Liet. matem. rink., T. 42, spec. nr., 331–334

Computer modeling of solutions to stochastic differential equations: weak approximations

Antanas LENKŠAS (VU)
e-mail: antanas.lenksas@maf.vu.lt

Introduction

Let us consider a one-dimensional stochastic differential equation

$$X_t = X_0 + \int_0^t b(X_s, s)\mathrm{d}s + \int_0^t \sigma(X_s, s)\mathrm{d}B_s, \quad t \geqslant 0, \tag{1}$$

where $B = \{B_t,\ t \geqslant 0\}$ is Brownian motion.

Unfortunately, stochastic differential equations which allow an explicit solution are rather an exception than the rule. Therefore, the need for numerical approximation methods is even bigger than in the case of deterministic differential equations. In this article we deal with the approximations of (1) in a weak sense (in a sense of distributions). That is, we are interested in X^h ($h > 0$), such that, for each fixed $t > 0$,

$$\left|\mathbf{E}f(X_t^h) - \mathbf{E}f(X_t)\right| \to 0, \quad \text{when } h \to 0,$$

for some (wide enough) class of functions $f : \mathbb{R} \to \mathbb{R}$.

For the models used in the field of economics or financial mathematics perhaps more important than a solution to stochastic differential equation is an expectation $\mathbf{E}f(X_t)$ for some function f (for example, $\mathbf{E}(X_t - K)^+$). In this case also one is rarely able to give an explicit formulae. And even then, such a formulae often involve some special functions that have to be approximated numerically.

Therefore, we will use a computer to find and visualize $\mathbf{E}f(X_t)$ for a given f and stochastic differential equation (1). It is a common practice to write a separate program (Pascal programming language here is the most popular choice among mathematicians for such a purpose) for each model involving (1). There is no need to say, that such an approach is time consuming and not convenient at all. We would suggest that more universal software offering a comfort of concentrating on mathematics rather than programming would be more helpful in modeling.

1. The program

First, we will write a program. A program that would allow to choose stochastic differential equation, approximation method, function f and many other parameters, for example, approximation step or number of simulated trajectories. It is clear that such a task is an example of interface between the theory of stochastic differential equations and computer

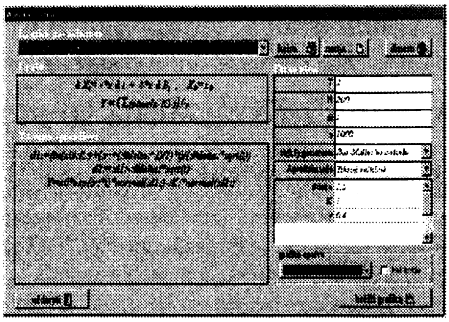

Fig. 1. Main program window.

science. Let us briefly look at the mathematical side. We implement three aproximations in our work: Euler, Milstein and Runge–Kutta kind of approximation suggested in (2). Let τ_n be a partition of $[0..T]$

$$0 = t_0 < t_1 < \ldots < t_{n-1} < t_n = T.$$

Denote, as usual, $\Delta t_i := t_i - t_{i-1}$ and $\Delta B_{t_i} := B_{t_i} - B_{t_{i-1}}$. Then we have following schemes of approximation

Euler scheme

$$X_0^h = x_0$$

$$\vdots$$

$$X_{t_n}^h = X_{t_{n-1}}^h + b(X_{t_{n-1}}^h, t_n)\Delta t_n + \sigma(X_{t_{n-1}}^h, t_n)\Delta B_{t_n}$$

Milstein scheme

$$X_0^h = x_0$$

$$\vdots$$

$$X_{t_n}^h = X_{t_{n-1}}^h + b(X_{t_{n-1}}^h, t_n)\Delta t_n + \sigma(X_{t_{n-1}}^h, t_n)\Delta B_{t_n} + \tfrac{1}{2} b(X_{t_{n-1}}^h, t_n) b'(X_{t_{n-1}}^h, t_n)(\Delta B_{t_n}^2 - \Delta t_n)$$

Values of X_t^h on the interval (t_{i-1}, t_i) are obtained by simple linear interpolation of the points $(t_{i-1}, X_{t_{i-1}}^h)$ and $(t_i, X_{t_i}^h)$.

The value of $\mathbf{E}f(X_{t_i})$ is evaluated by averaging the values $f(X_{t_i}^{h,j})$, where $X_{t_i}^{h,j}$, $j = 1, 2, \ldots, \gamma$ are γ simulated trajectories of the approximation of the solution to stochastic differential equation (1).

2. Modeling

Now it is simple to use the program (Fig. 1) for visualization of functionals of solutions to various stochastic differential equations.

As an example, we will choose a simple enough equation for which an exact solution is known. Function f should also be chosen so, that expectation $\mathbf{E}f(X_t)$ could be explicitly found. Therefore, consider an Ornstein–Ulhenbeck process, satisfying

$$X_t = -0.5X_t \mathrm{d}t + \mathrm{d}B_t,$$

and function $f(x) = x$. Knowing, that in this case, $\mathrm{E}X_t = X_0 \mathrm{e}^{-0.5t}$, we will try different schemes and values for the number of simulated trajectories γ and approximation step h, and compare these approximations with the exact solution. From the Figs. 2–3 it seems that both schemes behave almost identically and when γ equals 10000 appear to be quite indistinguishable visually from the exact solution.

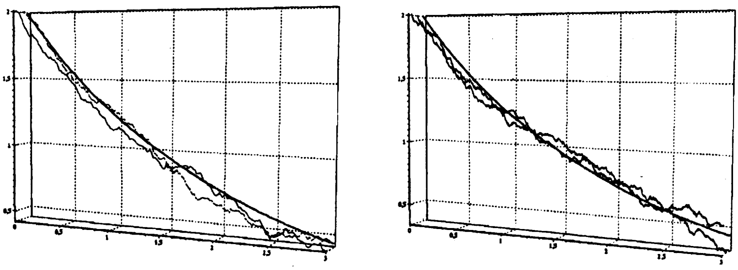

Fig. 2. A comparison of Euler and Milshtein approximations. Solid line: values of $X_0 \mathrm{e}^{-0.5t}$. On the left approximation step $h = 0.01$, on the right: $h = 0.001$. Number of simulated trajectories $\gamma = 100$.

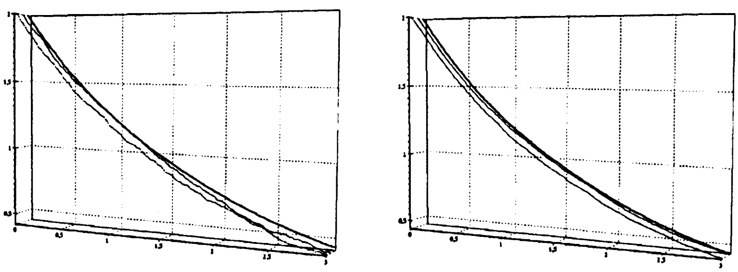

Fig. 3. A comparison of Euler and Milshtein approximations. Solid line: values of $X_0 \mathrm{e}^{-0.5t}$. Number of simulated trajectories $\gamma = 1000$ on the left and $\gamma = 10000$ on the right. Approximation step $h = 0.01$.

3. Conclusions

The user of the program has a freedom of choosing all the parameters, including stochastic differential equation itself and modeling weak approximations without even any knowledge of programing. However, a lot more there needs to be done. For example, though user (a mathematician) is free to choose all the parameters, he is still bounded to one-dimensional case. Therefore, two-dimensional case should be added as well. Also, more approximation schemes would be preferable. And, finally, the case of strong approximation would also be interesting to study.

References

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

[2] V. Mackevičius, J. Navikas, Second order weak Runge–Kutta type methods for Ito equations, *Mathematics and Computers in Simulation*, **57**, 29–34 (2001).

Stochastinių diferencialinių lygčių sprendimų modeliavimas kompiuteriu: silpnosios aproksimacijos

A. Lenkšas

Daugelyje ekonomikoje ar finansų matematikoje taikomų modelių yra naudojamos stochastinės diferencialinės lygtys. Kaip žinia, retai kada galima rasti jų sprendinį išreikštiniu pavidalu arba, jeigu ir galima, jo išraiškoje yra sudėtingos funkcijos, kurias šiaip ar taip reikia skaitiškai aproksimuoti. Todėl siūloma kompiuterinė programa, leidžianti matematikui neprogramuojant, „patogiai" modeliuoti įvairių stochastinių diferencialinių lygčių silpnąsias aproksimacijas, vizualiai stebėti, kaip, keičiant įvairius parametrus, keičiasi aproksimacijos pobūdis ir lyginti su tikruoju sprendiniu, jei tik šis yra žinomas.