`SDE_DensityTracking`
A Python Package

The implementation consists of one single class that tracks the density distribution of the solution to a stochastic differential equation (SDE). Starting from a generic SDE

$$\mathrm{d}X_t = \mu(X_t)\ \mathrm{d}t + \sigma(X_t)\ \mathrm{d}W_t \tag{1}$$

with some time-independent drift and volatility functions μ and σ, and initial position X_0, the class calculates the probability density $p(x,t)$, to be at position x at time t. The class can furthermore deal with absorbing or reflective boundary conditions.

1 Input & Output

Upon initialization of the class with appropriate arguments, one starts tracking the evolution of the density by calling the `track_density` method. Once finished, the density $p = p(x,t)$ can be retrieved by calling the `get_density` method. The return value can be either a functional or discretized object, and it can be either a function of x, of t, or both. See documentation inside `SDE_DensityTracking.py` for a detailed description of all input and output values.

2 Theoretical Background

In our implementation, we have followed the approach of Bhat and Madushani (2016), which is straight forward. We first discretize (1) in time, giving

$$X_{t+1} = X_t + \mu(X_t)\Delta t + \sigma(X_t)\sqrt{\Delta t} Z_t \tag{2}$$

with Δt the discretization step width and Z_t a normally distributed random variable. Looking at (2), we can see immediately that the position of X_{t+1} is just a Gaussian with mean $X_t + \mu(X_t)\Delta t$ and standard deviation $\sigma(X_t)\sqrt{\Delta t}$. Let us denote by $G(x,y)$ a Gaussian with mean $y + \mu(y)\Delta t$ and standard deviation $\sigma(y)\sqrt{\Delta t}$. Then, the density at position x at time $t+1$ is just given by the Chapman-Kolmogorov equation

$$p(x, t+1) = \int_{\mathbb{R}} \mathrm{d}y\ G(x,y)\ p(y,t). \tag{3}$$

Discretizing (3) yields

$$p(x, t+1) \approx k \sum_{i=-\infty}^{\infty} \mathrm{d}y\ G(x,y)\ p(y_i,t). \tag{4}$$

where k is the spatial discretization step, ie. $k = y_{i+1} - y_i\ \forall i$. For the numerical implementation, the infinite sum in (4) is truncated naturally by just summing over all values y_i

calculated at time t. The considered value range x_i at time $t+1$ is determined dynamically at runtime as the minimum and maximum value below and above which the density is less than some pre-specified precision value `tol`, that we can safely approximate as zero. [1] This new range of values at time $t+1$ naturally truncates the summation in (4) in the next iteration, and so on. If there is an absorbing boundary, we must just replace the Gaussian $G(x, y)$ in (4) by

$$p(s, t \; ; \; s_0, \underline{s} = 0) = \frac{1}{\sqrt{2\pi\sigma^2 t}} \left[\exp\left(-\frac{(s_0 + \mu t - s)^2}{2\sigma^2 t} \right) - \exp\left(-\frac{2 s_0 \mu}{\sigma^2} \right) \exp\left(-\frac{(s_0 - \mu t + s)^2}{2\sigma^2 t} \right) \right], \tag{5}$$

which is the density of a random walk with drift μ and volatility σ, in presence of an absorbing boundary at $s = 0$. A formula similar to equation (5) exists for the reflective barrier. See for instance Molini et al. (2011) for a derivation of these results.

Note that technically, we are not solving the Fokker-Planck equation, but the Chapman-Kolmogorov forward equation. This implementation has the additional advantage that the range of x-values is dynamic (no previous grid must be specified), and μ and σ must not necessarily be differentiable.

3 Consistency check

We have checked the validity of the implementation in figure 1.

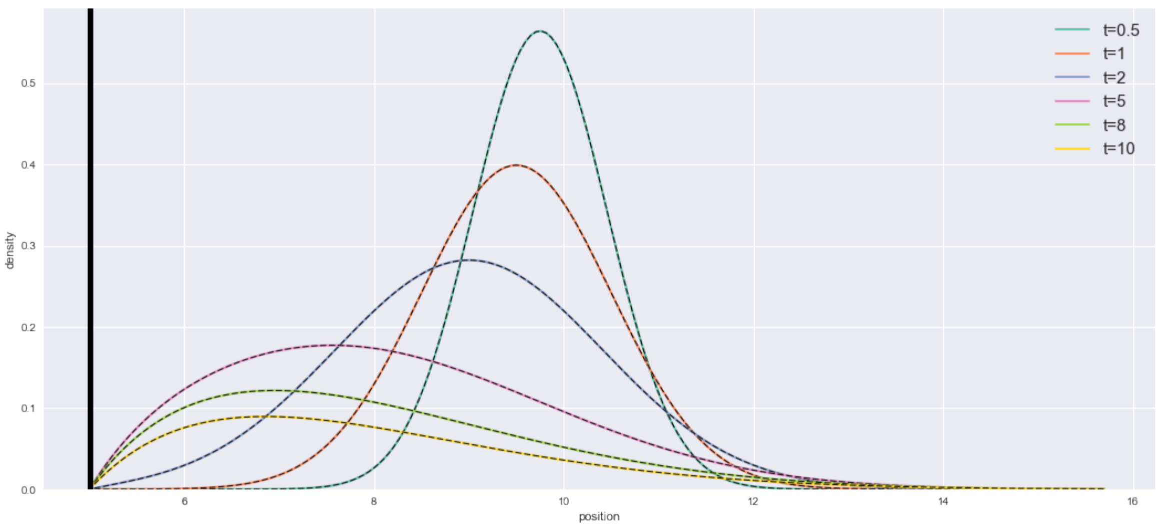

Figure 1: We test the implementation of our PDE solver by considering the case of constant drift and volatility, $\mu(x) = -0.5$, $\sigma = 1$, initial condition $p(x, t = 0) = \delta(x - 10)$ and an absorbing barrier at $x = 5$. The black dashed lines on top of the colored lines indicates the analytical solution (5), in excellent agreement with the numerical one.

[1] In principle, this implementation can cause some problem in case of a a bimodal initial distribution with two unconnected peaks, i.e. with regions of zero probability mass between the two peaks. Because this is a very specific case, we ignore it for here for simplicity.

References

Bhat, H. S. and R. W. M. A. Madushani
2016. Density tracking by quadrature for stochastic differential equations. *arXiv preprint arXiv:1610.09572*.

Molini, A., P. Talkner, G. Katul, and A. Porporato
2011. First passage time statistics of Brownian motion with purely time dependent drift and diffusion. *Physica A*, 390(11):1841–1852.

4 Usage