The three point approximation method: A tool for comparisons.
User guide

March 1, 2024

Abstract

This is a application user's guide which was designed for comparison between the three point density expansion method which was introduced by the authors in [8] called the tree approximation method for one dimensional diffusions. This method is a simplification of a general algorithmic argument introduced in [7]. The results with slightly different proof methods have been already used in [4] and [6].

1 Set-up

Let X be an $\mathbb{R}$-valued diffusion process defined by an SDE

$$X_t = x + \int_0^t a(X_s)\,\mathrm{d}s + \int_0^t \sigma(X_s)\,\mathrm{d}W_s, t \in [0,T] \tag{1.1}$$

where W is an $\mathbb{R}$-valued Brownian motion and coefficients $a : \mathbb{R} \to \mathbb{R}, \sigma : \mathbb{R} \to \mathbb{R}$. The aim of this application is to study and compare three approximations of the solution X_t to the equation (1.1).

The first approximation implements the Euler scheme

$$X^{x,1}(t) = x + a(x)t + \sigma(x)W(t). \tag{1.2}$$

The second one implements the Milstein scheme

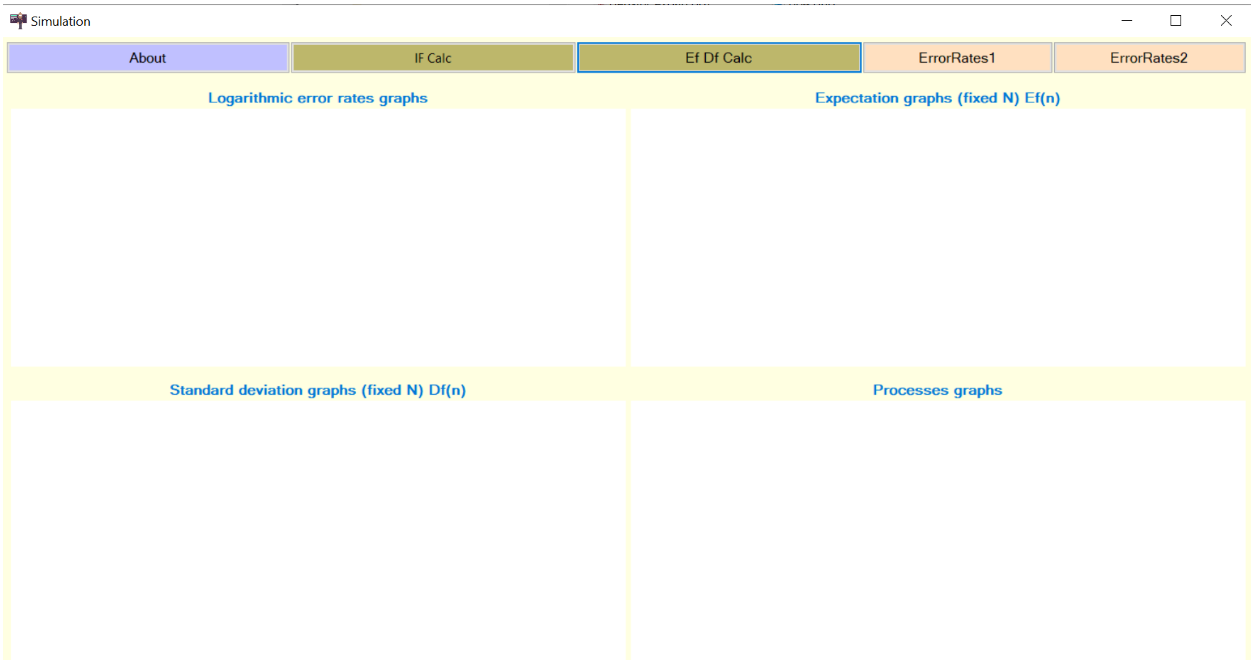
$$X^{x,2}(t) = x + a(x)t + \sigma(x)W(t) + \frac{1}{2}\sigma'(x)\sigma(x)(W(t)^2 - t) \tag{1.3}$$

And the third one

$$\begin{aligned} X^{x,3}(t) = x + a(x)t + \sigma(x)W(t) + \frac{1}{2}\sigma'(x)\sigma(x)(W^2(t) - t) \\ + \left(\frac{1}{2}a'(x)\sigma(x) + \frac{1}{2}\sigma'(x)a(x) + \frac{1}{4}\sigma''(x)\sigma(x)^2\right) tW_t \\ + \frac{1}{6}\left(\sigma''(x)\sigma(x)^2 + (\sigma'(x))^2\sigma(x)\right)(W_t^3 - 3tW_t). \end{aligned} \tag{1.4}$$

2 Goal window

The main window of the application consists of five buttons and four fields for displaying graphs.



The first button **"About"** opens the pdf viewer window, which displays the help file which is this user manual. Pressing the other four buttons opens separate dialog boxes with settings and output fields.

The second button is called **"IF Calc"**, which stands for integral functional calculation. In this option one can simulate an expectation that depends on the whole path trajectory. In order to test the performance of this approximation, we propose an alternative formula which depends on the marginal laws. More details can be found in Section 4. This allows you to evaluate the quality of all three approximation schemes (1.2) – (1.4) on path functionals. More details can be found in the section 4.

The name of the next button **"Ef Df Calc"** stands for the calculation of the mathematical expectation (letter **E**) and standard deviation (letter **D**) of the test function (**f**) taken at the approximations (1.2) – (1.4) of the trajectory of the process given by (1.1). This functionality is described in more detail in the section 3.

The next two buttons, **"ErrorRates1, ErrorRates2"** allows the user to calculate and visualize the weak errors of the approximation schemes (1.2) – (1.4). That is, the logarithm of the absolute value of the difference between the expectation of the test function at the approximation scheme and the theoretical expectation is proposed as a characteristic. The difference between these two buttons is that the first one assumes that the theoretical expectation is known and the user has to provide its value, while the second one assumes that a strong solution of the process (1.1) is known.

In both cases, the errors are used to compute and graph in logarithmic scale the corresponding regression lines. In the graph windows of the main window graph appears and their values appear in the accesory window. The first graph on the upper left side **"Logarithmic error rates graphs"** displays the dependence of the logarithmic error with respect to the logarithm of the number of points in the time interval.

The **"Expectation graphs (fixed N) Ef(n)"** window displays the graphs of the expectations of the test function evaleated at each of the three approximating processes. The number of partition intervals is plotted on the bottom axis using the logarithmic scale.

The lower left window **"Standard deviation graphs (fixed N) Df(n)"** plots the dependence of the logarithm of the standard deviation of the test function at the approximating processes on the logarithm of the number of time intervals used in the approximations.

The last window **"Processes graphs"** provides one of the trajectories of the approximating processes generated using the same Wiener process. Details on how to use the buttons **"ErrorRates1, ErrorRates2"** can be found in sections 5 and 6, respectively.

2.1 Common settings and general notations

Drift Drift coefficient in (1.1) (a).

Diffusion Diffusion coefficient in (1.1) (σ).

Monte-Carlo The number of simulated process trajectories for Monte Carlo integration. The range of possible values for this parameter is $(10^2, 10^9)$.

Euler, Milst (Milstein), Ex-n (expansion) The names of the three process approximation schemes that are compared in this application and correspond to the formulas (1.2), (1.3) and (1.4).

Start point The starting point of the diffusion process, x, in (1.1).

Time The time interval where the process will be simulated, T in (1.1).

Start Partition This is the initial number of partition time intervals in binary format. That is, the initial number of intervals is $2^{\textbf{Start Partition}}$

Step This the increasing parameter used to increase the number of partition time intervals. That is, one computes expectations for the sequence,

$$2^{StartPartition}, 2^{StartPartition+Step}, \dots, 2^{StartPartition+Step*(Points\ number-1)}.$$

Points number See the above formula.

Ef Expectation of the test function f evaluated at the corresponding stochastic process (see detailes in section 3.2). **Ef(n)** denotes the expectation computed through Monte Carlo simulations with 2^n time intervals.

Df Standard deviation of test function f at the trajectory of the corresponding stochastic process (see detailes in section 3.2). **Df(n)** denotes the standard deviation computed through Monte Carlo simulations with 2^n time intervals.

2.1.1 Noise construction mode

There are two modes for building a Wiener process which generates the corresponding diffusions ((1.2)–(1.4), etc.). If the **"standard"** checkbox is selected, the increments of the Wiener process are given by independent normal random variables with zero mean and variance equal to the length of the time interval. Otherwise, if the **Levy** checkbox is selected, the Wiener process is based on the scheme used by Levy to prove the existence of Brownian motion (see, e.g., Section 1.1, Theorem 1.1 in [1])

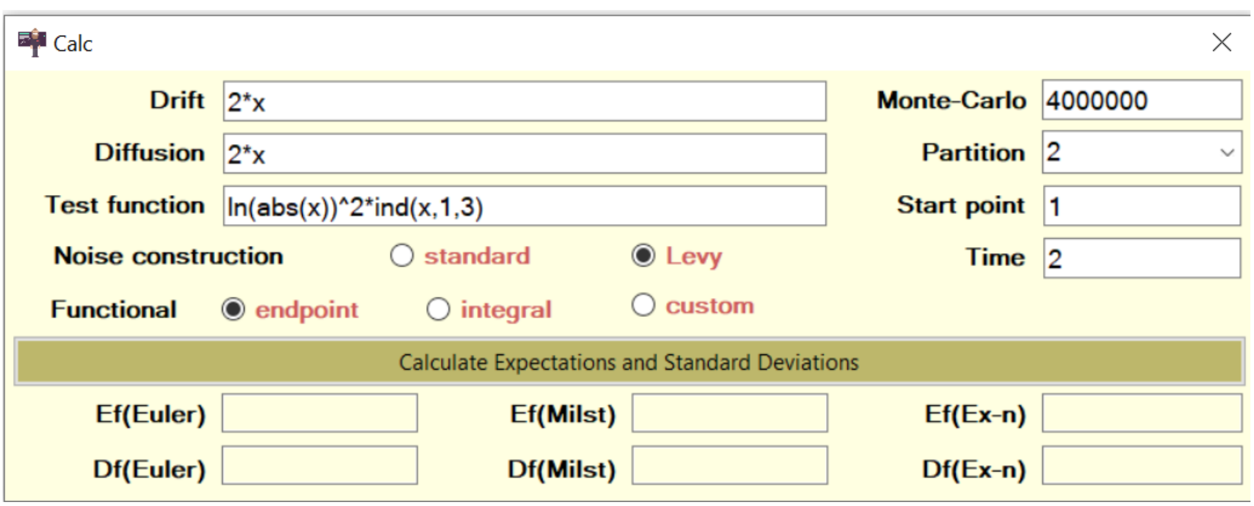
2.1.2 Function text-boxes

You can use some elementary functions using the following syntax

- $sin(x),\ cos(x),\ tg(x),\ ctg(x),\ sec(x),\ cosec(x)$
- $arcsin(x),\ arccos(x),\ arctg(x),\ arcctg(x)$ (these functions are considered with the usual range values such as $[-\pi/2, \pi/2]$ for the $arctg$ function).
- $log(a, x),\ ln(x),\ lg(x)(= log(10, x))\ (NaN\ if x \le 0\ and/or\ a \le 0\ or\ a = 1)$
- $sh(x),\ ch(x),\ th(x),\ cth(x),\ sech(x),\ cosech(x)$
- $sqrt(x) = \sqrt{x},\ (NaN\ if\ x < 0)\ exp(x) = e^x,\ abs(x) = |x|$
- $ind(x, a, b)$ – indicator function of the interval $[a, b]$
- Let **Time** is t and $0 \le a, b \le t$ then $\max(a, b) = \max\limits_{s\in[at,bt]} X_s$ and $\min(a, b) = \min\limits_{s\in[at,bt]} X_s$ be the maxima and minima of the trajectory process X on the time interval $[at, bt]$.
- $indplus(x, a)$; $indminus(x, a)$ these are the one-sided indicators ($I\{x >= a\}$ and $I\{x <= a\}$)
- The code used to represent the power function is through the operator $\wedge$, for example $x^5 =$ "x^5" or "pow(f(x), g(x))", for example $x^{cos(x)} =$ "pow(x,cos(x))".
- You can use standard operators such as "$*,\ +,\ -,\ /$" and brackets
- The program also accepts parentheses as expression of function composition, for example, $ind(t * sh(x), 0, 5)$.

3 Button EF Df Calc

After clicking this button, a new window (**Calc**) will open.



In this window, in addition to the main settings **Drift, Diffusion, Start Point, Time, Monte-Carlo, Noise construction**, there are:

3.1 Additional settings

3.1.1 Test function

In this window, the user is able to calculate the Monte Carlo approximations for the mathematical expectation and standard deviation of a function which depends on the "trajectory" of the process and which is approximated using one of the methods in (1.2)-(1.4). This function will be called the **test function**. First of all, the user must select one of three modes from the **Functional** block:

endpoint When this option is selected, the test function depends only on last point of the trajectory which is input into the **test function**.

integral In this case, the time integral of the **test function** evaluated at the process X will be calculated (here we use a simple numerical integration using the rectangular method or also called Riemann sum approximation).

custom When this checkbox is selected the user uses the custom path dependent **test function** explained in Section 6.1, in particular this is a combination of maxima and minima of the process trajectory at intervals. (*At the code level, it is possible to add custom functional from parts of the trajectory, hence the name of this mode*)

Example 3.1. For example in **"custom"** mode you can use the function

$$10 \ln \max_{s\in[0,0.33t]} X_s 1_{[1.2,3.0]} \left(\min_{s\in[0.2t,0.66t)} X_s \right),$$

(t is the length of time interval or **Time** in our notations) which can be coded as **"10*ln(abs(max(0,0.33)))*ind(min(0.2,0.66),1.2,3)"**.

The test function can accept two arguments, space, x and time t. For example,

Example 3.2. The function

$$f(x,t) = \sin^5(tx) + I_{[0.1,0.31]}(x) * \tanh(t) - \sqrt{|x|}$$

can be coded as **"sin(t*x)^5+ind(x,0.1,0.31)*th(t)-sqrt(abs(x))"**.

If this function is used in the **"endpoint"** mode, the last point of the process trajectory is substituted for argument x (see 3.2.1). If this function is used in the **"integral"** mode, then in the process of calculating the corresponding statistics (see 3.2.2), all of the trajectory partition points will be substituted into it sequentially. In the **"custom"** mode, an error message will be thrown when you try to use this function, since this mode assumes that the user sets a rule for processing the entire trajectory. However, if you need to use a function from a trajectory point in custom mode, you can use use an expression $max(\alpha, \alpha)$ instead of $X_{\alpha t}$. For example $\sin(X_{0.33t})$ can be coded as sin(max(0.33)).

It is important to use the **"custom"** mode if you plan to use the maxima or/and minima, otherwise the program will generate an input error.

3.2 Calculation formulas

Depending on the scheme and the selected mode, different formulas are used for calculations. Let us fix the start point x, time t, number of trajectories N and partition parameter n and define the samples

$$\mathbf{X_k^i} = \{X_k^{l,i} | l = 1, \ldots, 2^n\}, \quad i = 1, 2, 3, \quad k = 1, \ldots, N,$$

where

$$X_k^{0,i} = x_0, \quad X_k^{l,i} = X^{X_k^{l-1,i},i}(t/2^n),$$

and $X^{\cdot,i}(t/2^n)$ given by (1.2) – (1.4). Denote further $\xi_k^i, \zeta_k^i, i = 1,2,3$ the functionals

$$\xi_k^i := \frac{t}{2^n} \sum_{l=1}^{2^n} f\left(X_k^{l,i}, lt/2^n\right), \quad \zeta_k^i := \max_{1 \le l \le 2^n} X_k^{l,i}.$$

Recall that $i = 1,2,3$ denotes the approximation method as in (1.2) – (1.4) respectively. Hence, taking into account the **Functional** regime, we have:

3.2.1 endpoint

$$\mathbf{E} f(X^{x_0,1}(T),T) \approx \textbf{Ef(Euler)} = \frac{1}{N}\sum_{k=1}^{N} f(X_k^{2^n,1},T),$$

$$\mathbf{E} f(X^{x_0,2}(T),T) \approx \textbf{Ef(Milst)} = \frac{1}{N}\sum_{k=1}^{N} f(X_k^{2^n,2},T),$$

$$\mathbf{E} f(X^{x_0,3}(T),T) \approx \textbf{Ef(Ex-n)} = \frac{1}{N}\sum_{k=1}^{N} f(X_k^{2^n,3},T),$$

$$StdDev f(X^{x_0,1}(T),T) \approx \textbf{Df(Euler)} = \frac{1}{\sqrt{N}}\sqrt{\frac{1}{N}\sum_{k=1}^{N} f(X_k^{2^n,1},T)^2 - \left(\frac{1}{N}\sum_{k=1}^{N} f(X_k^{2^n,1},T)\right)^2},$$

$$StdDev f(X^{x_0,2}(T),T) \approx \textbf{Df(Milst)} = \frac{1}{\sqrt{N}}\sqrt{\frac{1}{N}\sum_{k=1}^{N} f(X_k^{2^n,2},T)^2 - \left(\frac{1}{N}\sum_{k=1}^{N} f(X_k^{2^n,2},T)\right)^2},$$

$$StdDev f(X^{x_0,3}(T),T) \approx \textbf{Df(Ex-n)} = \frac{1}{\sqrt{N}}\sqrt{\frac{1}{N}\sum_{k=1}^{N} f(X_k^{2^n,3},T)^2 - \left(\frac{1}{N}\sum_{k=1}^{N} f(X_k^{2^n,3},T)\right)^2}.$$

3.2.2 integral

$$\mathbf{E}\int_0^t f(X^{x_0,1}(s),s)ds \approx \textbf{Ef(Euler)} = \frac{1}{N}\sum_{k=1}^{N} \xi_k^1,$$

$$\mathbf{E}\int_0^t f(X^{x_0,2}(s),s)ds \approx \textbf{Ef(Milst)} = \frac{1}{N}\sum_{k=1}^{N} \xi_k^2,$$

$$\mathbf{E}\int_0^t f(X^{x_0,3}(s),s)ds \approx \textbf{Ef(Ex-n)} = \frac{1}{N}\sum_{k=1}^{N} \xi_k^3,$$

$$StdDev \int_0^t f(X^{x_0,1}(s),s)ds \approx \textbf{Df(Euler)} = \frac{1}{\sqrt{N}}\sqrt{\frac{1}{N}\sum_{k=1}^{N} (\xi_k^1)^2 - \left(\frac{1}{N}\sum_{k=1}^{N} \xi_k^1\right)^2},$$

$$StdDev \int_0^t f(X^{x_0,2}(s),s)ds \approx \mathbf{Df(Milst)} = \frac{1}{\sqrt{N}}\sqrt{\frac{1}{N}\sum_{k=1}^{N}(\xi_k^2)^2 - \left(\frac{1}{N}\sum_{k=1}^{N}\xi_k^2\right)^2},$$

$$StdDev \int_0^t f(X^{x_0,3}(s),s)ds \approx \mathbf{Df(Ex\text{-}n)} = \frac{1}{\sqrt{N}}\sqrt{\frac{1}{N}\sum_{k=1}^{N}(\xi_k^3)^2 - \left(\frac{1}{N}\sum_{k=1}^{N}\xi_k^3\right)^2}.$$

3.2.3 custom

If in this mode in the **test function** field you specify $\max\limits_{s\in(0,t)} X_s$ (in our syntax the test function is $\max(0,t)$), then

$$\mathbf{E}\max_{s\in[0,t]} X^{x_0,1}(s) \approx \mathbf{Ef(Euler)} = \frac{1}{N}\sum_{k=1}^{N}\zeta_k^1,$$

$$\mathbf{E}\max_{s\in[0,t]} X^{x_0,2}(s) \approx \mathbf{Ef(Milst)} = \frac{1}{N}\sum_{k=1}^{N}\zeta_k^2,$$

$$\mathbf{E}\max_{s\in[0,t]} X^{x_0,3}(s) \approx \mathbf{Ef(Ex\text{-}n)} = \frac{1}{N}\sum_{k=1}^{N}\zeta_k^3,$$

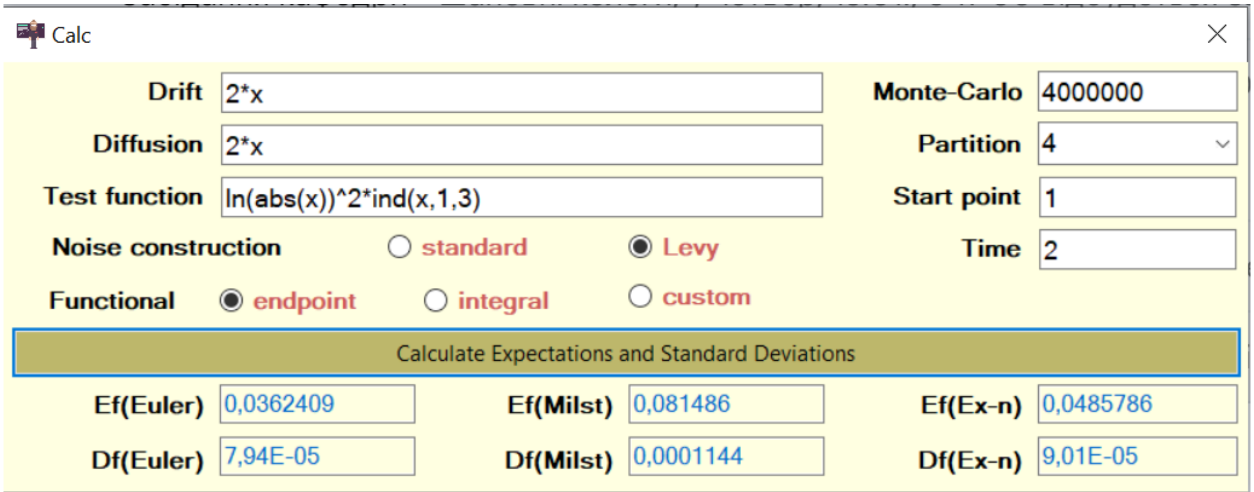
$$StdDev \max_{s\in[0,t]} X^{x_0,1}(s) \approx \mathbf{Df(Euler)} = \frac{1}{\sqrt{N}}\sqrt{\frac{1}{N}\sum_{k=1}^{N}(\zeta_k^1)^2 - \left(\frac{1}{N}\sum_{k=1}^{N}\zeta_k^1\right)^2},$$

$$StdDev \max_{s\in[0,t]} X^{x_0,2}(s) \approx \mathbf{Df(Milst)} = \frac{1}{\sqrt{N}}\sqrt{\frac{1}{N}\sum_{k=1}^{N}(\zeta_k^2)^2 - \left(\frac{1}{N}\sum_{k=1}^{N}\zeta_k^2\right)^2},$$

$$StdDev \max_{s\in[0,t]} X^{x_0,3}(s) \approx \mathbf{Df(Ex\text{-}n)} = \frac{1}{\sqrt{N}}\sqrt{\frac{1}{N}\sum_{k=1}^{N}(\zeta_k^3)^2 - \left(\frac{1}{N}\sum_{k=1}^{N}\zeta_k^3\right)^2}.$$

3.3 Results window

Before the calculations are performed you must fill all of the fields. Clicking the button **Calculate Expectations and Standard Deviations** starts a Monte-Carlo simulation for expected expectations and deviations for the three proposed methods. The results appear in the respective boxes of this computational window (**Ef(Euler), Ef(Milst), Ef(Ex-n), Df(Euler), Df(Milst), Df(Ex-n)**).



4 Integral Functional

4.1 Local time formula in the presence of a drift

Let $X_t = X_0 + A_t + M_t$ be a continuous semimartingale and $L_t^r(X)$ be its local time. Assume that the martingale part M is integrable. Then we have, for a bounded measurable nonnegative test function f,

$$\int_0^T f(X_t) d < M >_t = \int_{\mathbb{R}} f(r) L_T^r(X) dr.$$

On the other hand, the Itô-Tanaka formula gives

$$(X_T - r)^+ = (X_0 - r)^+ + \int_0^T 1_{X_t > r} \, dX_t + \frac{1}{2} L_T^r(X).$$

After taking expectation, this leads to

$$\mathbf{E}\Big[(X_T - r)^+ - (X_0 - r)^+\Big] = \mathbf{E} \int_0^T 1_{X_t > r} \, dA_t + \frac{1}{2} \mathbf{E} L_T^r(X)$$

Multiplying by $f(r)$ and integrating over r, we get

$$\frac{1}{2}\mathbf{E}\left[\int_0^T f(X_t) d < M >_t\right] = \mathbf{E}\left[\int_{\mathbb{R}} f(r) \left((X_T - r)^+ - (X_0 - r)^+ - \int_0^T 1_{X_t > r} \, dA_t\right) dr\right].$$

We have

$$F(x) = \int_{\mathbb{R}} f(r) 1_{x > r} \, dr = \int_{-\infty}^x f(r) \, dr,$$

hence

$$F' = f, \quad F(-\infty) = 0. \tag{4.1}$$

Summarizing the above calculation we can write

$$\mathbf{E}\left[\int_0^T F(X_t) \, dA_t + \frac{1}{2} \int_0^T f(X_t) d < M >_t\right] = \mathbf{E}\left[\int_{\mathbb{R}} f(r) \left((X_T - r)^+ - (X_0 - r)^+\right) dr\right]. \tag{4.2}$$

If X is a diffusion process with the coefficients a, σ, then the above identity turns into

$$\mathbf{E}\left[\int_0^T h(X_t)\,dt\right] = \mathbf{E}\left[\int_{\mathbb{R}} f(r)\left((X_T - r)^+ - (X_0 - r)^+\right)\,dr\right] \tag{4.3}$$

with

$$h = a \cdot F + \frac{1}{2}\sigma^2 F', \quad f = F',$$

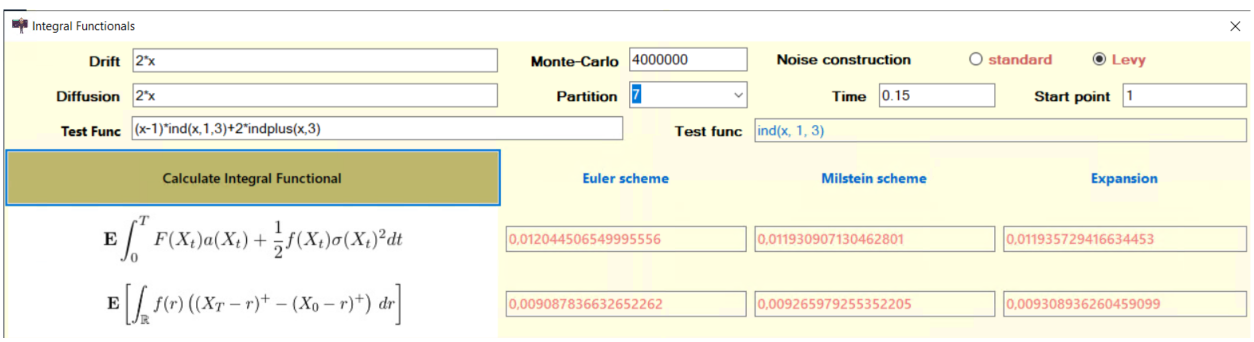
provided that $F(-\infty) = 0$. We have assumed that $f \geq 0$ but integrability of either side of the above equations for the cases of f^+ and f^- also makes the equality true for general functions $f : \mathbb{R} \to \mathbb{R}$.

4.2 Integral Functionals window and Button "IF Calc"

This functionality makes it possible to compare the left and right parts of the formula (4.2).

After clicking on the button **"IF Calc"**, a **Integral Functionals** window will open. The fields **Drift, Diffusion, Test Function, Start p., Time, Start Point, Monte-Carlo, Partition** will be filled automaticaly, but you can change them before clicking the button **Calculate Integral Functionals**. Look at the formula (4.1) to fill in the field correctly "Test Func" (recall that $f = F'$).

After clicking **"Calculate Integral Functionals"** button the integral functionals according to three schemes (**Euler, Milstein and Expansion**) will be calculated, also the field **Test function** (is a derivative of Test Function) will be filled.



5 Logarithmic error rates and Button "Error Rates 1"

This section allows you to analyze the rate of convergence of the error of approximation as a function of the partition size and the error from the theoretical average value.

We use the functional as the deviation value:

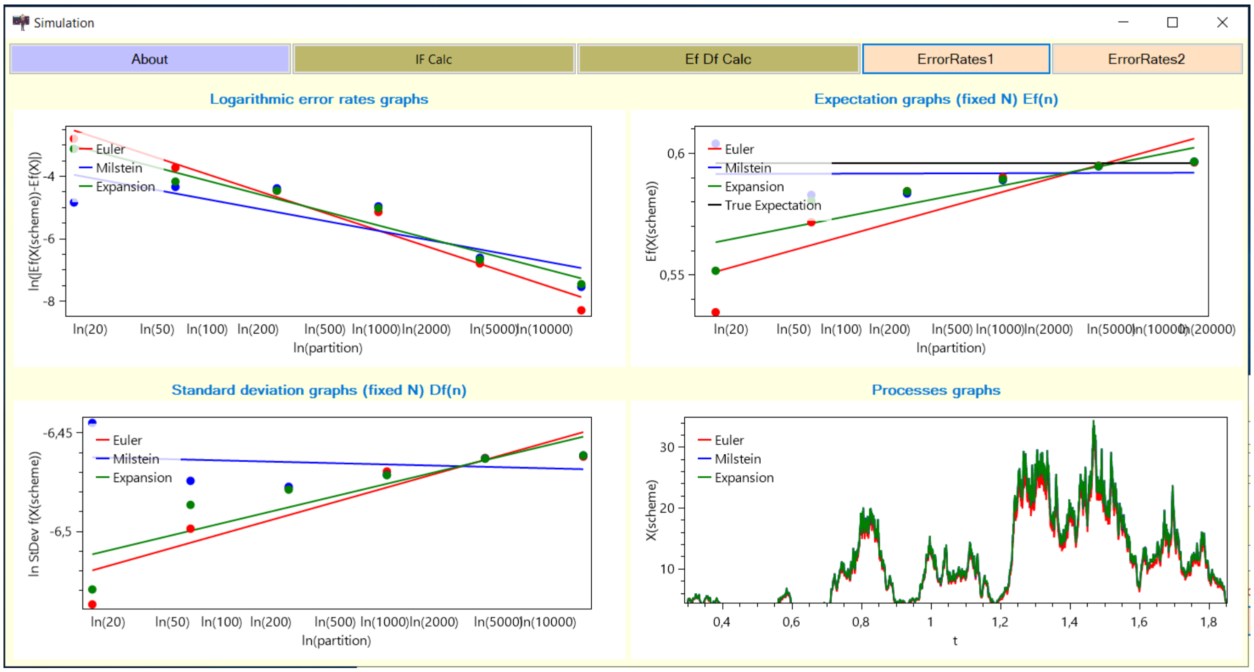
$$l(n) = ln\left|\mathbf{E}f(scheme)(n) - \mathbf{E}f(X)\right|, \tag{5.1}$$

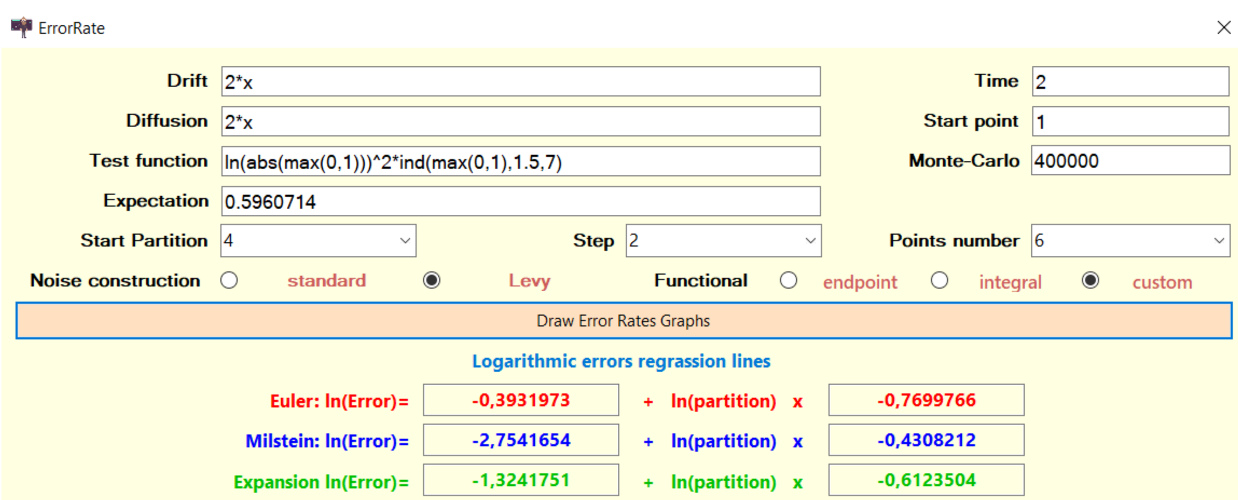
where X is a process given by (1.1), f is one of the functionals from the list $\{endpoint, integral, custom\}$ (see the definition in subsection 3.1.1) and $\mathbf{E}f(X)$ is the corresponding theoretical expectation; **scheme** $\in \{Euler, Milstein, Expansion\}$, 2^n is the number of partition intervals (see the definition of **"Start partition"** in Section 2.1). For each n $\mathbf{E}f(scheme)(n)$ is given by $\mathbf{E}f(scheme)$ (see subsections 3.2.1–3.2.3).

In order to set up the parameters of the horizontal axis ($\ln(partition)$) in the graphs, one must fill in the fields **Start Partition, Step and Points number**. Also you must fill in the fields **Monte-Carlo** (number of Monte Carlo simulations) and **Expectation** (The theoretical value to be approximated). After clicking the button **Draw Error Rates Graphs** three sets of points will be generated for each scheme separately:

$$\left\{(\ln(k), l(k)) \mid k \in \{2^{StartPartition}, 2^{StartPartition+Step}, \ldots, 2^{StartPartition+Step*(Points\ number-1)}\}\right\}.$$

These points generate the **Logarithmic error rates graphs** on the upper left of the window. Corresponding regression lines are also drawn, and the equations of these lines will appear at the bottom of the window **ErrorRate**. In the main window, you will also see graphs of the expectation approximation, and the logarithm of standard deviations, and finally one of the trajectories of the process itself using the three approximations under the same noise on the lower right corner of the window.





6 Logarithmic error rates and Button "Error Rates 2"

In this mode, the application allows you to analyze the functional (5.1) for a model in which the solution to (1.1) is known in advance.

6.1 Model and test functions

We consider a time non-homogeneous test function in order to emphasize small time effects. The general format is as follows for $0 < T_1 < T_2 < T$

$$F(X) = \varphi_1(\max_{t\in[0,T_1]} X_t) + \varphi_2(\min_{t\in[T_1,T_2]} X_t) + \varphi_3(\max_{t\in[T_2,T]} X_t),$$

where $\varphi_1, \varphi_2, \varphi_3$ are some functions to be provided by the user (using the syntax provided in Section 2.1.2) and (say) $T_1 = \frac{1}{3}T, T_2 = \frac{2}{3}T$. To make it possible the expected value of $F(X)$ to be checked, it looks reasonable to take explicitly solvable model like the one we had:

$$dX_t = 2X_t\, dt + 2X_t dW_t, \quad X_0 = 1,$$

then

$$X_t = e^{2W_t}.$$

For instance, if

$$\varphi_1(x) = 10\ln(x), \quad \varphi_2(x) = 5\ln(x)^2, \quad \varphi_3(x) = 3\ln(x)^4$$

then

$$\mathbf{E}F(X) = 20\mathbf{E}(\max_{t\in[0,T_1]} W_t) + 20\mathbf{E}(\min_{t\in[T_1,T_2]} W_t)^2 + 48\mathbf{E}(\max_{t\in[T_2,T]} W_t)^4.$$

6.2 How to calculate expectation

In our current algorithm, we need the exact value of the expectation in order to calculate the errors and put them into a logarithmic regression.

One possibility is to calculate it analytically using the fact that we know the laws of the maximum and minimum on a closed interval for a Wiener process. Say,

$$\mathbf{E}(\max_{t\in[T_2,T]} W_t)^4 = \int_{\mathbb{R}} \frac{1}{\sqrt{2\pi T_2}} e^{-\frac{1}{2T_2}x^2}\, dx \int_x^{\infty} \frac{2}{\sqrt{2\pi(T-T_2)}} e^{-\frac{1}{2(T-T_2)}(y-x)^2} y^4\, dy,$$

which can be calculated explicitly.

6.3 Model and tests

Original diffusion:

$$dX_t = X_t\, dt + X_t dW_t, \quad X_0 = 1,$$

Let **LTX** be the original diffusion after the Lamperti transformation:

$$\overline{X}_t = \ln(X_t),$$

$$d\overline{X}_t = \frac{1}{2}\, dt + dW_t, \quad \overline{X}_0 = \ln 1 = 0.$$

We use the functional as the deviation value:

$$l_1(n) = ln \left| \mathbf{E}F(X)(n) - \mathbf{E}\overline{F}(\overline{X})(n) \right|, \tag{6.1}$$

where the dependence on n in the functionals $\mathbf{E}F(X)(n)$, $\mathbf{E}\overline{F}(\overline{X})(n)$ means that we take an average over n trajectories.

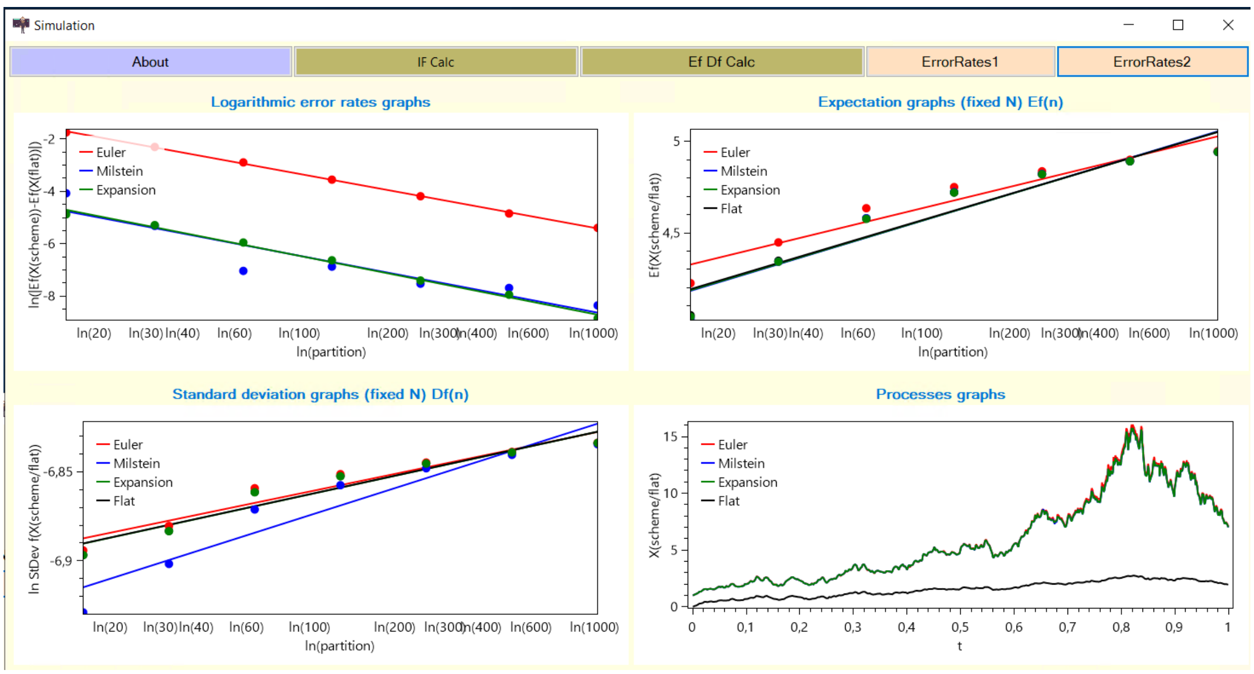
Functional A:

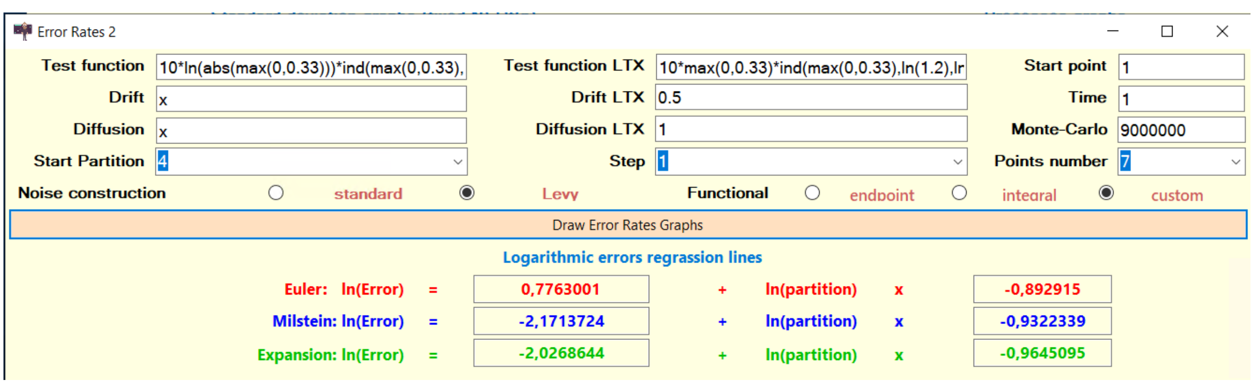
Let's take for simplicity $T = 1$. For the original process:

$$F(X) = 10 \ln \left| \max_{t\in[0,0.33)} X_t \right| 1_{[1.2,3.0]} \left(\max_{t\in[0,0.33)} X_t \right) - 5 \ln \left| \min_{t\in[0.33,0.66)} X_t \right| 1_{[0.4,0.8]} \left(\min_{t\in[0.33,0.66)} X_t \right) + \ln \left| \max_{t\in[0.66,1)} X_t \right| 1_{[1.1,2.0]} \left(\max_{t\in[0.66,1)} X_t \right).$$

For the **LTX** process:

$$\overline{F}(\overline{X}) = 10 \left(\max_{t\in[0,0.33)} \overline{X}_t \right) 1_{[\ln(1.2),\ln(3)]} \left(\max_{t\in[0,0.33)} \overline{X}_t \right) - 5 \left(\min_{t\in[0.33,0.66)} \overline{X}_t \right) 1_{[\ln(0.4),\ln(0.8)]} \left(\min_{t\in[0.33,0.66)} \overline{X}_t \right) + \left(\max_{t\in[0.66,1)} \overline{X}_t \right) 1_{[\ln(1.1),\ln(2)]} \left(\max_{t\in[0.66,1)} \overline{X}_t \right).$$





Other settings: In the above example, the time interval is $[0, 1]$, the starting point is $x_0 = 1$ and the number of trajectories (Monte-Carlo) is $9 * 10^6$. The starting number of partition intervals is 2^4, the increasing step is 1 (that is the number of partition intervals are $2^4, 2^{4+1}, 2^{4+2}, \dots$), points number is 7 i.e. points are displayed on the chart: $\{(\ln(k), l_1(k)) \mid k \in \{16, 32, 64, 128, 256, 512, 1024\}\}$

Measure	Euler	Milstein	Expansion
Slope	-0,892915	-0,9322339	-0,9645095
Intercept	0,7763001	-2,1713724	-2,0268644

Table 1: Results of the experiment for **Start partition** = 16, **Points number** = 7 **Monte-Carlo** $= 9 * 10^6$

Functional B:

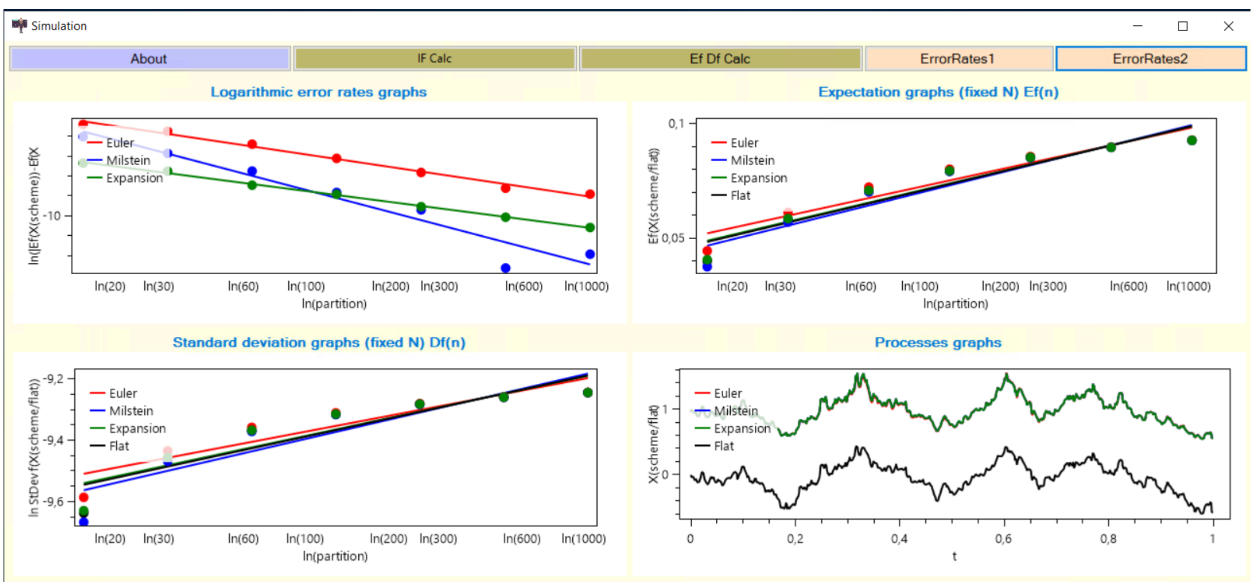
For the original process:

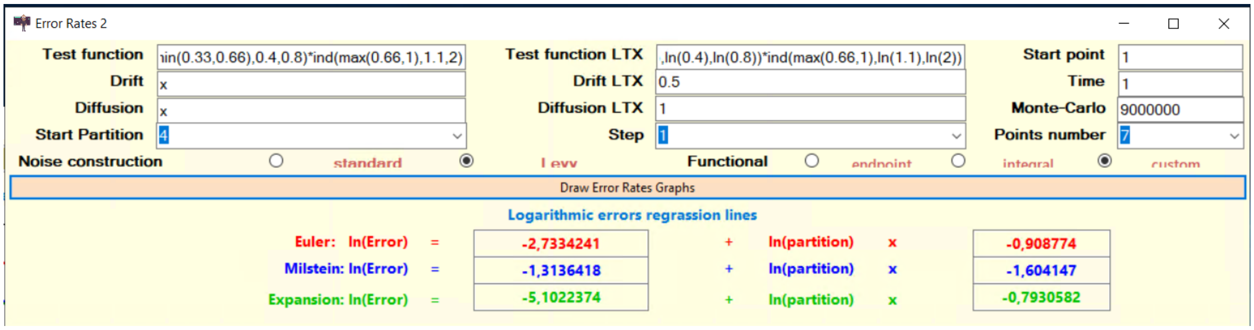
$$F(X) = 1_{[1.2,3.0]}\left(\max_{t\in[0,0.33)} X_t\right) 1_{[0.4,0.8]}\left(\min_{t\in[0.33,0.66)} X_t\right) 1_{[1.1,2.0]}\left(\max_{t\in[0.66,1)} X_t\right).$$

For the **LtDiffusion** process:

$$\overline{F}(\overline{X}) = 1_{[\ln(1.2),\ln(3.0)]}\left(\max_{t\in[0,0.33)} \overline{X}_t\right) 1_{[\ln(0.4),\ln(0.8)]}\left(\min_{t\in[0.33,0.66)} \overline{X}_t\right) 1_{[\ln(1.1),\ln(2.0)]}\left(\max_{t\in[0.66,1)} \overline{X}_t\right)$$

Here, we have divided the time period into three equal parts for reasons of ensuring that each maximum/minimum is counted by the same number of observations.





Measure	Euler	Milstein	Expansion
Slope	-0,908774	-1,604147	-0,7930582
Intercept	-2,7334241	-1,3136418	-5,1022374

Table 2: Results of the experiment for **Start partition** $= 4$ (i.e. the initial number of intervals is 2^4), **Points number** $= 7$, **Monte-Carlo** $= 9 * 10^6$.

7 About

If you received this file from a third-party location, you'll be glad to know it's always available through the application itself. It is enough to press the button **About** and a copy of this file will appear. In order to return to the functionality of the application, you need to close this description file.

8 Experiments

Our first experiment is the SDE with

$$\begin{aligned} \sigma(x) &= \sigma_0(\sin(\omega x) + 2), \\ a(x) &= -\frac{x}{x^2 + \frac{c_1}{3c_3}}\sigma^2(x), \\ f(x) &= c_3x^3 + c_1x + c_0. \end{aligned} \tag{Case 1}$$

Here $\sigma_0, \omega, c_0, c_1$ and c_3 are constants. With this particular model choice, we have that

$$a(x)f'(x) + 2^{-1}\sigma^2(x)f''(x) = 0,$$

and therefore $f(X_t)$ is a martingale with $\mathbf{E}f(X_T) = f(X_0)$ for any T. In the experiment we choose the parameters given in Table 3 which give $f(X_0) = 2$. Select the noise construction mode **Levy**, and select the functional **endpoint**. (We use here button **Ef Df Calc**)

σ_0	ω	c_0	c_1	c_3	X_0	2^n	N	$a(x)$	$\sigma(x)$	$f(x)$
2	3	0	1	1	1	2^{10}	2^{20}	$-\frac{12x(sin(3x)+2)^2}{3x^2+1}$	$2(sin(3x)+2)$	x^3+x

Table 3: Parameters in experiment

Measure	Euler	Milstein	Expansion
Ef	1,6256695	2,3830532	2,3322897
Df	0,092603	0,0924982	0,0898824

Table 4: Results of the experiment with $T = 2$

Measure	Euler	Milstein	Expansion
Ef	1,8348953	2,0546978	2,0494653
Df	0,042516	0,0424202	0,0419884

Table 5: Results of the same experiment with $T = 1$

Measure	Euler	Milstein	Expansion
Ef	1,9982057	2,0496934	2,0532795
Df	0,0213261	0,0212073	0,0212026

Table 6: Results of the same experiment with $T = 0.5$

Measure	Euler	Milstein	Expansion
Ef	1,9935544	2,013727	2,0167519
Df	0,0133162	0,01324	0,0132496

Table 7: Results of the same experiment with $T = 0.3$

Second experiment The second experiment also involves the use of local time as explained in Section 4. In this example, we will use the button **"IF Calc"** and **"Ef Df Calc"** (see Sections 3 and 4). Let's consider the Black-Scholes model

$$dX_t = \sigma X_t dW_t$$

and specify the following parameters

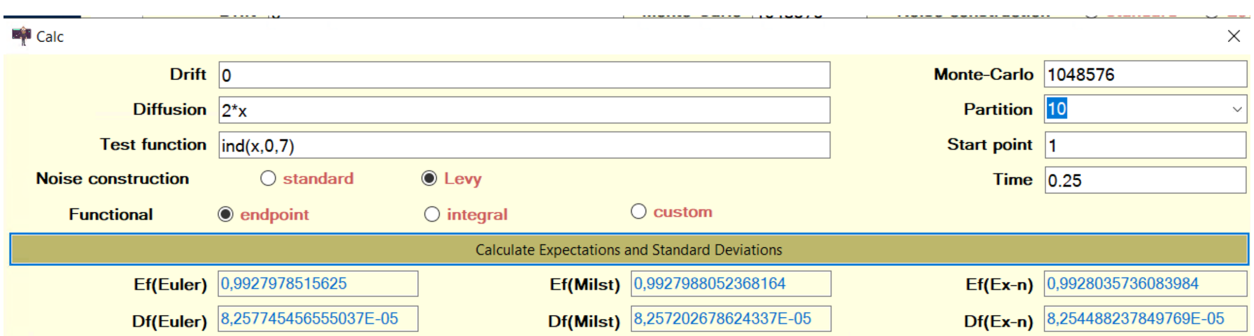
T	X_0	2^n	N	$a(x)$	$\sigma(x)$	$f(x)$	$F(x)$
0.25	1	2^{10}	2^{20}	0	$\mathbf{2x}$	$\mathbf{1}_{[\mathbf{0},\mathbf{7}]}(\mathbf{x})$	$\mathbf{x1}_{[\mathbf{0},\mathbf{7}]}(\mathbf{x}) + \mathbf{71}_{(\mathbf{7},+\infty)}(\mathbf{x})$

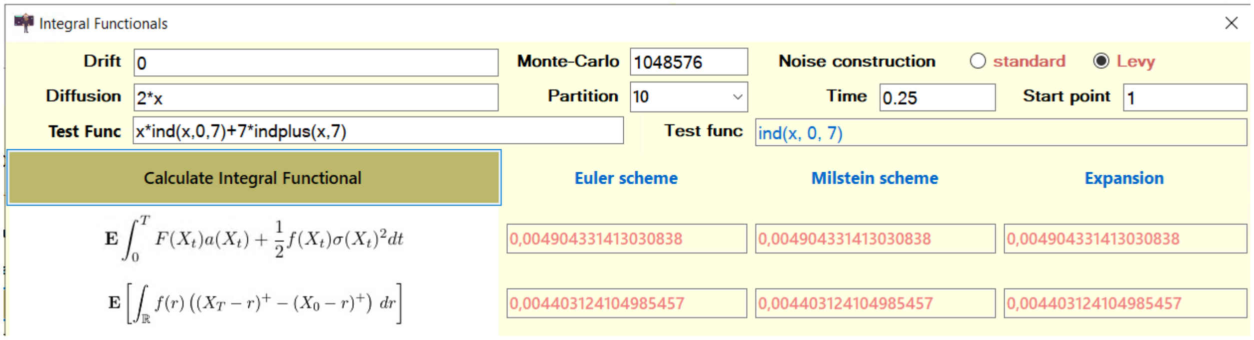
Table 8: Parameters in experiment

Note that $h(x) = a(x)F(x) + \frac{1}{2}\sigma^2(x)f(x)$ (see Section 4). Denote $h_1(r) = f(r)\left((X_T - r)^+ - (X_0 - r)^+\right)$

Measure	Euler	Milstein	Expansion
$\mathbf{E}\int_0^T h(X_t)\,dt$	0,004904331413030838	0,004904331413030838	0,004904331413030838
$\mathbf{E}\int_{\mathbb{R}} h_1(r)dr$	0,004403124104985457	0,004403124104985457	0,004403124104985457
Ef	0,9927978515625	0,9927988052368164	0,9928035736083984
Df	$8,257745456555037 * 10^{-5}$	$8,257202678624337 * 10^{-5}$	$8,254488237849769 * 10^{-5}$

Table 9: Results of the experiment with the parameters from the Table 8



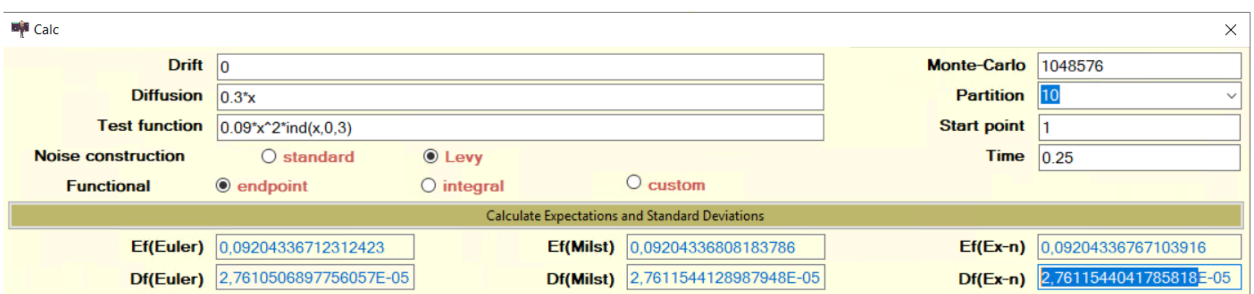


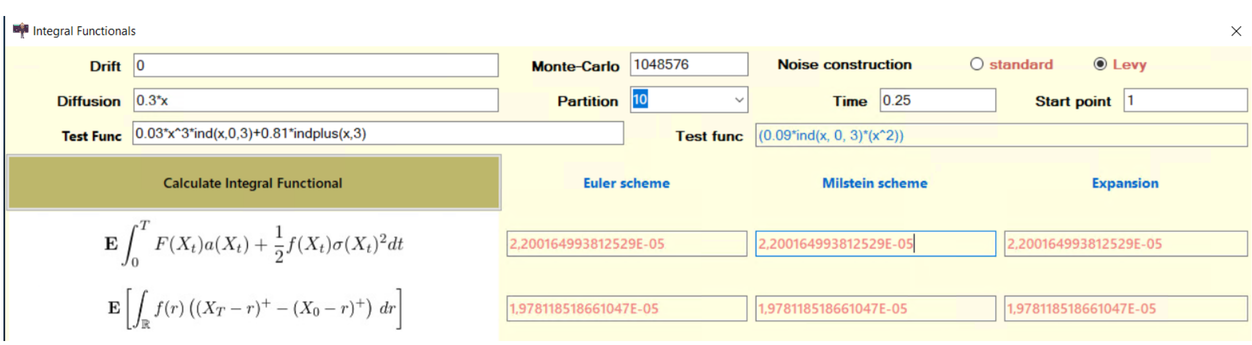
T	X_0	2^n	N	$a(x)$	$\sigma(x)$	$f(x)$	$F(x)$
0.25	1	2^{10}	2^{20}	0	$\mathbf{0.3x}$	$\mathbf{0.09x^2 1_{[0,3]}(x)}$	$\mathbf{0.03x^3 1_{[0,3]}(x) + 0.81 1_{(3,+\infty)}(x)}$

Table 10: Parameters in experiment

Measure	Euler	Milstein	Expansion
$\mathbf{E}\int_0^T h(X_t)dt$	$2,200164993812529 * 10^{-5}$	$2,200164993812529 * 10^{-5}$	$2,200164993812529 * 10^{-5}$
$\mathbf{E}\int_{\mathbb{R}} h_1(r)dr$	$1,978118518661047 * 10^{-5}$	$1,978118518661047 * 10^{-5}$	$1,978118518661047 * 10^{-5}$
Ef	0,09204336712312423	0,09204336808183786	0,09204336767103916
Df	$2,7610506897756057 * 10^{-5}$	$2,7611544128987948 * 10^{-5}$	$2,7611544041785818 * 10^{-5}$

Table 11: Results of the experiment with the parameters from the Table 10





Third experiment Another model is the Vasicek model

$$dr_t = c(b - r_t)\,dt + \sigma\,dW_t$$

This model satisfies that $r_t \sim N\left(\mu_t, \sigma_t^2\right)$ where

$$\mu_t := r_0 e^{-ct} + b(1 - e^{-ct})$$
$$\sigma_t^2 := \frac{\sigma^2}{2c}(1 - e^{-2ct})$$

Classical parameters maybe

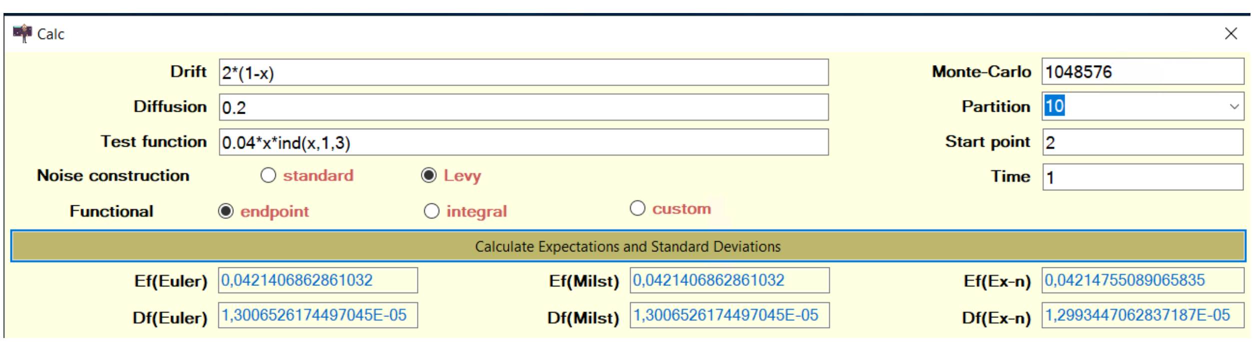
T	X_0	2^n	N	$a(x)$	$\sigma(x)$	$f(x)$	$F(x)$
1	2	2^{10}	2^{20}	$2(1-x)$	0.2	$0.04x\mathbf{1}_{[1,3]}(x)$	$0.02(x^2-1)\mathbf{1}_{[1,3]}(x) + 0.16\mathbf{1}_{(3,+\infty)}(x)$

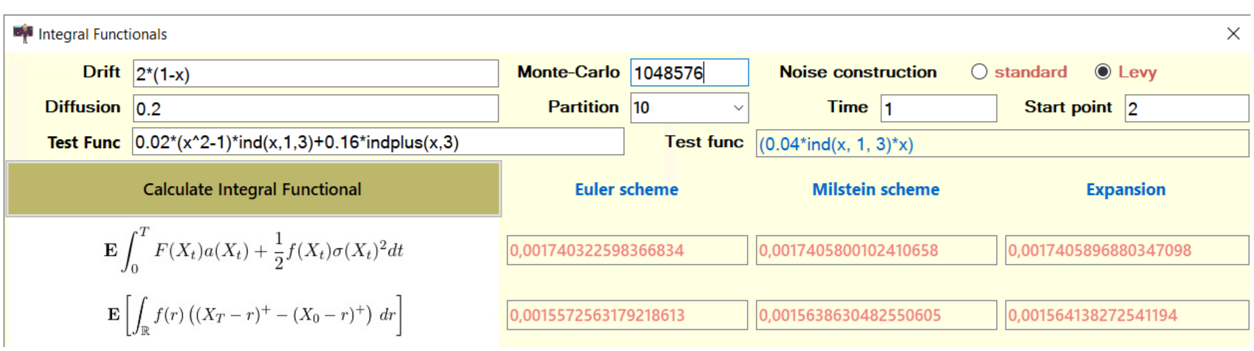
Table 12: Parameters in experiment

Recall that c measures the strength of mean reversion and b its center.

Measure	Euler	Milstein	Expansion
$\mathbf{E}\int_0^T h(X_t)dt$	0,001740322598366834	0,0017405800102410658	0,0017405896880347098
$\mathbf{E}\int_{\mathbb{R}} h_1(r)dr$	0,0015572563179218613	0,0015638630482550605	0,001564138272541194
Ef	0,0421406862861032	0,0421406862861032	0,04214755089065835
Df	$1,3006526174497045 * 10^{-5}$	$1,3006526174497045 * 10^{-5}$	$1,2993447062837187 * 10^{-5}$

Table 13: Results of the experiment with the parameters from the Table 12





9 Maximum and total variation

Some explicit examples:

If one chooses $b = \frac{1}{2}\sigma^2$

$$dX_t = bX_t dt + \sigma X_t dW_t$$

then $X_t = X_0 \exp(\sigma W_t)$ and therefore $\max_{t\in[0,T]} X_t = X_0 \exp(\sigma \max_{t\in[0,T]} W_t)$. With this information one can compute explicitly the law of $\max_{t\in[0,T]} X_t$. In that setting, we have as first example for $(x-1) * ind(x,1,a)$ that the exact answer is

```
myanswer <- function(sig,T,a){
         answer = 2*exp(sig*sig*T/2)
         *(pnorm((log(a)/sig-sig*T)/(sqrt(T)))-pnorm(-sig*sqrt(T)))-
         return(answer)
}
```

In the case that the test function is $log(e,x) * log(e,x) * ind(x,1,a)$

```
myanswer2 <- function(sig,T,a){
         answer = (2*sig*sig)*(-sqrt(T)*log(a)
         *exp(-(log(a)/sig)^2/(2*T))/(sig*sqrt(2*pi))+T*pnorm(log(a)
         return(answer)
}
}
```

References

[1] P. Morters and Y. Peres. *Brownian Motion*, Cambridge University Press, 2010, 416pp., **ISBN: 978-0-521-76018-8**.

[2] Y. Ait-Sahahia. Closed-Form Likelihood Expansions for Multivariate Diffusions, Annals of Statistics, 2008, 36, 906-937.

[3] R. Azencott. Densité des diffusions en temps petit: développements asymptotiques. I, Seminar on probability, XVIII, Lecture Notes in Math.1059, Springer, Berlin, 1984, 402–498.

[4] S. Bodnarchuk, D. Ivanenko, A. Kohatsu-Higa, A. Kulik. Improved local approximation for multidimensional diffusions: the G-rates. Teoriya Ĭmovirnosteĭ ta Matematychna Statystyka, 101 (2019), No. 2, pp. 15 - 39

[5] A. Friedman. Partial differential equations of parabolic type, Dover Publications, Inc., 1964.

[6] D.O. Ivanenko, A. Kohatsu-Higa and A.M. Kulik, LA(M)N property for diffusion models with low regularity coefficients, *Manuscript*.

[7] D.O. Ivanenko, A. Kohatsu-Higa and A.M. Kulik, Small time chaos approximations for heat kernels of multi-dimensional diffusions. BIT, 63, article number 16, (2023),

[8] D.O. Ivanenko, A. Kohatsu-Higa and A.M. Kulik, The three point parametrix method. In preparation.

[9] V. Konakov, E. Mammen. Local limit theorems for transition densities of Markov chains converging to diffusions. Probab Theory Relat. Fields 117, 551–587, 2000.

[10] V. Konakov, E. Mammen. Small time Edgeworth-type expansions for weakly convergent nonhomogeneous Markov chains. Probab. Theory Relat. Fields 143, 137, 2009.

[11] M. Lorig, S. Pagliarani, A. Pascucci. Analytical expansions for parabolic equations SIAM J. Appl. Math., Vol.75 n.2 pp.468-491, 2015