From $1/\sqrt{n}$ to $1/n$: Accelerating SDE Simulation with Cubature Formulae

Peter Koepernik[1,⋆] Thomas Coxon[2,⋆] James Foster[3]

[1]University of Oxford [2]Loughborough University [3]University of Bath

[*]Equal contribution

23 June 2026

Setting

Consider an SDE

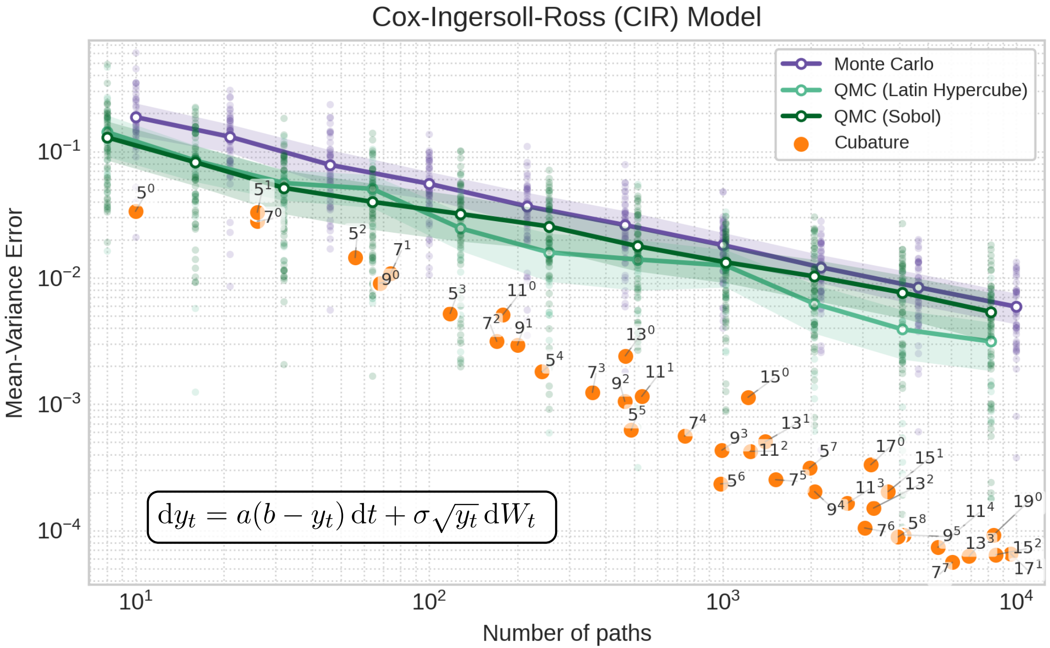
$$\begin{aligned} \mathrm{d}y_t &= \mu(y_t)\,\mathrm{d}t + \sigma(y_t)\,\mathrm{d}W_t, \\ y_0 &= \xi, \end{aligned} \tag{1}$$

where y_t takes values in $\mathbb{R}^e$, and $\mu, \sigma \colon \mathbb{R}^e \to \mathbb{R}^e$ are the *drift* and *diffusion*, respectively.

Statistics like $\mathbb{E}[f(y_T)]$ for some $f \colon \mathbb{R}^e \to \mathbb{R}$ are traditionally estimated with Monte-Carlo:

$$\mathbb{E}[f(y_T)] \approx \frac{1}{M} \sum_{i=1}^{M} f(y_T(\omega_i)) \quad \longleftarrow \text{error } O\left(1/\sqrt{M}\right)$$

Here $y_T(\omega) \in \mathbb{R}^e$ denotes the solution at time T of (1) driven by a path $\omega \colon [0,T] \to \mathbb{R}$, and ω_i are independent samples of Brownian motion.



Cox-Ingersoll-Ross (CIR) Model
Monte Carlo
QMC (Latin Hypercube)
QMC (Sobol)
Cubature
Mean-Variance Error
Number of paths
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{1}
10^{2}
10^{3}
10^{4}
$\mathrm{d}y_t = a(b - y_t)\,\mathrm{d}t + \sigma\sqrt{y_t}\,\mathrm{d}W_t$

Cubatures in Numerical Integration

Numerical integration of a function $f\colon [0,1] \to \mathbb{R}$ can also be done with Monte-Carlo:

$$\int_0^1 f(x)\,\mathrm{d}x = \mathbb{E}[f(U)] \approx \frac{1}{M}\sum_{i=1}^{M} f(u_i)$$

for $U \sim \mathcal{U}([0,1])$ and independent samples $u_i \sim \mathcal{U}([0,1])$. The error is $O(1/\sqrt{M})$.

One can do much better with uniformly spaced points, which have error $O(1/M)$, and even better with *cubature formulae*: deterministic collections $(\lambda_i, u_i)_{i=1}^{M}$ of points $u_i \in [0,1]$ with weights $\lambda_i \geq 0$, $\sum_i \lambda_i = 1$, that are chosen in such a way that the quadrature approximation

$$\int_0^1 f(x)\,\mathrm{d}x \approx \sum_{i=1}^{M} \lambda_i f(u_i)$$

is exact for polynomials f with degree up to some $D \in \mathbb{N}$, the *degree* of the cubature.

Cubatures on Wiener Space

Can we do something similar in the SDE setting?

$$\mathbb{E}\big[f(y_T)\big] \approx \sum_{i=1}^{M} \lambda_i f\big(y_T(\omega_i)\big),$$

where the i.i.d. paths $\omega_i\colon [0,T] \to \mathbb{R}$ and uniform weights $\lambda_i = 1/M$ are replaced by a *deterministic* collection $(\lambda_i, \omega_i)_{i=1}^{M}$ of paths ω_i and weights $\lambda_i \geq 0$ that matches the expected Brownian "*signature*" up to some degree D:

$$\sum_{i=1}^{M} \lambda_i \, \mathrm{S}_{0,T}^{(k)}(\omega_i) = \mathbb{E}[\mathrm{S}_{0,T}^{(k)}(W)]$$

for all $k \in \{0, \ldots, D\}$.

This is called a *Cubature on Wiener space* of degree D [Lyons and Victoir, 2004].

Cubatures on Wiener Space

The signature is made up of terms of the form

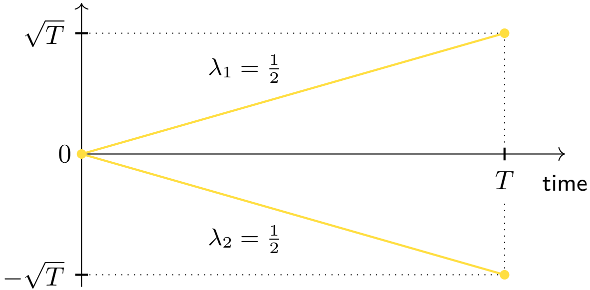
$$\mathrm{S}_{0,T}^{(i_1 \dots i_k)}(\omega) = \int_{0<t_1<\dots<t_k<T} \mathrm{d}\omega^{(i_1)}(t_1) \dots \mathrm{d}\omega^{(i_k)}(t_k),$$

where $\omega^{(0)}(t) = t$ and $\omega^{(1)}(t) = \omega(t)$. The *degree* of the term is the number of indices, where zeros are counted twice.

- $\mathrm{S}^{(1)}(\omega) = \{\int_0^T \mathrm{d}\omega_t\}$
- $\mathrm{S}^{(2)}(\omega) = \{\int_0^T \mathrm{d}\omega_{t_1} \mathrm{d}\omega_{t_2}, \int_0^T \mathrm{d}t\}$
- $\mathrm{S}^{(3)}(\omega) = \{\int_0^T \mathrm{d}\omega_{t_1} \mathrm{d}\omega_{t_2} \mathrm{d}\omega_{t_3}, \int_0^T \mathrm{d}t_1 \mathrm{d}\omega_{t_2}, \int_0^T \mathrm{d}\omega_{t_1} \mathrm{d}t_2\}$
- $\dots$

Cubatures on Wiener Space

The smallest non-trivial cubature has degree 3 and consists of two paths:



We call it the “Rademacher cubature”.

Cubatures on Wiener Space

It is known that cubatures of arbitrarily high degree exist [Lyons and Victoir, 2004], but they are difficult to construct. Known cubatures:

- *Degree 3:* Rademacher cubature
- *Degree 5:* [Lyons and Victoir, 2004]
- *Degree 7:* [Hayakawa and Tanaka, 2022, Malyarenko and Nohrouzian, 2022]
- *Degree 9+:* [Koepernik et al., 2026][1] (our work)

The error of the cubature approximation to $\mathbb{E}[f(y_h)]$ is of order $O(h^{(D+1)/2})$ for small h [Lyons and Victoir, 2004] $\to$ subdivide $[0,T]$ into N subintervals and apply the cubature on each of them $\to$ total number of summands $M = |\text{cubature}|^N$, and

$$\text{global error} = O\Big(N^{-(D-1)/2}\Big) = O\Big((\log M)^{-(D-1)/2}\Big) \gg O\Big(1/\sqrt{M}\Big)$$

Monte-Carlo Error

[1] Degree 9 technically constructed by [Gyurkó and Lyons, 2011] but not practically usable

Our Contributions

Our contributions are threefold:

(i) We introduce a scalable algorithm that automatically constructs cubature formulae of arbitrary degree. We obtain cubatures with **degree up to 19** within hours on modest hardware.

(ii) We prove that the error of the cubature approximation goes to zero *globally* on $[0, T]$ as the degree $D \to \infty$ (under some assumptions).

(iii) We demonstrate empirically that the cubature approximation error with our cubatures (applied globally) is of order $O(1/M)$.

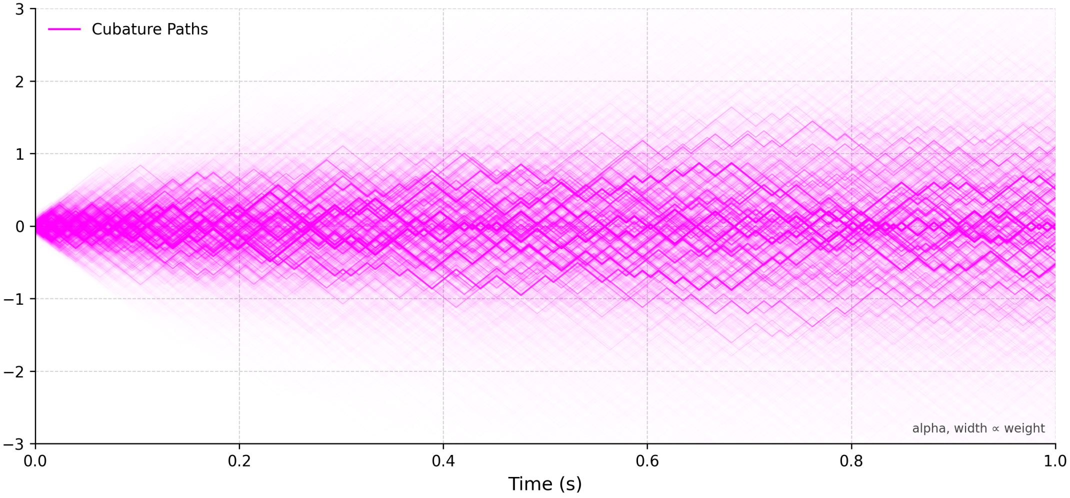


Figure: Degree 17 cubature, consisting of 3194 paths.

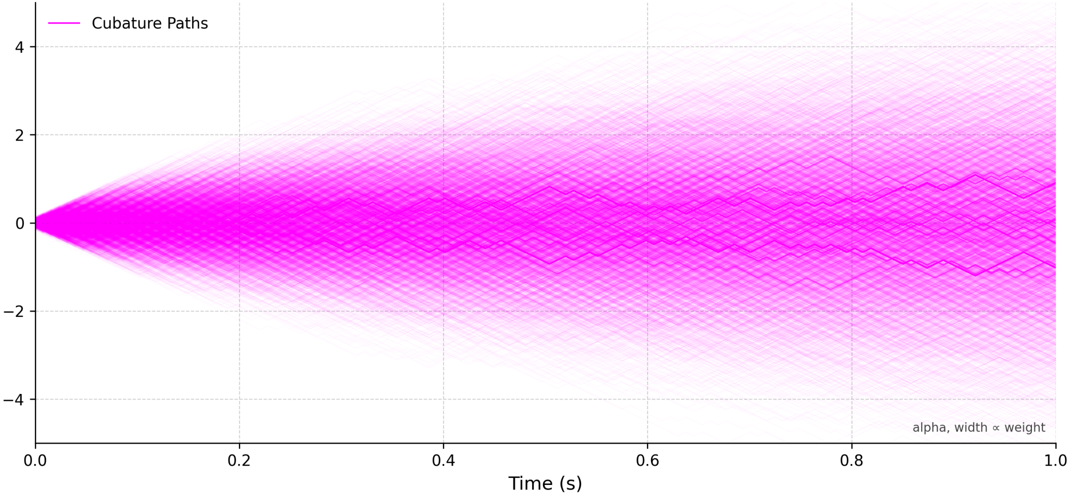


Figure: Degree 19 cubature, consisting of 8362 paths.

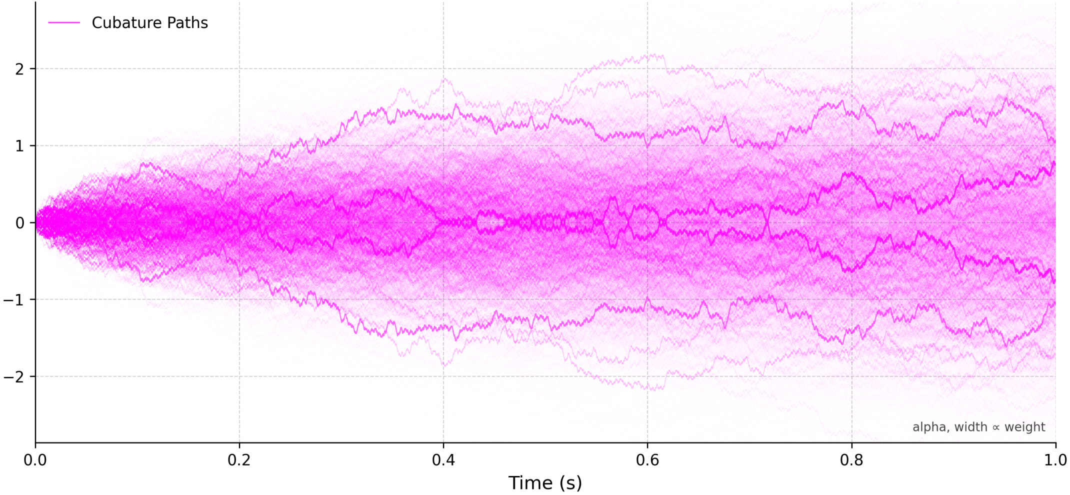
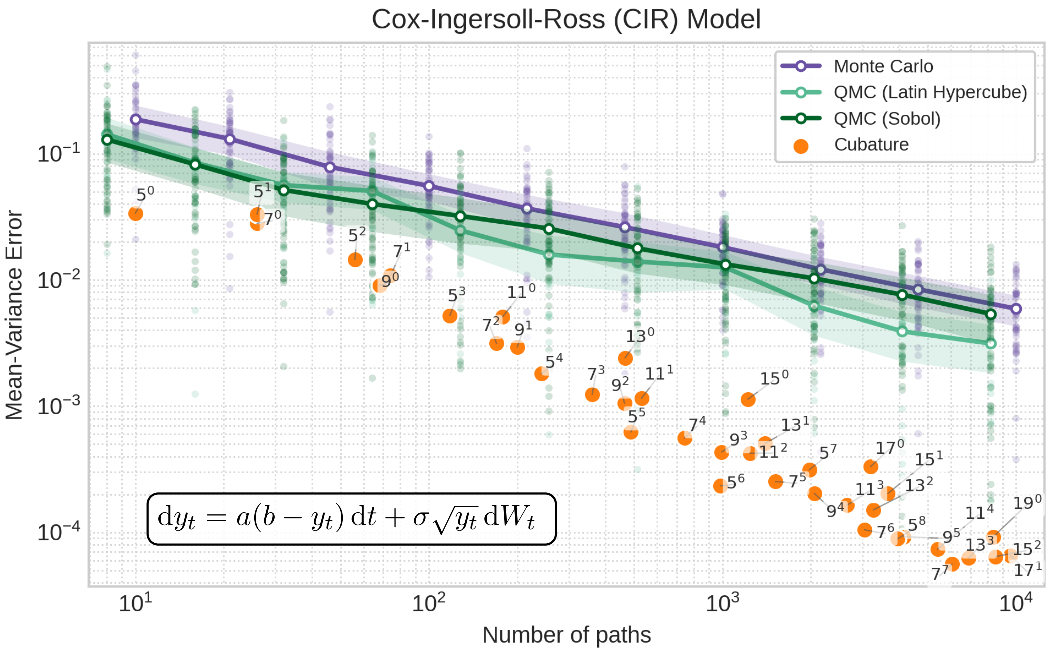
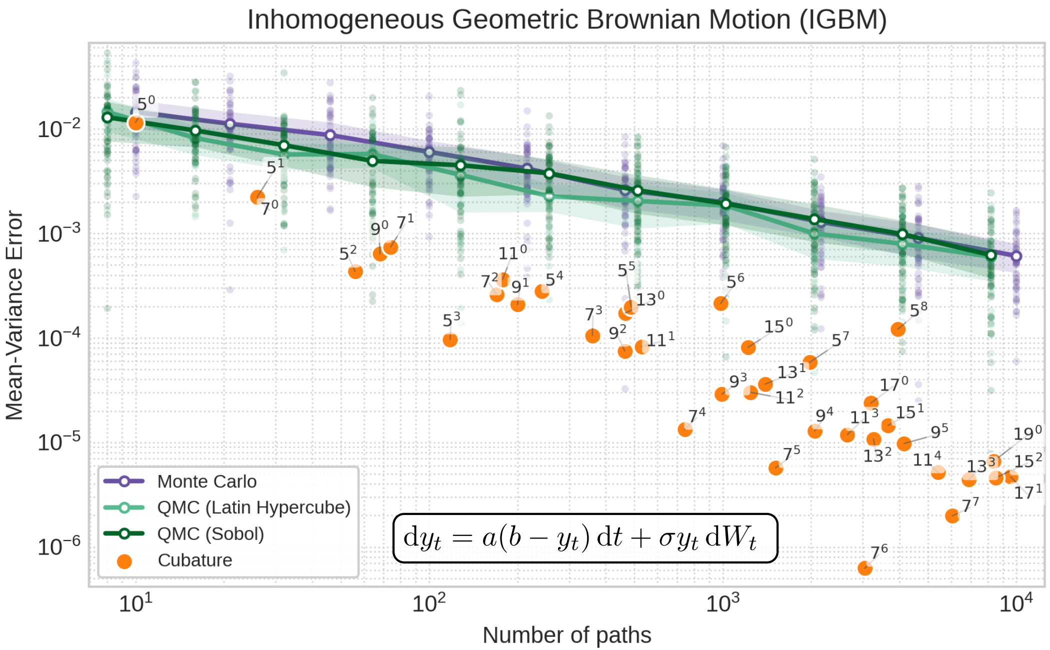


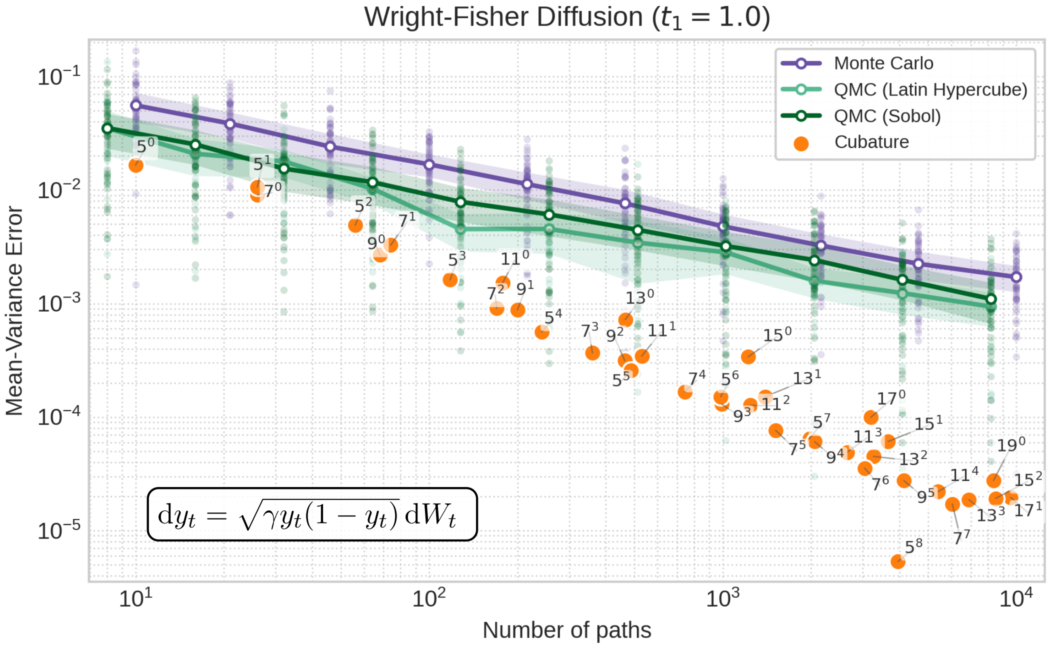
Figure: Degree 7 cubature (with "dyadic depth" 7).

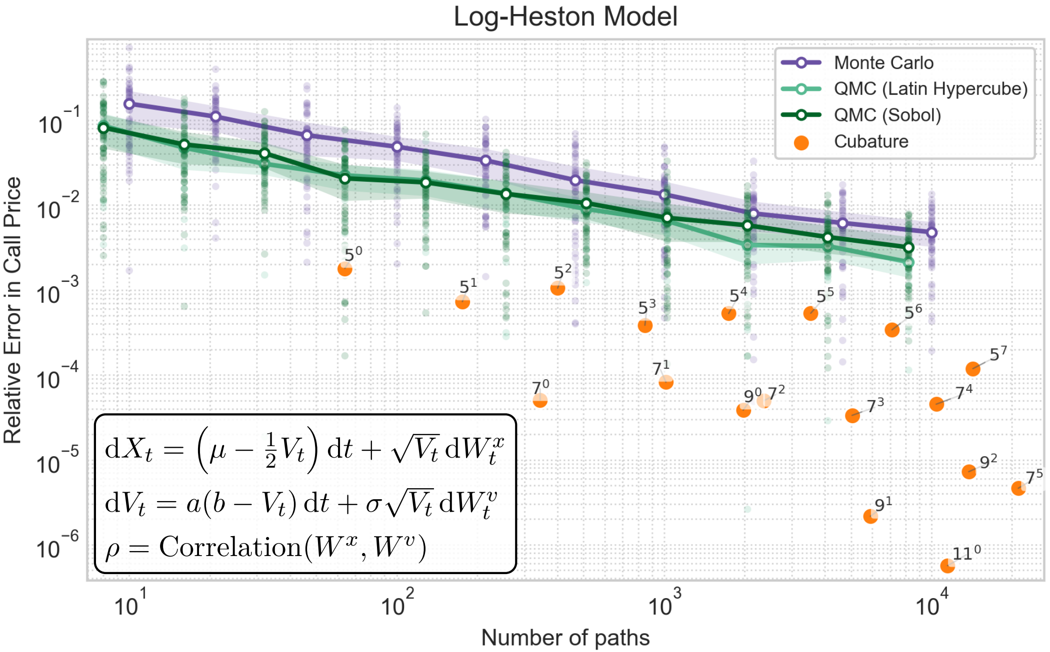


Cox-Ingersoll-Ross (CIR) Model
Monte Carlo
QMC (Latin Hypercube)
QMC (Sobol)
Cubature
Mean-Variance Error
Number of paths
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{1}
10^{2}
10^{3}
10^{4}
$\mathrm{d}y_t = a(b - y_t)\,\mathrm{d}t + \sigma\sqrt{y_t}\,\mathrm{d}W_t$

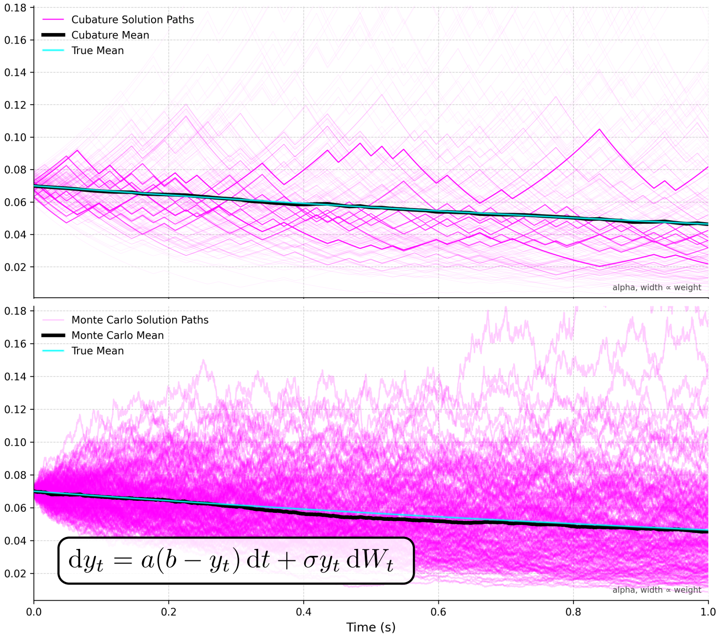


Inhomogeneous Geometric Brownian Motion (IGBM)
Mean-Variance Error
Number of paths
Monte Carlo
QMC (Latin Hypercube)
QMC (Sobol)
Cubature
$\mathrm{d}y_t = a(b - y_t)\,\mathrm{d}t + \sigma y_t\,\mathrm{d}W_t$


Wright-Fisher Diffusion ($t_1 = 1.0$)
Monte Carlo
QMC (Latin Hypercube)
QMC (Sobol)
Cubature
Mean-Variance Error
Number of paths
$\mathrm{d}y_t = \sqrt{\gamma y_t(1-y_t)}\,\mathrm{d}W_t$



Log-Heston Model
Monte Carlo
QMC (Latin Hypercube)
QMC (Sobol)
Cubature
Relative Error in Call Price
Number of paths
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{1}
10^{2}
10^{3}
10^{4}
5^0
5^1
5^2
5^3
5^4
5^5
5^6
5^7
7^0
7^1
7^2
7^3
7^4
7^5
9^0
9^1
9^2
11^0
$\mathrm{d}X_t = \left(\mu - \frac{1}{2}V_t\right)\mathrm{d}t + \sqrt{V_t}\,\mathrm{d}W_t^x$
$\mathrm{d}V_t = a(b - V_t)\,\mathrm{d}t + \sigma\sqrt{V_t}\,\mathrm{d}W_t^v$
$\rho = \mathrm{Correlation}(W^x, W^v)$



Cubature Solution Paths
Cubature Mean
True Mean
alpha, width ∝ weight
Monte Carlo Solution Paths
Monte Carlo Mean
True Mean
alpha, width ∝ weight
$\mathrm{d}y_t = a(b - y_t)\,\mathrm{d}t + \sigma y_t\,\mathrm{d}W_t$
0.18
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.0
0.2
0.4
0.6
0.8
1.0
Time (s)

Step 1: Construction of Approximate Cubature

A discrete random walk on $[0, T]$ with N steps of sizes $\pm\sqrt{T/N}$ is an approximate cubature of size 2^N with error $O(1/N)$ (for fixed degree D).

Key observation: The expected signature of this random walk does not change up to degree D if its N increments are only D-*wise* independent $\rightarrow$ binary orthogonal arrays.

Outcome: Approximate cubature of size $N^{\lfloor D/2 \rfloor}$ with error $O(1/N)$.

Step 2: Recombination

Recombination is an algorithm that takes a set of M points $x_i \in \mathbb{R}^m$ with probability weights $(\lambda_i)_{i=1}^M$, and returns a subset $(x_i)_{i\in I}$ of points with probability weights $\bar{\lambda}_i$ of size $|I| \leq m+1$ such that

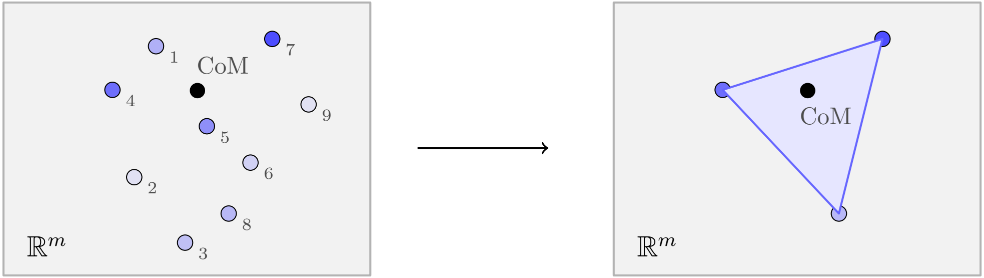
$$\sum_{i\in I} \bar{\lambda}_i x_i = \sum_{i=1}^{M} \lambda_i x_i.$$

This algorithm is highly parallelisable and can be efficiently implemented on GPU. Thinking of the paths making up the approximate cubature as vectors of signature terms, we have

$$m(D) = |\{I \in \{0,1\}^* \colon I \text{ has degree at most } D\}| \in [2^{(D+1)/2}, 2^{D+1}].$$

Outcome: Approximate cubature of size at most $\underbrace{m(D)+1}_{=:M(D)}$ with error $O(1/N)$.

Step 2: Recombination



Step 3: Sharpening

Finally, we attempt to modify the weights of the approximate cubature to make it an exact cubature. Abstractly, the task is "*given points $x_i \in \mathbb{R}^m$ and a point $p \in \mathbb{R}^m$, does p lie in the convex hull of $(x_i)_{i=1}^M$, and if so what are the weights?*". Let $X \in \mathbb{R}^{m \times M}$ be the matrix with the x_i as columns.

(i) **Solve $\min_{\boldsymbol{\lambda}} \|X\boldsymbol{\lambda} - p\|$ without non-negativity restriction:** Every $\boldsymbol{\lambda} \in \mathbb{R}^M$ with $\sum_{i=1}^M \lambda_i = 1$ that minimizes $\|X\boldsymbol{\lambda} - p\|$ can be written as $\boldsymbol{\lambda}_0 + Vz$ for some $z \in \mathbb{R}^k$, where $k \in \mathbb{N}$, $V \in \mathbb{R}^{M \times k}$, and $\boldsymbol{\lambda}_0 \in \mathbb{R}^M$ can be computed efficiently using the SVD of X. If $\min_{\boldsymbol{\lambda}} \|X\boldsymbol{\lambda} - p\| > 0$ then we failed, otherwise proceed to step (ii).

(ii) **Find non-negative solution:** For example, find $z^* \in \mathbb{R}^k$ with $\boldsymbol{\lambda}^* = \boldsymbol{\lambda}_0 + Vz^* \geq 0$ by doing gradient descent on the "non-negativity loss" $\ell(z) = \sum_{i=1}^M \max(0, -z_i)^2$. We also implemented an alternating projections method – which was much faster.

Outcome: Exact cubature of size at most $M(D)$.

Overview

Degree	Minimum N	Step 1 Output Size		Steps 2 & 3 Output Size
D	N	naive 2^N	actual $N^{\lfloor D/2 \rfloor}$	$M(D)$
5	32	4.3×10^9	1.0×10^3	10
7	32	4.3×10^9	3.3×10^4	26
9	64	1.8×10^{19}	1.7×10^7	68
11	64	1.8×10^{19}	1.1×10^9	178
13	64	1.8×10^{19}	6.9×10^{10}	466
15	64	1.8×10^{19}	4.4×10^{12}	1220
17	64	1.8×10^{19}	2.8×10^{14}	3194
19	128?	3.4×10^{38}	9.2×10^{18}	8362

Optional Step 4: Dyadic Construction

Given a cubature $(\lambda_i, \omega_i)_{i=1}^M$,

(i) Set $\tilde{\omega}_i(t) = \frac{1}{\sqrt{2}}\omega_i(2t)$ (rescales the paths to the interval $[0, T/2]$)

(ii) Build the *product cubature* $(\lambda_i\lambda_j, \tilde{\omega}_i \odot \tilde{\omega}_j)_{i,j=1}^M$, where $\odot$ denotes concatenation.

$\longrightarrow$ It has size M^2 and matches the expected Brownian signature on $[0, T]$ as well as $[0, T/2]$ and $[T/2, T]$.

(iii) Apply recombination to the product cubature.

This can be repeated an arbitrary number of times. We call a cubature that matches the expected Brownian signature up to degree D on all dyadic intervals of lengths $2^{-l}T$ for $l \leq m$ a cubature with degree D and *dyadic depth* m.

Theoretical Results

Definition

Let $(\lambda_i^n, \omega_i^n)$ be a sequence of cubatures. Denote by y^n is the solution to an SDE driven by $(\lambda_i^n, \omega_i^n)$, and by y the solution to the same SDE driven by Brownian motion. Then we call the cubature(s) *asymptotically consistent* (for that SDE) if $y^n \to y$ in distribution in the space of continuous paths.

We conjecture that this is true for the cubatures constructed by our algorithm under no assumptions other than that the degree $D_n \to \infty$.

Theoretical Results

Theorem

Suppose $(\lambda_i^n, \omega_i^n)$ *is a cubature constructed by our algorithm with degree* $D_n \to \infty$, *and dyadic depth* $m_n \in \mathbb{N}$ *such that*

$$m_n \geq \left(\frac{3}{4} + \varepsilon\right) \log_2 N_n - C$$

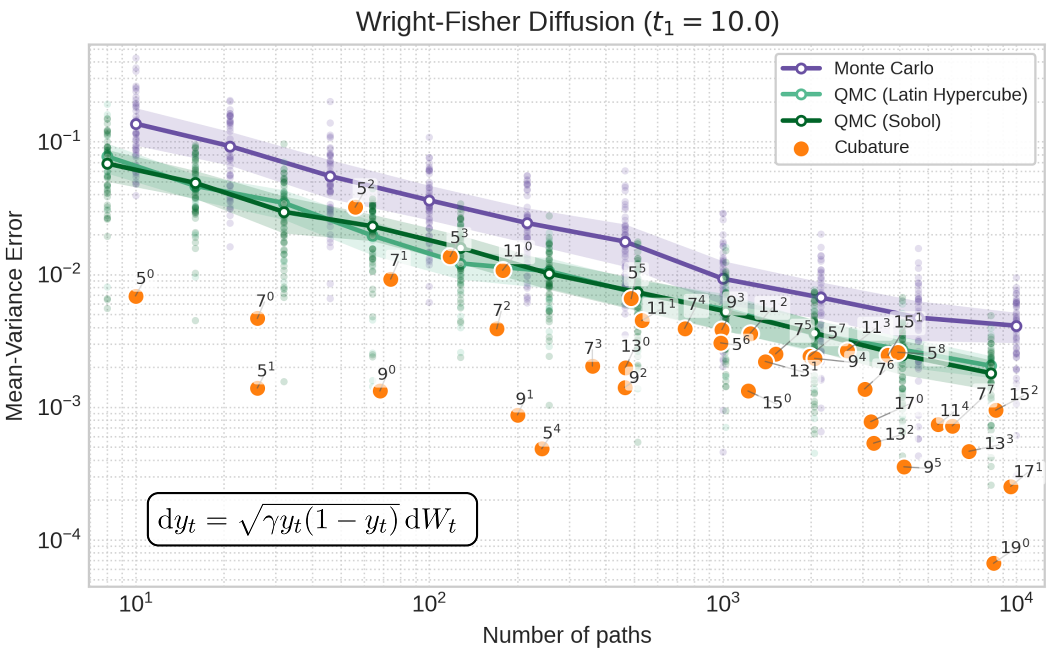
for some $\varepsilon, C > 0$. *Then the cubature is asymptotically consistent.*

Theorem

Let $(\lambda_i^n, \omega_i^n)$ *be the approximate cubature resulting from step 1 of the algorithm, prior to recombination and sharpening, with* $N_n \to \infty$ *and degree* $D_n \to \infty$. *Then it is asymptotically consistent.*

Limitations

- Cubature approximation can become inaccurate for non-smooth vector fields or very long time intervals.
- Algorithm technically works in any dimension but scales poorly, making it impractical for high-dimensional Brownian motion.
  Recombination scales $\mathcal{O}\left(nm + \log_2\left(\frac{n}{2m}\right)m^3\right)$, where n is the number of paths and $m = m(D)$.


Wright-Fisher Diffusion ($t_1 = 10.0$)
Mean-Variance Error
Number of paths
Monte Carlo
QMC (Latin Hypercube)
QMC (Sobol)
Cubature
$\mathrm{d}y_t = \sqrt{\gamma y_t(1 - y_t)}\,\mathrm{d}W_t$
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{1}
10^{2}
10^{3}
10^{4}

References I

Gyurkó, L. G. and Lyons, T. (2011).
Efficient and Practical Implementations of Cubature on Wiener Space.
In *Stochastic Analysis 2010*, pages 73–111. Springer Berlin Heidelberg.

Hayakawa, S. and Tanaka, K. (2022).
Monte Carlo construction of cubature on Wiener space.
Japan Journal of Industrial and Applied Mathematics, 39(2):543–571.

Koepernik, P., Coxon, T., and Foster, J. (2026).
ARCANE: Scalable high-degree cubature formulae for simulating SDEs without Monte Carlo error.
arxiv.org/abs/2602.17151.

References II

Lyons, T. and Victoir, N. (2004).
Cubature on Wiener Space.
Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 460(2041):169–198.

Malyarenko, A. and Nohrouzian, H. (2022).
Testing Cubature Formulae on Wiener Space Versus Explicit Pricing Formulae.
In *Stochastic Processes, Statistical Methods, and Engineering Mathematics*, pages 223–248, Cham. Springer International Publishing.