cases}\dot{x}=-x+y\,\xi_1\\ \dot{y}=-y+x\,\xi_2,\end{cases}\qquad <x^2>=<y^2>=4\,e^{-t} \tag{8}$$

with the initial values $x_0 = 2, y_0 = 2$. An effective implementation has been made picking up numbers from two Halton sequences characterized by two different bases of primes, e.g. $p = 2$ and $p = 3$, each for every dimension. Reordering here is not straightforward, in that the plane is not a totally ordered set as the line is. A reordering, first according to the distance from the origin, and then according to the angle, was accomplished in this case. This rearrangement can be termed a full reordering, while a partial reordering consists only acting on the distances from the origin. From Fig. 6, and Tables 4, and 5, the advantage of using quasi-random instead of pseudorandom numbers appears clearly, as well as that of using full reordering instead of partial reordering.

Table 5
CPU time in seconds and relative error for example (a)

	Quasi-random		**Pseudorandom**	
N	Time	Error	Time	$\overline{Error^2}^{1/2}$
1001	3.06	1.14×10^{-3}	2.28	2.87×10^{-2}
10001	36.54	2.45×10^{-4}	23.51	7.21×10^{-3}
100001	450.55	2.29×10^{-5}	255.06	2.88×10^{-3}

Table 6
CPU time in seconds and relative error for example (b)

	Quasi-random		**Pseudorandom**	
N	Time	Error	Time	$\overline{Error^2}^{1/2}$
1001	1.56	1.07×10^{-2}	1.16	7.30×10^{-2}
10001	18.37	1.60×10^{-3}	12.08	2.45×10^{-2}
100001	228.92	7.66×10^{-4}	123.10	7.15×10^{-3}

The entire approach proposed in this paper can be followed also to solve the more realistic problem of stochastic dynamical systems affected by colored noise, see [15], e.g. In fact, replacing white noise with colored noise in some system amounts to substituting the original system with a larger one only affected by white noise, see [16–18]. First passage times are often much longer in systems driven by colored noise, due to their inherent correlation time. Therefore, computing first passage times by quasi-random numbers may be less effective in this case.

3 Conclusion

Summarizing, we have shown that stochastic dynamical systems can be solved numerically adopting sequences of quasi-random numbers. This can done efficiently provided that some care is paid to their implementation. In particular, a reordering strategy should be used to destroy correlations which affect such sequences. It was also shown that such an action, in general, may not suffice, in view of periodicity or quasi-periodicity which characterizes such sequences. This difficulty can be overcome by a careful choice of the sample size being used in the numerical simulations.

Acknowledgment

This work has been supported, in part, by the GNFM of the Italian INdAM.

References

[1] Niederreiter, H., *Quasi-Monte Carlo methods and pseudo-random numbers*, Bull. Amer. Math. Soc. **84** (1978) 957–1041.

[2] Caflisch, R.E.,*Monte Carlo and quasi-Monte Carlo methods*, Acta Numerica (1998) 1–49.

[3] Caflisch, R.E., Morokoff, W., and Owen,A.B., *Valuation of mortgage-backed securities using Brownian bridges to reduce effective dimension*, J. Comput. Finance **1** (1997) 27–46.

[4] Hofmann, N., and Mathé, P., *On quasi–Monte Carlo simulation of stochastic differential equations*, Math. Comp. **66** (1997) 573–589.

[5] Kloeden, P.E., and Platen, E., *Numerical Solution of Stochastic Differential Equations*, (Springer, Berlin, 1992)

[6] Mannella, R., *Integration of stochastic differential equations on a computer*, Int. J. Mod. Phys. **C13** (2002) 1177–1194.

[7] Morokoff, W.J., and Caflisch, R.E., *A quasi-Monte Carlo approach to particle simulation on the heat equation*, SIAM J. Numer. Anal. **30** (1993) 1558–1573.

[8] Lecot, C., and El Khettabi, F., *Quasi-Monte Carlo simulation of diffusion*, J. of Complexity. **15** (1999) 342–359.

[9] Press, W.H., Flannery, B.P., Teukolsky, S.A., and Vetterling, W.T., *Numerical Recipes: The Art of Scientific Computing*, (Cambridge University Press, New York, 1986).

[10] Hänggi,P., Talkner, P., and Borkovec, M., *Reaction-rate theory - 50 years after Kramers*, Rev. Mod. Phys. **62** (1990) 251–341.

[11] Ricciardi, L., *Diffusion Proccesses and Related Topics in Biology*, Lecture Notes in Biomathematics **14**, (Springer, Berlin, 1977).

[12] Jansons, K.M., and Lythe, G.D., *Efficient numerical solution of stochastic differential equations using exponential timestepping*, J. Stat. Phys. **100** (2000) 1097–1109.

[13] Jansons, K.M., and Lythe, G.D., *Exponential timestepping with boundary test for stochastic differential equations* SIAM J. Sci. Comput. **24** (2003) 1809–1822.

[14] Mannella, R., *Absorbing boundaries and optimal stopping in a stochastic differential equation*, Phys. Lett. **A254** (1997) 257–262.

[15] Acebrón, J.A., and Spigler, R., *Second harmonics effects in random Duffing oscillators*, SIAM J. Appl. Math. submitted (2003).

[16] Mannella, R., and Palleschi, V., *Fast and precise algorithm for computer-simulation of stochastic differential equations*, Phys. Rev. **A40** (1989) 3381–3386.

[17] Honeycutt, R.L., *Stochastic Runge-Kutta algorithms. II.colored noise*, Phys. Rev. **A45** (1992) 604–610.

[18] Ramirez-Piscina, L., Sancho, J.M., de la Rubia, F.J., Lindenberg, K., and Tsironis, G.P., *1st-passage time in a bistable potential with colored noise*, Phys. Rev. **A40** (1989) 2120–2127.