Generative modelling of Lévy area for high order SDE simulation

James Foster

University of Bath

joint with Andraž Jelinčič (Oxford),
Jiajie Tao (UCL), William Turner (Imperial),
Thomas Cass (Imperial) and Hao Ni (UCL)

Outline

Introduction

Consider the stochastic differential equation (defined in the Itô sense),

$$dy_t = f(y_t)\,dt + \sum_{i=1}^{d} g_i(y_t)\,dW_t^{(i)}, \tag{1}$$

where $f, g_i : \mathbb{R}^n \to \mathbb{R}^n$ are each smooth and bounded vector fields on $\mathbb{R}^n$ and $W = \{W_t\}$ denotes a standard d-dimensional Brownian motion.

SDEs can model random time-evolving systems and have applications ranging from finance [1] to statistical physics and machine learning [2].

We use the notation $X_{s,t} := X_t - X_s$ to denote increments of a process X and for vectors $A, B \in \mathbb{R}^m$, we write $A \otimes B = \{A_i B_j\}_{1\leq i,j\leq m}$.

Introduction

There are two particularly well-known numerical methods for Itô SDEs:

- Euler-Maruyama method

$$Y_{k+1} := Y_k + f(Y_k)\,(t_{k+1} - t_k) + \sum_{i=1}^{d} g_i(Y_k)\big(W_{t_{k+1}}^{(i)} - W_{t_k}^{(i)}\big)$$

- Milstein's method

$$\begin{aligned} Y_{k+1} := Y_k + f(Y_k)\,(t_{k+1} - t_k) &+ \sum_{i=1}^{d} g_i(Y_k)\big(W_{t_{k+1}}^{(i)} - W_{t_k}^{(i)}\big) \\ &+ \sum_{i,j=1}^{d} g_j'(Y_k)g_i(Y_k)\int_{t_k}^{t_{k+1}} \big(W_t^{(i)} - W_{t_k}^{(i)}\big)\,dW_t^{(j)} \end{aligned}$$

Introduction

Definition (Strong convergence)

Let $Y = \{Y_k\}_{0\leq k\leq N}$ denote a numerical solution with N steps and $h = \frac{T}{N}$. Then Y is said to converge strongly to the solution of (1) with order α if there exists $C > 0$ such that

$$\mathbb{E}\Big[\big\|Y_k - y(t_k)\big\|_2^2\Big]^{\frac{1}{2}} \leq Ch^{\alpha},$$

for all $k \in \{0, 1, \cdots, N\}$ and sufficiently small h (where $t_k := kh$).

Theorem (Cameron & Clark [3] and Dickinson [4])

Numerical approximations using only increments (or linear functionals) of Brownian motion achieve at best a strong convergence rate of $O(h^{\frac{1}{2}})$.

Theorem (Milstein & Tretakov [5])

For smooth f and g_i, Milstien's method converges strongly with order 1.

Introduction

Therefore, if we could accurately generate the iterated integrals in the Milstein method, then we would obtain significantly faster convergence (e.g. 100 steps instead of 10,000 steps).

Using Itô's lemma, we can rewrite iterated integrals as the sum of a (tractable) symmetric term and an (intractable) antisymmetric term.

Theorem (Decomposition of second iterated integrals)

$$\int_s^t \big(W_u^{(i)} - W_s^{(i)}\big)\, dW_u^{(j)} = \frac{1}{2}\Big(\big(W_t^{(i)} - W_s^{(i)}\big)\big(W_t^{(j)} - W_s^{(j)}\big) - (t-s)\delta_{ij}\Big) + A_{s,t}^{(i,j)},$$

where δ_{ij} is the Kronecker delta and $A_{s,t}$ is an antisymmetric matrix with

$$A_{s,t}^{(i,j)} := \frac{1}{2}\bigg(\int_s^t \big(W_u^{(i)} - W_s^{(i)}\big)\, dW_u^{(j)} - \int_s^t \big(W_u^{(j)} - W_s^{(j)}\big)\, dW_u^{(i)}\bigg). \qquad (2)$$

Introduction

Definition

The Lévy area of a standard d-dimensional Brownian motion over an interval $[s,t]$ is the antisymmetric $d \times d$ matrix $A_{s,t}$ given by (2).

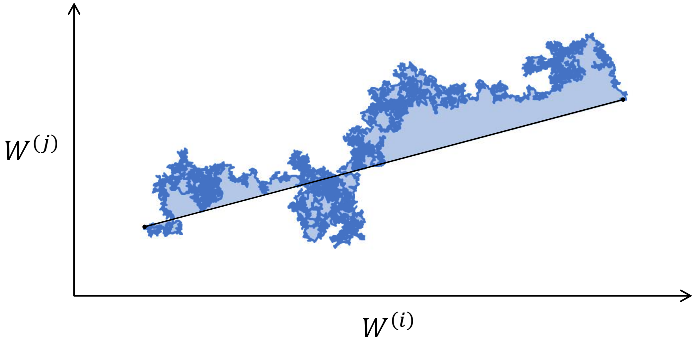


Figure: Each entry $A_{s,t}^{(i,j)}$ can be understood as the chordal area between the independent Brownian motions $W^{(i)}$ and $W^{(j)}$ (diagram adapted from [6]).

Approximations so far...

Lévy area approximation	Strong convergence rate (with N random variables)
Exact [7] $(d = 2)$	N/A
Fourier Series [8, 9] (with correlation)	$\sim \frac{1}{\pi}\frac{1}{\sqrt{N}}$
Fourier series [9, 10] (ignoring correlation)	$\sim \frac{\sqrt{3}}{\pi}\frac{1}{\sqrt{N}}$
Polynomial [11, 12]	$\sim \frac{1}{2\sqrt{2}}\frac{1}{\sqrt{N}}$
"Moment matching" [6, 13, 14, 15, 16, 17]	$O(\frac{1}{N})^*$

* This faster convergence rate requires non-trivial couplings, which means that the asymptotic constants are unknown.

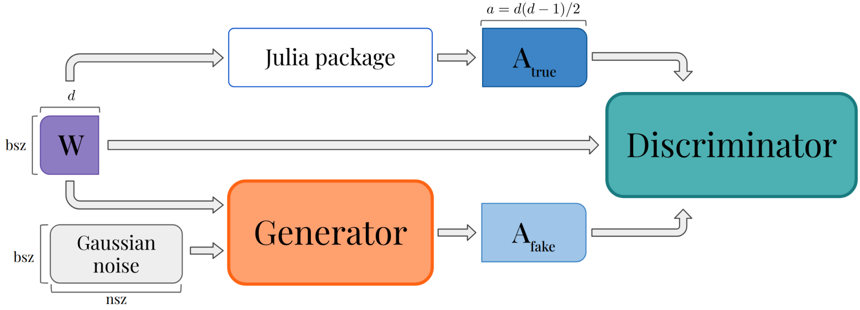
Outline

Generating Lévy area with machine learning

Since we expect Lévy area to have a smooth "bell shaped" distribution, it may be reasonable to consider **learning-based generative models**.

For example, we can use a conditional Generative Adversarial Network (GAN), trained on a dataset of "finely discretized" Lévy area samples.

Generating accurate Lévy areas is expensive, but can certainly be done! (e.g. using the Julia package [18])



Generating Lévy area with machine learning

This works... but can we exploit the properties of Lévy Area?

- Independence of $W^{(i)}$ and $W^{(j)}$ (when $i \neq j$).
- Relationship between increments and areas

$$A_{s,t}^{(i,j)} = H_{s,t}^{(i)} W_{s,t}^{(j)} - W_{s,t}^{(i)} H_{s,t}^{(j)} + b_{s,t}^{(i,j)},$$

where $H_{s,t}, b_{s,t}$ depend on the Brownian bridge (independent of $W_{s,t}$).

- Unbiasedness of Lévy area

$$\mathbb{E}\big[A_{s,t}^{(i,j)} \,\big|\, W_{s,t}\big] = 0.$$

- Self-similarity of Brownian motion and Lévy area

$$A_{s,t}^{(i,j)} = A_{s,u}^{(i,j)} + \frac{1}{2}\Big(W_{s,u}^{(i)} W_{u,t}^{(j)} - W_{s,u}^{(j)} W_{u,t}^{(i)}\Big) + A_{u,t}^{(i,j)}.$$

Incorporating structure and unbiasedness

Underlying our generator is the following property about Lévy area:

Theorem (Decomposition of Brownian Lévy area)

$$A_{s,t} = H_{s,t} \otimes W_{s,t} - W_{s,t} \otimes H_{s,t} + b_{s,t}\,, \tag{3}$$

where $W_{s,t} \sim \mathcal{N}^d\big(0, (t-s)\big)$, $H_{s,t} \sim \mathcal{N}^d\big(0, \frac{1}{12}(t-s)\big)$ *and* $b_{s,t}$ *are given by*

$$W_{s,t} := W_t - W_s\,,$$

$$H_{s,t} := \frac{1}{t-s}\int_s^t \bigg(\big(W_s - W_u\big) - \frac{u-s}{t-s}W_{s,t}\bigg)\,du,$$

$$b_{s,t}^{(i,j)} := \int_s^t B_{s,u}^{(i)}\,dB_u^{(j)},$$

with $B_{s,u} := W_{s,u} - \frac{u-s}{t-s}W_{s,t}$ *denoting the associated Brownian bridge.*

Incorporating structure and unbiasedness

Unsurprisingly, the following all have the same "Lévy area" distribution (conditional on $W_{s,t}$):

$$A_{s,t} := H_{s,t} \otimes W_{s,t} - W_{s,t} \otimes H_{s,t} + b_{s,t}$$

$$\widehat{A}_{s,t} := \big(-H_{s,t}\big) \otimes W_{s,t} - W_{s,t} \otimes \big(-H_{s,t}\big) + b_{s,t}$$

$$\overline{A}_{s,t} := \big(-H_{s,t}\big) \otimes W_{s,t} - W_{s,t} \otimes \big(-H_{s,t}\big) - b_{s,t} = -A_{s,t}$$

The middle term is obtained by "flipping" the Brownian bridge $B \mapsto -B$.
The last uses symmetry of the "Brownian bridge Lévy area" distribution.

Can we incorporate these symmetries into our generator?

Incorporating structure and unbiasedness

Therefore, we propose a "bridge-flipping" generator of the form:

$$BF\big(w, h, b, \xi_0, \xi\big)^{(i,j)} = \xi_0\Big(\big(\xi^{(i)}h^{(i)}\big)w^{(j)} - w^{(i)}\big(\xi^{(j)}h^{(j)}\big) + \xi^{(i)}\xi^{(j)}b^{(i,j)}\Big),$$

where ξ_0 and $\xi^{(1)}, \cdots, \xi^{(d)}$ are independent Rademacher variables, $w^{(1)}, \cdots, w^{(d)}$ are Brownian increments, $h^{(i)} \sim \mathcal{N}\big(0, \frac{1}{12}(t-s)\big)$ and

$$b^{(i,j)} := NN_\theta\Big(\big(h^{(i)}, z^{(i)}\big), \big(h^{(j)}, z^{(j)}\big)\Big), \tag{4}$$

with $z^{(1)}, \cdots, z^{(d)} \sim \mathcal{N}(0,1)$ denoting latent Gaussian noise.

The above "pairwise" neural network is based on the fact that each Lévy area $b_{s,t}^{(i,j)}$ only depends on the Brownian bridges $B^{(i)}$ and $B^{(j)}$.

Incorporating structure and unbiasedness

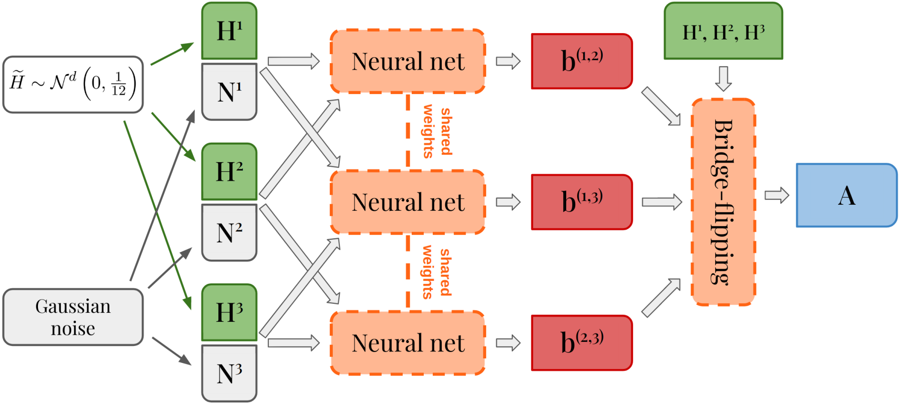


Figure: Schematic of the "PairNet and Bridge-Flipping" generator when $d = 3$.

Incorporating structure and unbiasedness

Theorem

Conditional on the Brownian increment $W_{0,1}$, both the Lévy area $A_{0,1}$ and the output of the "Bridge-Flipping" generator have zero odd moments.

This generator's unbiasedness is particularly helpful for SDE numerics.

Theorem

Milstien's method with "fake" Lévy area converges strongly with rate 0.5 provided that the "fake" Lévy area has mean zero and $O(h^2)$ covariance.

On the other hand, if the fake Lévy area has an $O(h)$ expectation, then the resulting biases can "add up" over time and result in an $O(1)$ error!

But how is the generator actually trained...

Training based on Brownian motion's self-similarity

Chen's relation allows us to "combine" Lévy areas over two intervals.

Theorem (Chen's relation for (second) iterated integrals)

$$W_{s,t} = W_{s,u} + W_{u,t},$$
$$A_{s,t} = \underbrace{A_{s,u} + \frac{1}{2}\big(W_{s,u}\otimes W_{u,t} - W_{u,t}\otimes W_{s,u}\big) + A_{u,t}}_{:=\,\mathrm{Chen}((W_{s,u},\,A_{s,u}),\,(W_{u,t},\,A_{u,t}))}.$$

By the self-similarity / scaling property of Brownian motion, we want

$$G_\theta\big(W_{0,1}, z\big) \sim \mathrm{Chen}\Bigg(\Big(W_{0,\frac{1}{2}}, \frac{1}{4}G_\theta\big(\widetilde{W}_{0,\frac{1}{2}}, \widetilde{z}\big)\Big), \Big(W_{\frac{1}{2},1}, \frac{1}{4}G_\theta\big(\widetilde{W}_{\frac{1}{2},1}, \widehat{z}\big)\Big)\Bigg),$$

where G_θ is our generator, $\widetilde{W}_{a,a+\frac{1}{2}} := \sqrt{2}W_{a,a+\frac{1}{2}} \sim \mathcal{N}(0,1)$ are rescaled Brownian increments and $z, \widetilde{z}, \widehat{z} \sim \mathcal{N}(0,1)$ denote latent Gaussian noise.

Training based on Brownian motion's self-similarity

Therefore, we propose an adversarial "Chen training" for our model:

$$\min_{\theta}\max_{\phi}\Big(\mathrm{Loss}\big(\theta,\phi,\boldsymbol{w}\big)\Big),$$

with $\boldsymbol{w}\in\mathbb{R}^{N\times d}$ denoting N samples of the increment $W_{0,1}\sim\mathcal{N}(0,1)$, and the parametrized loss function is given by

$$\mathrm{Loss}\big(\theta,\phi,\boldsymbol{w}\big) := \frac{1}{M}\sum_{i=1}^{M}\Big\|D_{\phi_i}\big(\boldsymbol{w},G_\theta(\boldsymbol{w})\big)-D_{\phi_i}\big(\boldsymbol{w}_{\mathrm{Chen}},G_\theta(\boldsymbol{w}_{\mathrm{Chen}})\big)\Big\|,$$

where each discriminator D_{ϕ_i} is a "unitary characteristic function", but with the expectation computed empirically at points given by ϕ_i.

See preprint for further details... (including a Theorem!)

Training based on Brownian motion's self-similarity

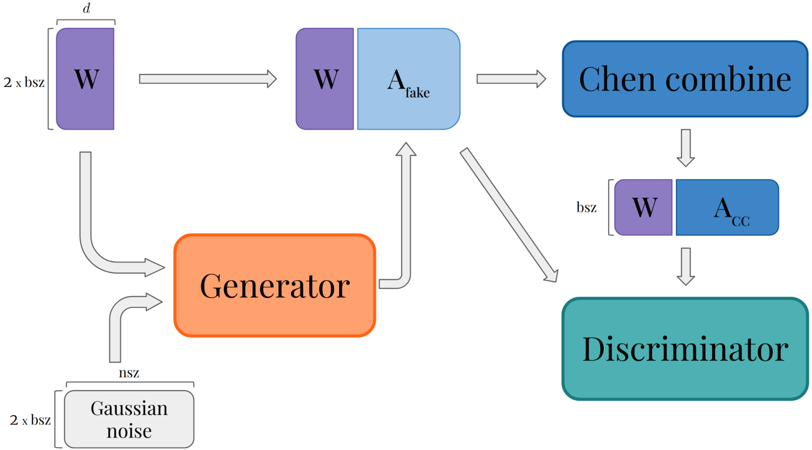


Figure: Schematic of the adversarial "Chen Training".

Outline

Numerical results in low dimensions

Architecture of generator

- Feed-forward NN with 3 hidden layers and 16 hidden dimensions.
- LeakyRelu activation function with slope = 0.01.
- Latent Gaussian noise with dimension 3.

Test Metric	LévyGAN	Foster [6]	Davie [14]	Fourier [18]
Computational time (seconds)	0.019	0.0071	0.002	3.1
Marginal 2-Wasserstein error (10^{-2})	$\mathbf{.246 \pm .013}$	$.254 \pm .010$	$2.03 \pm .013$	$.27 \pm 0.008$

Table: Computational times for generating 2^{20} samples with different models. The marginal error is calculated using Python Optimal Transport package [19].

Numerical results in low dimensions

We also estimate errors relating to the joint distribution of Lévy area.

d	Test Metric	LévyGAN	Foster [6]	Davie [14]
2	Fourth Moment	$.004 \pm .002$	$\mathbf{.002 \pm .002}$	$.042 \pm .001$
	Polynomial MMD (10^{-5})	$\mathbf{.341 \pm .070}$	$.654 \pm .131$	$.646 \pm .188$
	Gaussian MMD (10^{-6})	$1.47 \pm .125$	$\mathbf{1.44 \pm .128}$	$34.6 \pm .683$
3	Fourth Moment	$\mathbf{.004 \pm .002}$	$\mathbf{.004 \pm .002}$	$.043 \pm .001$
	Polynomial MMD (10^{-5})	$\mathbf{2.18 \pm .568}$	$2.30 \pm .732$	$2.26 \pm .773$
	Gaussian MMD (10^{-6})	$1.87 \pm .002$	$\mathbf{1.84 \pm .001}$	$16.3 \pm .001$
4	Fourth Moment	$\mathbf{.004 \pm .000}$	$.006 \pm .002$	$.043 \pm .002$
	Polynomial MMD (10^{-5})	$\mathbf{4.04 \pm .436}$	4.65 ± 1.31	$5.62 \pm .808$
	Gaussian MMD (10^{-6})	$\mathbf{1.90 \pm .001}$	$\mathbf{1.90 \pm .001}$	$263 \pm .003$

Outline

Achieving high order convergence in practice

We consider the Heston model (a popular SDE in mathematical finance):

$$dS_t = \mu S_t\, dt + \sqrt{\nu_t} S_t\, dW_t^{(1)},$$
$$d\nu_t = \kappa\big(\theta - \nu_t\big)\, dt + \sigma\sqrt{\nu_t}\, dW_t^{(2)},$$

where the parameters $\mu, \kappa, \theta, \sigma > 0$ satisfy Feller's condition $\sigma^2 < 2\kappa\theta$, which ensures the stochastic volatility term ν_t remains strictly positive.

Our goal is to compute the price of a call option at maturity time $T > 0$,

$$\mathbb{E}\big[\varphi(S_T)|(S_0, \nu_0)\big],$$

where the payoff φ is

$$\varphi(S_t) := e^{-rT}\max\big(S_T - K, 0\big),$$

with $r \in \mathbb{R}$ and $K > 0$ denoting the interest rate and strike price.

Achieving high order convergence in practice

Instead of Milstein's method, which only has $O(h)$ weak convergence, we use the "Strang-log-ODE" splitting [20] $\big($strong $O(h)$, weak $O(h^2)\big)$.

This involves generating an increment and Lévy area before solving a sequence of ODEs (which can be done explicitly for the Heston model). We use Multilevel Monte Carlo (MLMC) to achieve variance reduction.

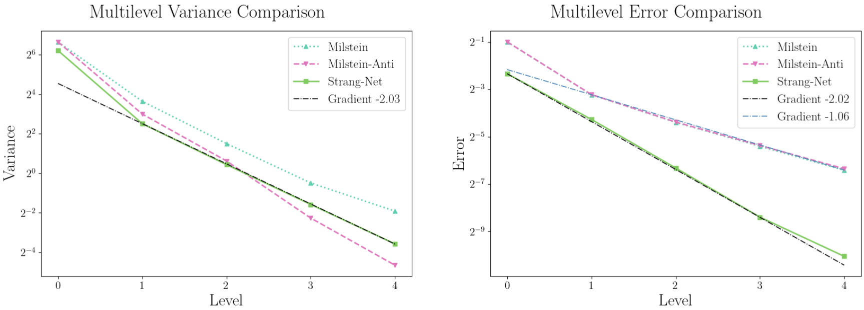


Figure: Variance and empirical error vs standard [21] and antithetic MLMC [22]

Achieving high order convergence in practice

But, “fake” Lévy area does introduce a fixed bias in the MLMC estimator.

To demonstrate this, we compare against Talay’s approximation [17], which simply replaces each $A^{(i,j)}_{t_k,t_{k+1}}$ with a Rademacher (scaled by $\frac{1}{2}h$).

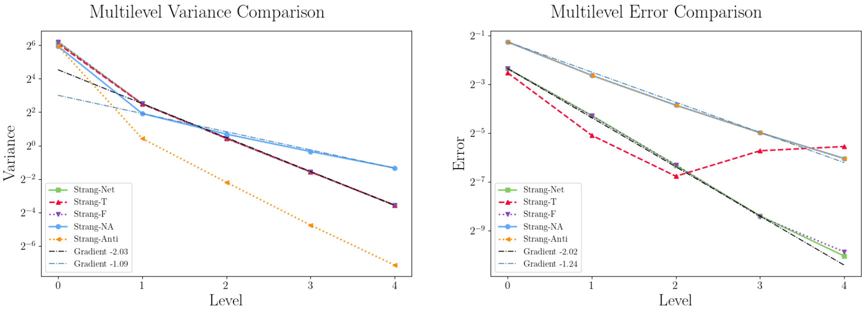


Figure: Variance and empirical error for different generative Lévy area models.

Achieving high order convergence in practice

Given a target accuracy of ε, MLMC with a LévyGAN-enabled high order method can achieve the desired error faster than a standard approach.

Target root mean squared error for a call option price	Time taken by Milstein with antithetic MLMC (seconds)	Time taken by Strang log-ODE with LévyGAN (seconds)
0.1	$\mathbf{0.0097}$	0.0102
0.0441	0.0256	$\mathbf{0.0128}$
0.0129	0.376	$\mathbf{0.142}$
0.0086	1.03	$\mathbf{0.311}$
0.0057	2.86	$\mathbf{0.806}$
0.0038	8.63	$\mathbf{2.25}$
0.0025	23.6	$\mathbf{5.83}$

Outline

Conclusion and future work

Conclusion

- We introduce "LévyGAN", a Generative Adversarial Network (GAN) for generating the Lévy area of multidimensional Brownian motion.
- We incorporate symmetries of Lévy area into the model & training.
- We show that LévyGAN can lead to high order weak convergence and variance reduction in practice (that is, the MLMC bias is small).

Future work

- A learnt "Lévy construction" for Brownian motion and its Lévy area. This would enable higher order adaptive solvers for general SDEs.
- "Space-space-time" Lévy area [23] (e.g. for SDEs with scalar noise).

Thank you
for your attention!

and our preprint can be found at:

A. Jelinčič, J. Tao, W. F. Turner, T. Cass, J. Foster and H. Ni, *Generative Modelling of Lévy Area for High Order SDE Simulation*, arxiv:2308.02452, 2023.

Outline

References I

J. Cox, J. Ingersoll and S. Ross. *A Theory of the Term Structure of Interest Rates*, Econometrica, vol. 53, no. 2, pp. 385–407, 1985.

M. Welling and Y. W. Teh. *Bayesian Learning via Stochastic Gradient Langevin Dynamics*, Proceedings of the 28th ICML, 2011.

J. M. C. Clark and R. J. Cameron.*The maximum rate of convergence of discrete approximations for stochastic differential equations*. In Stochastic Differential Systems Filtering & Control. Springer, 1980.

A. S. Dickinson.*Optimal Approximation of the Second Iterated Integral of Brownian Motion*. Stochastic Analysis and Applications. vol. 25, no. 5, pp. 1109–1128, 2007.

G. N. Milstein and M. V. Tretyakov. *Stochastic Numerics for Mathematical Physics (second edition)*, Springer, 2021.

References II

- J. Foster. *Numerical approximations for stochastic differential equations*, University of Oxford, 2020.
- J. G. Gaines and T. Lyons. *Random Generation of Stochastic Area Integrals*, SIAM Journal on Applied Mathematics, vol. 54, no. 4, pp. 1132–1146, 1994.
- P. E. Kloeden and E. Platen. *Numerical Solution of Stochastic Differential Equations*, Springer, 1992.
- G. N. Milstein. *Numerical Integration of Stochastic Differential Equations*, volume 313. Springer Science & Business Media, 1994.
- P. E. Kloeden, E. Platen, and I. W. Wright. *The approximation of multiple stochastic integrals*. Stochastic Analysis and Applications, vol. 10, no. 4, pp. 431–441, 1992.

References III

- J. Foster and K. Habermann. *Brownian bridge expansions for Lévy area approximations and particular values of the Riemann zeta function*. Combinatorics, Probability and Computing, vol. 32, no. 3, pp. 370–397, 2023.

- P. F. Kuznetsov. *A method of expansion and approximation of repeated stochastic Stratonovich integrals based on multiple Fourier series on full orthonormal systems*. [In Russian]. Electronic Journal "Differential Equations and Control Processes", vol. 1, pp. 18–77, 1997.

- J. Mrongowius and A. Rößler. *On the approximation and simulation of iterated stochastic integrals and the corresponding Lévy areas in terms of a multidimensional Brownian motion*. Stochastic Analysis and Applications, vol. 40, no. 3, pp. 397–425, 2022.

References IV

A. Davie. *KMT theory applied to approximations of SDE*. In Stochastic Analysis and Applications, Springer Proceedings in Mathematics and Statistics, vol. 100, pp. 185–201, Springer, 2014.

G. Flint and T. Lyons.*Pathwise approximation of SDEs by coupling piecewise abelian rough paths*. arxiv:1505.01298, 2015.

M. Wiktorsson. *Joint characteristic function and simultaneous simulation of iterated Itô integrals for multiple independent Brownian motions*. Annals of Applied Probability, vol. 11, no. 2, pp. 470–487, 2001.

D. Talay. *Second-order discretization schemes of stochastic differential systems for the computation of the invariant law*. Stochastics and Stochastic Reports, vol. 29, no. 1, pp. 13–36, 1990.

References V

- F. Kastner and A. Rößler. *An Analysis of Approximation Algorithms for Iterated Stochastic Integrals and a Julia and MATLAB Simulation Toolbox*. Numerical Algorithms, vol. 93, pp. 27–66, 2023.
- R. Flamary et al. *POT: Python Optimal Transport*. Journal of Machine Learning Research, vol. 22, no. 78, 2021.
- J. Foster, G. dos Reis and C. Strange. *High order splitting methods for SDEs satisfying a commutativity condition*, SIAM Journal on Numerical Analysis, vol. 62, no. 1, pp. 500–532, 2024.
- M. Giles. *Improved Multilevel Monte Carlo Convergence using the Milstein Scheme*. Monte Carlo and Quasi-Monte Carlo Methods, Springer, 2006.

References VI

M. B. Giles, L. Szpruch. *Antithetic multilevel Monte Carlo estimation for multi-dimensional SDEs without Lévy area simulation*. Annals of Applied Probability, vol. 24, no. 4, pp. 1585–1620, 2014.

J. Foster, T. Lyons and H. Oberhauser. *An optimal polynomial approximation of Brownian motion*, SIAM Journal on Numerical Analysis, vol. 58, no. 3, pp. 1393–1421, 2020.

A. Jelinčič, J. Tao, W. F. Turner, T. Cass, J. Foster and H. Ni, *Generative Modelling of Lévy Area for High Order SDE Simulation*, arxiv:2308.02452, 2023.