Dipartimento di Matematica,
Università di Bologna

The Itô-stochastic Magnus expansion for state-dependent SPDEs with two space variables

Kevin Kamm

April 19, 2022

Supervisor (Industry)
Dr. Luca Caputo, Banco Santander

Supervisor (University)
Prof. Dr. Andrea Pascucci, Bologna

Goal

Today, we are developing an efficient numerical method to solve the following SPDE.

$$\begin{cases} \Big(-\partial_t u_t(x,v) + h(x,v)u_t(x,v) + f^x(x,v)\partial_x u_t(x,v) + f^v(x,v)\partial_v u_t(x,v) \\ \quad + \frac{1}{2}g^{xx}(x,v)\partial_{xx}u_t(x,v) + g^{xv}(x,v)\partial_{xv}u_t(x,v) + \frac{1}{2}g^{vv}(x,v)\partial_{vv}u_t(x,v)\Big)dt \\ \quad + \left(\sigma^x(x,v)\partial_x u_t(x,v) + \sigma^v(x,v)\partial_v u_t(x,v)\right)dW_t = 0, \\ u_0(x,v) = \varphi(x,v), \end{cases} \tag{SPDE}$$

where the coefficient functions are chosen such that there exists a unique strong solution.

Goal

We will see that after discretizing (SPDE) in space and vectorizing the equation we can find matrices $A, B \in \mathbb{R}^{d^2,d^2}$ such that

$$dU_t = BU_t dt + AU_t dW_t, \quad U_0 = \operatorname{vec}(\Phi) \in \mathbb{R}^{d^2,1}, \tag{SDE}$$

which can be solved by first computing the Magnus expansion for

$$dX_t = BX_t dt + AX_t dW_t, \quad X_0 = I_{d^2}$$

and multiplying it with the discretized and vectorized initial datum, i.e.

$$U_t = X_t U_0.$$

Undoing the vectorization of U_t yields a numerical approximation of (SPDE).

Table of Contents

Magnus expansion
- Heuristical derivation
- Expansion formulas

SPDE
- Space discretization
- Vectorization
- The stochastic Langevin equation

Numerics
- Sparsity
- Errors and Computational Times

Magnus expansion

Heuristical derivation
Expansion formulas

Idea

Solve the matrix-valued SDE

$$dX_t = BX_t dt + AX_t dW_t, \quad X_0 = I_d$$

by assuming that there exists a solution $X_t = \exp(Y_t)$ for small times $t > 0$ depending on a stopping time and

$$Y_t = \int_0^t \mu(Y_s)\, ds + \int_0^t \sigma(Y_s)\, dW_s, \quad Y_0 = 0_{\mathbb{R}^{d\times d}}.$$

Determine μ and σ

$$dX_t = BX_t + AX_t dW_t$$ ← Equation

Determine μ and σ

$$
\begin{aligned}
dX_t &= BX_t + AX_t dW_t && \leftarrow \text{Equation} \\
&= B\exp(Y_t) + A\exp(Y_t)\, dW_t && \leftarrow \text{Assumption}
\end{aligned}
$$

Determine μ and σ

$$
\begin{aligned}
dX_t &= BX_t + AX_t dW_t && \leftarrow \text{Equation} \\
&= B\exp(Y_t) + A\exp(Y_t)\, dW_t && \leftarrow \text{Assumption} \\
&= d\exp(Y_t) && \leftarrow \text{Assumption}
\end{aligned}
$$

Determine μ and σ

$$
\begin{aligned}
dX_t &= BX_t + AX_t dW_t && \leftarrow \text{Equation} \\
&= B\exp(Y_t) + A\exp(Y_t)\, dW_t && \leftarrow \text{Assumption} \\
&= d\exp(Y_t) && \leftarrow \text{Assumption} \\
&= \left(\mathcal{L}_{Y_t}(\mu(Y_s)) + \frac{1}{2}\mathcal{Q}_{Y_t}(\sigma(Y_t), \sigma(Y_t))\right)\exp(Y_t)\, dt && \leftarrow \text{Itô's formula} \\
&\quad + \mathcal{L}_{Y_t}(\sigma(Y_t))\exp(Y_t)\, dW_t.
\end{aligned}
$$

Determine μ and σ

$$\begin{aligned} dX_t &= BX_t + AX_t dW_t & \leftarrow \text{Equation} \\ &= B\exp(Y_t) + A\exp(Y_t)\, dW_t & \leftarrow \text{Assumption} \\ &= d\exp(Y_t) & \leftarrow \text{Assumption} \\ &= \left(\mathcal{L}_{Y_t}(\mu(Y_s)) + \frac{1}{2}\mathcal{Q}_{Y_t}(\sigma(Y_t), \sigma(Y_t))\right)\exp(Y_t)\, dt & \leftarrow \text{Itô's formula} \\ &\quad + \mathcal{L}_{Y_t}(\sigma(Y_t))\exp(Y_t)\, dW_t. \end{aligned}$$

A comparison of coefficients yields

$$\begin{aligned} B &\stackrel{!}{=} \mathcal{L}_{Y_t}(\mu(Y_t)) + \frac{1}{2}\mathcal{Q}_{Y_t}(\sigma(Y_t), \sigma(t, Y_t)) \\ A &\stackrel{!}{=} \mathcal{L}_{Y_t}(\sigma(Y_t)). \end{aligned}$$

Determine μ and σ

Inverting $\mathcal{L}_Y$ by using Baker's lemma yields

$$\sigma\left(Y_t\right) \equiv \sum_{n=0}^{\infty} \frac{\beta_n}{n!} \mathrm{ad}_{Y_t}^n\left(A\right) \tag{1.1}$$

$$\mu\left(Y_t\right) \equiv \sum_{k=0}^{\infty} \frac{\beta_k}{k!} \mathrm{ad}_{Y_t}^k \left(B - \frac{1}{2} \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} \frac{\mathrm{ad}_{Y_t}^n\left(\sigma\left(Y_t\right)\right)}{(n+1)!} \frac{\mathrm{ad}_{Y_t}^m\left(\sigma\left(Y_t\right)\right)}{(m+1)!} + \frac{\left[\mathrm{ad}_{Y_t}^n\left(\sigma\left(Y_t\right)\right), \mathrm{ad}_{Y_t}^m\left(\sigma\left(Y_t\right)\right)\right]}{(n+m+2)(n+1)!m!} \right) \tag{1.2}$$

where β_n denote the Bernoulli numbers, e.g. $\beta_0 = 1$, $\beta_1 = -\frac{1}{2}$, $\beta_2 = \frac{1}{6}$, $\beta_3 = 0$ and $\beta_4 = -\frac{1}{30}$.

Solve the SDE by Picard-iteration

Now, we solve the SDE for Y_t by Picard-iteration

$$Y_t^n = \int_0^t \mu\left(Y_s^{n-1}\right) ds + \int_0^t \sigma\left(Y_s^{n-1}\right) dW_s. \tag{1.3}$$

In order to derive the Magnus expansion formulas, we will introduce some bookkeeping parameters $\varepsilon, \delta > 0$ and substitute A by εA, as well as B by δB. Henceforth, we will denote the n-th order Picard iteration with the substitution by $Y_t^{n,\varepsilon,\delta}$.

Order 1

Notice that the zero matrix commutes with all matrices and therefore by definition of $\mathrm{ad}_Y^0(A) = A$ we have

$$\sigma\left(Y_t^{0,\varepsilon,\delta}\right) = \varepsilon A.$$

Inserting this into the formula for μ yields

$$\mu\left(Y_t^{0,\varepsilon,\delta}\right) = \delta B - \frac{1}{2}\varepsilon^2 A^2,$$

because A commutes with itself as well.
Since the Itô-correction term is of order ε^2, it will not be part of the first-order Magnus expansion and we have

$$Y_t^1 = \int_0^t B ds + \int_0^t A dW_s = Bt + AW_t.$$

Order 2

Let us focus on the term for σ

$$\sigma\left(Y_t^{1,\varepsilon,\delta}\right) =$$

Order 2

Let us focus on the term for σ

$$\sigma\left(Y_t^{1,\varepsilon,\delta}\right) = \frac{1}{0!}\mathrm{ad}^0_{Y_t^{1,\varepsilon,\delta}}(\varepsilon A) - \frac{1}{2}\mathrm{ad}^1_{Y_t^{1,\varepsilon,\delta}}(\varepsilon A) + \mathcal{O}(\varepsilon^3,\dots,\delta^3) \qquad \leftarrow \text{Definition of } \sigma$$

Order 2

Let us focus on the term for σ

$$\sigma\left(Y_t^{1,\varepsilon,\delta}\right) = \frac{1}{0!}\mathrm{ad}^0_{Y_t^{1,\varepsilon,\delta}}(\varepsilon A) - \frac{1}{2}\mathrm{ad}^1_{Y_t^{1,\varepsilon,\delta}}(\varepsilon A) + \mathcal{O}(\varepsilon^3, \dots, \delta^3) \qquad \leftarrow \text{Definition of } \sigma$$

$$\approx \varepsilon A - \frac{1}{2}\left[Y_t^{1,\varepsilon,\delta}, \varepsilon A\right] \qquad \leftarrow \text{Definition of } \mathrm{ad}$$

Order 2

Let us focus on the term for σ

$$
\begin{aligned}
\sigma\left(Y_t^{1,\varepsilon,\delta}\right) &= \frac{1}{0!}\mathrm{ad}^0_{Y_t^{1,\varepsilon,\delta}}(\varepsilon A) - \frac{1}{2}\mathrm{ad}^1_{Y_t^{1,\varepsilon,\delta}}(\varepsilon A) + \mathcal{O}(\varepsilon^3,\ldots,\delta^3) && \leftarrow \text{Definition of } \sigma \\
&\approx \varepsilon A - \frac{1}{2}\left[Y_t^{1,\varepsilon,\delta}, \varepsilon A\right] && \leftarrow \text{Definition of } \mathrm{ad} \\
&= \varepsilon A - \frac{1}{2}\left[\delta B, \varepsilon A\right] t - \frac{1}{2}\left[\varepsilon A, \varepsilon A\right] W_t && \leftarrow \text{Definition of } Y_t^{1,\varepsilon,\delta}
\end{aligned}
$$

Order 2

Let us focus on the term for σ

$$\begin{aligned}
\sigma\left(Y_t^{1,\varepsilon,\delta}\right) &= \frac{1}{0!}\mathrm{ad}^0_{Y_t^{1,\varepsilon,\delta}}(\varepsilon A) - \frac{1}{2}\mathrm{ad}^1_{Y_t^{1,\varepsilon,\delta}}(\varepsilon A) + \mathcal{O}(\varepsilon^3,\dots,\delta^3) && \leftarrow \text{Definition of } \sigma \\
&\approx \varepsilon A - \frac{1}{2}\left[Y_t^{1,\varepsilon,\delta}, \varepsilon A\right] && \leftarrow \text{Definition of } \mathrm{ad} \\
&= \varepsilon A - \frac{1}{2}[\delta B, \varepsilon A]\, t - \frac{1}{2}[\varepsilon A, \varepsilon A]\, W_t && \leftarrow \text{Definition of } Y_t^{1,\varepsilon,\delta} \\
&= \varepsilon A - \frac{1}{2}[\delta B, \varepsilon A]\, t. && \leftarrow [A, A] = 0
\end{aligned}$$

Order 2

The term for μ can be treated the same way. We have

$$\mu\left(Y_t^{1,\varepsilon,\delta}\right) = \delta B - \frac{1}{2}\left[\varepsilon A, \delta B\right] W_t - \frac{1}{2}\varepsilon^2 A^2 + \mathcal{O}(\varepsilon^3, \dots, \delta^3).$$

In total, we get

$$Y_t^2 = \int_0^t B - \frac{1}{2}\left[A, B\right] W_s - \frac{1}{2}A^2 ds + \int_0^t A - \frac{1}{2}\left[B, A\right] s dW_s.$$

Order 2

The term for μ can be treated the same way. We have

$$\mu\left(Y_t^{1,\varepsilon,\delta}\right) = \delta B - \frac{1}{2}\left[\varepsilon A, \delta B\right] W_t - \frac{1}{2}\varepsilon^2 A^2 + \mathcal{O}(\varepsilon^3, \dots, \delta^3).$$

In total, we get

$$Y_t^2 = \int_0^t B - \frac{1}{2}\left[A, B\right] W_s - \frac{1}{2}A^2 ds + \int_0^t A - \frac{1}{2}\left[B, A\right] s dW_s.$$

Now, we want to replace the stochastic integral by a Lebesgue integral to reduce the computational effort for the implementation and avoid iterated stochastic integrals.

Order 2

We observe the following relationship by Itô's integration by parts formula

$$d\left(sW_s\right) = sdW_s + W_s ds + 0$$
$$\Leftrightarrow \int_0^t sdW_s = tW_t - \int_0^t W_s ds$$

Applying this formula yields

$$Y_t^2 = Bt - \frac{1}{2}A^2 t + \frac{1}{2}\left[B, A\right]\int_0^t W_s ds + AW_t - \frac{1}{2}\left[B, A\right]\left(tW_t - \int_0^t W_s ds\right)$$
$$= Y_t^1 - \frac{1}{2}A^2 t + \left[B, A\right]\int_0^t W_s ds - \frac{1}{2}\left[B, A\right] tW_t .$$

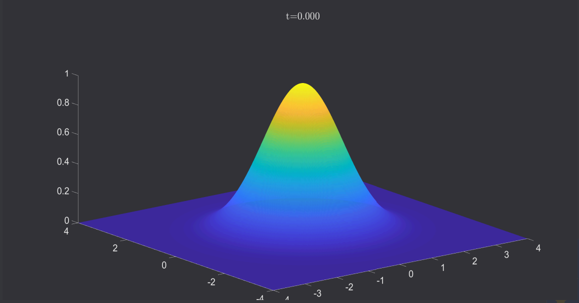
Order 3, 4

$$Y_t^3 = Y_t^2 + [[B,A],A]\left(\frac{1}{2}\int_0^t W_s^2 ds - \frac{1}{2}W_t\int_0^t W_s ds + \frac{1}{12}tW_t^2\right)$$
$$+ [[B,A],B]\left(\int_0^t sW_s ds - \frac{1}{2}t\int_0^t W_s ds - \frac{1}{12}t^2 W_t\right).$$

$$Y_t^4 = Y_t^3 + [[[B,A],A],A]\left(\frac{1}{12}W_t^2\int_0^t W_s ds - \frac{1}{4}W_t\int_0^t W_s^2 ds + \frac{1}{6}\int_0^t W_s^3 ds\right)$$
$$+ [[[B,A],B],A]\left(-\frac{1}{2}W_t\int_0^t sW_s ds + \frac{1}{2}\int_0^t sW_s^2 ds + \frac{1}{6}tW_t\int_0^t W_s ds + \frac{1}{24}t^2 W_t^2 - \frac{1}{4}t\int_0^t W_s^2 ds\right)$$
$$+ [[[B,A],B],B]\left(-\frac{1}{2}t\int_0^t sW_s ds + \frac{1}{2}\int_0^t s^2 W_s ds + \frac{1}{12}t^2\int_0^t W_s ds\right)$$
$$+ \left[[B,A],A^2\right]\left(\frac{1}{24}t^2 W_t - \frac{1}{2}\int_0^t sW_s ds + \frac{1}{4}t\int_0^t W_s ds\right)$$

SPDE

Space discretization
Vectorization
The stochastic Langevin equation



Recap

We want to discretize the following SPDE in space only to apply the Magnus expansion

$$\begin{cases} \Big(-\partial_t u_t(x,v) + h(x,v)u_t(x,v) + f^x(x,v)\partial_x u_t(x,v) + f^v(x,v)\partial_v u_t(x,v) \\ \quad + \frac{1}{2}g^{xx}(x,v)\partial_{xx}u_t(x,v) + g^{xv}(x,v)\partial_{xv}u_t(x,v) + \frac{1}{2}g^{vv}(x,v)\partial_{vv}u_t(x,v) \Big) dt \\ \quad + \left(\sigma^x(x,v)\partial_x u_t(x,v) + \sigma^v(x,v)\partial_v u_t(x,v)\right) dW_t = 0, \\ u_0(x,v) = \varphi(x,v), \end{cases}$$

Finite Differences

Let $\mathbb{X}^{n_x}_{a_x,b_x}$ be the grid for the position of a particle with $n_x + 2$ points on the subset $[a_x, b_x] \subset \mathbb{R}$ and $\mathbb{V}^{n_v}_{a_v,b_v}$ be the grid of its velocity with $n_v + 2$ points on the subset $[a_v, b_v] \subset \mathbb{R}$

$$\mathbb{X}^{n_x}_{a_x,b_x} := \{x_i^{n_x} \in [a_v, b_v] : x_i^{n_v} = a_x + i\Delta x,\ i = 0, \dots, n_v + 1\}, \qquad \Delta x := \frac{b_x - a_x}{n_x + 1},$$
$$\mathbb{V}^{n_v}_{a_v,b_v} := \left\{v_j^{n_v} \in [a_v, b_v] : v_j^{n_v} = a_v + j\Delta v,\ j = 0, \dots, n_v + 1\right\}, \qquad \Delta v := \frac{b_v - a_v}{n_v + 1},$$

For simplicity we set $d = n_x = n_v$, $[a_x, b_x] = [a_v, b_v] = [-4, 4]$ during our experiments later on.

Finite Differences

We will impose zero-boundary conditions and therefore define the central finite-difference matrices

$$D^x := \frac{1}{2\Delta x}\mathrm{tridiag}^{n_x,n_x}(-1,0,1), \qquad D^v := \frac{1}{2\Delta v}\mathrm{tridiag}^{n_v,n_v}(-1,0,1),$$
$$D^{xx} := \frac{1}{(\Delta x)^2}\mathrm{tridiag}^{n_x,n_x}(1,-2,1), \qquad D^{vv} := \frac{1}{(\Delta v)^2}\mathrm{tridiag}^{n_v,n_v}(1,-2,1).$$

$$Z^w := (z^w(x_i,v_j))_{\substack{i=1,\dots,n_x\\ j=1,\dots,n_v}}, \quad \Sigma^w := (\sigma^w(x_i,v_j))_{\substack{i=1,\dots,n_x\\ j=1,\dots,n_v}}, \quad u_t^{n_x,n_v} := (u_t(x_i,v_j))_{\substack{i=1,\dots,n_x\\ j=1,\dots,n_v}}$$

for $Z = F, G, H$, $z = f, g, h$, respectively, and $w \in \{x, v, xx, xv, vv\}$.

Finite Differences

$$f^x(x_i, v_j)\partial_x u_t(x_i, v_j) \approx f^x(x_i, v_j)\frac{u_t(x_{i+1}, v_j) - u_t(x_{i-1}, v_j)}{2\Delta x}$$

for all $i = 1, \dots, n_x$ and $j = 1, \dots, n_v$.
In our notations a derivative in x is a multiplication of the corresponding finite-difference matrix from the left to $u_t^{n_x,n_v}$, i.e.

$$\left(\frac{u_t(x_{i+1}, v_j) - u_t(x_{i-1}, v_j)}{2\Delta x}\right)_{\substack{i=1,\dots,n_x \\ j=1,\dots,n_v}} = D^x u_t^{n_x,n_v}.$$

A derivative in v on the other hand is a multiplication from the right with the transposed matrix. To get them both on the left hand side we need to vectorize the equation.

Kronecker product

Using the Kronecker product, it is well-known for compatible matrices $D_1 U D_2 = C$ that

$$\mathrm{vec}\,(C) = \mathrm{vec}\,(D_1 U D_2) = \left(D_2^T \otimes D_1\right) \mathrm{vec}\,(U)$$

Kronecker product

Using the Kronecker product, it is well-known for compatible matrices $D_1 U D_2 = C$ that

$$\mathrm{vec}\,(C) = \mathrm{vec}\,(D_1 U D_2) = \left(D_2^T \otimes D_1\right) \mathrm{vec}\,(U)$$

Now, applying this to our case yields

$$\mathrm{vec}\,\left(D^x u_t^{n_x, n_v}\right) = \mathrm{vec}\,\left(D^x u_t^{n_x, n_v} I_{n_v}\right) = \left(I_{n_v} \otimes D^x\right) U_t^{n_x n_v}.$$

Hadamard product

Using the Hadamard or element-wise product, it is well-known that

$$\mathrm{vec}\,(F \odot U) = \mathrm{vec}\,(F) \odot \mathrm{vec}\,(U) = \mathrm{diag}\,(\mathrm{vec}\,(F)) \cdot \mathrm{vec}\,(U)$$

Hadamard product

Using the Hadamard or element-wise product, it is well-known that

$$\mathrm{vec}\left(F \odot U\right) = \mathrm{vec}\left(F\right) \odot \mathrm{vec}\left(U\right) = \mathrm{diag}\left(\mathrm{vec}\left(F\right)\right) \cdot \mathrm{vec}\left(U\right)$$

Now, applying this to our case yields

$$\mathrm{vec}\left(F^x \odot \left(D^x \cdot u_t^{n_x,n_v}\right)\right) = \mathrm{diag}\left(\mathrm{vec}\left(F^x\right)\right) \cdot \mathrm{vec}\left(D^x \cdot u_t^{n_x,n_v}\right).$$

Hadamard product

Using the Hadamard or element-wise product, it is well-known that

$$\operatorname{vec}(F \odot U) = \operatorname{vec}(F) \odot \operatorname{vec}(U) = \operatorname{diag}(\operatorname{vec}(F)) \cdot \operatorname{vec}(U)$$

Now, applying this to our case yields

$$\operatorname{vec}\left(F^x \odot \left(D^x \cdot u_t^{n_x, n_v}\right)\right) = \operatorname{diag}\left(\operatorname{vec}\left(F^x\right)\right) \cdot \operatorname{vec}\left(D^x \cdot u_t^{n_x, n_v}\right).$$

In total, we have

$$[f^x(x_i, v_j)\partial_x u_t(x_i, v_j)]_{\substack{i=1,\dots,n_x \\ j=1,\dots,n_v}} = \operatorname{diag}\left(\operatorname{vec}\left(F^x\right)\right) \cdot \left(I_{n_v} \otimes D^x\right) \cdot U_t^{n_x n_v}.$$

Vectorization

Applying this logic to all other summands in the (SPDE) yields

$$
\begin{aligned}
B := \operatorname{diag}\left(\operatorname{vec}\left(H\right)\right) \\
&+ \operatorname{diag}\left(\operatorname{vec}\left(F^{x}\right)\right) \cdot \left(I_{n_v} \otimes D^{x}\right) \\
&+ \operatorname{diag}\left(\operatorname{vec}\left(F^{v}\right)\right) \cdot \left(D^{v} \otimes I_{n_x}\right) \\
&+ \frac{1}{2}\operatorname{diag}\left(\operatorname{vec}\left(G^{xx}\right)\right) \cdot \left(I_{n_v} \otimes D^{xx}\right) \\
&+ \operatorname{diag}\left(\operatorname{vec}\left(G^{xv}\right)\right) \cdot \left(D^{v} \otimes D^{x}\right) \\
&+ \frac{1}{2}\operatorname{diag}\left(\operatorname{vec}\left(G^{vv}\right)\right) \cdot \left(D^{vv} \otimes I_{n_x}\right) \\
A := \operatorname{diag}\left(\operatorname{vec}\left(\Sigma^{x}\right)\right) \cdot \left(I_{n_v} \otimes D^{x}\right) \\
&+ \operatorname{diag}\left(\operatorname{vec}\left(\Sigma^{v}\right)\right) \cdot \left(D^{v} \otimes I_{n_x}\right).
\end{aligned}
$$

Stochastic Langevin equation

$$h \equiv f^v \equiv g^{xx} \equiv g^{xv} \equiv \sigma^x \equiv 0, \quad f_x(x,v) := -v, \quad g^{vv} \equiv a, \quad \sigma^v \equiv \sigma. \tag{2.4}$$

In this special case, there exists an explicit fundamental solution Γ for $0 < \sigma \leq \sqrt{a}$ (cf. PASCUCCI and PESCE (2022):p. 4 Proposition 1.1.), which is given by

$$\Gamma\left(t, z; 0, \zeta\right) := \Gamma_0\left(t, z - m_t(\zeta)\right),$$
$$\Gamma_0\left(t, [x, v]\right) := \frac{\sqrt{3}}{\pi t^2(a-\sigma^2)} \exp\left(-\frac{2}{a-\sigma^2}\left(\frac{v^2}{t} - \frac{3vx}{t^2} + \frac{3x^2}{t^3}\right)\right)$$

where $\zeta := (\xi, \eta)$ is the initial point and

$$m_t(\zeta) := \begin{pmatrix} \xi + t\eta - \sigma \int_0^t W_s ds \\ \eta - \sigma W_t \end{pmatrix}.$$

Stochastic Langevin equation

Having the fundamental solution, we can solve the Cauchy-problem by integrating against the initial datum, i.e.

$$u_t(x, v) = \int_{\mathbb{R}^2} \Gamma(t, [x, v]; 0, [\xi, \eta])\varphi(\xi, \eta) d\xi d\eta.$$

To get an explicit solution for the double integral, we will choose φ to be Gaussian, i.e.

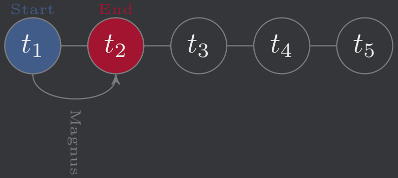
$$\varphi(\xi, \eta) := \exp\left(-\frac{(\xi^2 + \eta^2)}{2}\right).$$

Numerics

Sparsity
Errors and Computational Times

Implementation

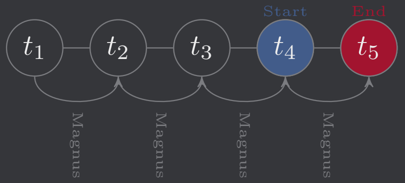
1. We apply the Magnus expansion iteratively on small subintervals to avoid blow-ups.
2. To compute the matrix exponential, we use a special matrix-vector exponential expmvtay2[a]: Without full matrix exponential, uses sparsity, works on GPU



[a] see Ibáñez et al. (2022)

Implementation

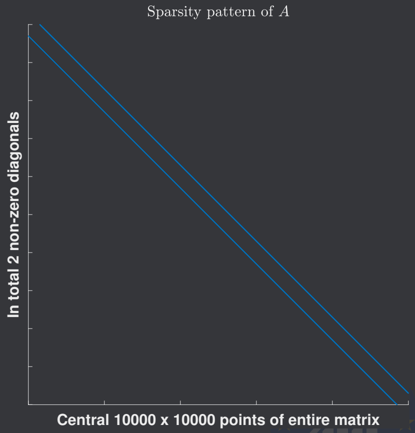
1. We apply the Magnus expansion iteratively on small subintervals to avoid blow-ups.
2. To compute the matrix exponential, we use a special matrix-vector exponential expmvtay2[a]: Without full matrix exponential, uses sparsity, works on GPU



[a] see Ibáñez et al. (2022)

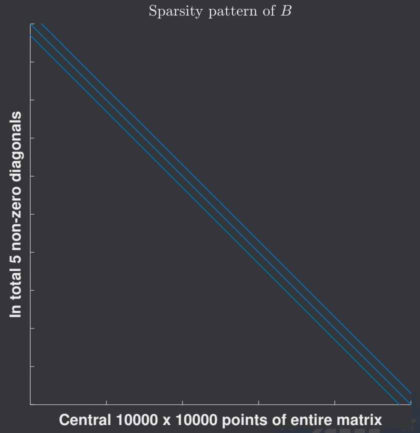
Sparsity is our friend

1. Number of non-zero diagonals of A: 2
2. Number of non-zero diagonals of B: 5
3. Number of non-zero diagonals of $[B, A]$: 5
4. Number of non-zero diagonals of $[[B, A], A]$: 8
5. Number of non-zero diagonals of $[[B, A], B]$: 10
6. This will allow us to use NVIDIA. cuSPARSE library efficiently for both the Magnus logarithm as well as the matrix-vector exponential.



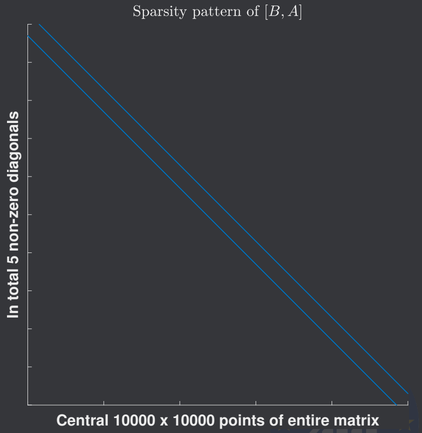
Sparsity is our friend

1. Number of non-zero diagonals of A: 2
2. Number of non-zero diagonals of B: 5
3. Number of non-zero diagonals of $[B, A]$: 5
4. Number of non-zero diagonals of $[[B, A], A]$: 8
5. Number of non-zero diagonals of $[[B, A], B]$: 10
6. This will allow us to use NVIDIA cuSPARSE library efficiently for both the Magnus logarithm as well as the matrix-vector exponential.



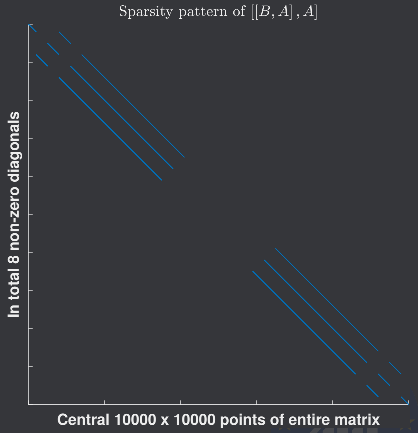
Sparsity is our friend

1. Number of non-zero diagonals of A: 2
2. Number of non-zero diagonals of B: 5
3. Number of non-zero diagonals of $[B, A]$: 5
4. Number of non-zero diagonals of $[[B, A], A]$: 8
5. Number of non-zero diagonals of $[[B, A], B]$: 10
6. This will allow us to use NVIDIA cuSPARSE library efficiently for both the Magnus logarithm as well as the matrix-vector exponential.



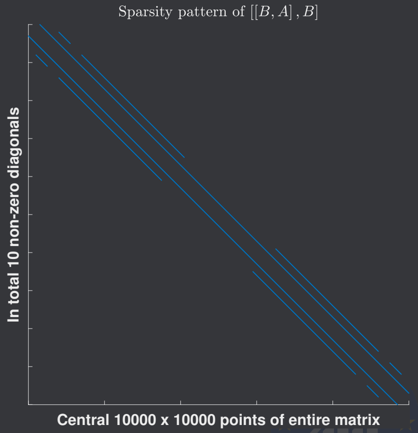
Sparsity is our friend

1. Number of non-zero diagonals of A: 2
2. Number of non-zero diagonals of B: 5
3. Number of non-zero diagonals of $[B, A]$: 5
4. Number of non-zero diagonals of $[[B, A], A]$: 8
5. Number of non-zero diagonals of $[[B, A], B]$: 10
6. This will allow us to use NVIDIA. cuSPARSE library efficiently for both the Magnus logarithm as well as the matrix-vector exponential.



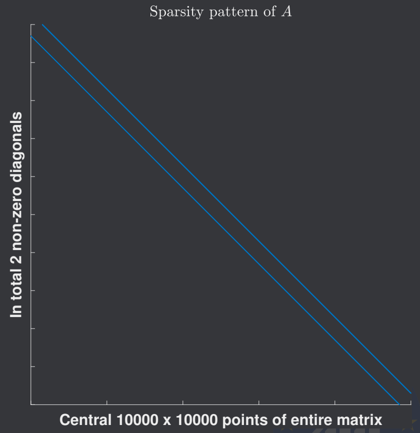
Sparsity is our friend

1. Number of non-zero diagonals of A: 2
2. Number of non-zero diagonals of B: 5
3. Number of non-zero diagonals of $[B, A]$: 5
4. Number of non-zero diagonals of $[[B, A], A]$: 8
5. Number of non-zero diagonals of $[[B, A], B]$: 10
6. This will allow us to use NVIDIA. cuSPARSE library efficiently for both the Magnus logarithm as well as the matrix-vector exponential.



Sparsity is our friend

1. Number of non-zero diagonals of A: 2
2. Number of non-zero diagonals of B: 5
3. Number of non-zero diagonals of $[B, A]$: 5
4. Number of non-zero diagonals of $[[B, A], A]$: 8
5. Number of non-zero diagonals of $[[B, A], B]$: 10
6. This will allow us to use NVIDIA cuSPARSE library efficiently for both the Magnus logarithm as well as the matrix-vector exponential.

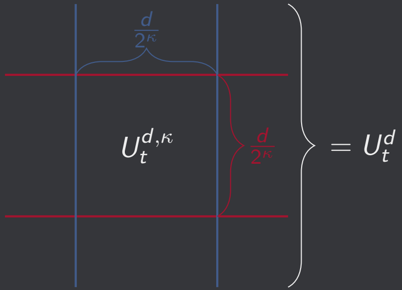


Error definition

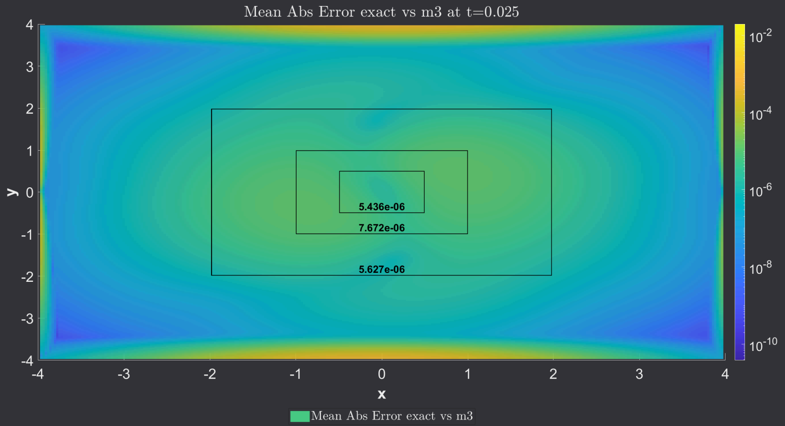
We consider the mean error in all grid points

$$\mathrm{AE}_t^{d,\kappa} := \frac{1}{M}\sum_{m=1}^{M}\left|U_{t,m}^{\mathrm{ref},d,\kappa} - U_{t,m}^{\mathrm{approx},d,\kappa}\right| \in \mathbb{R}^{\frac{d}{2^\kappa},\frac{d}{2^\kappa}}.$$

For our comparison later on we will take the average over the region $\kappa = 4$, i.e. $[-0.25, 0.25] \times [-0.25, 0.25]$ to avoid issues due to the zero boundary condition.

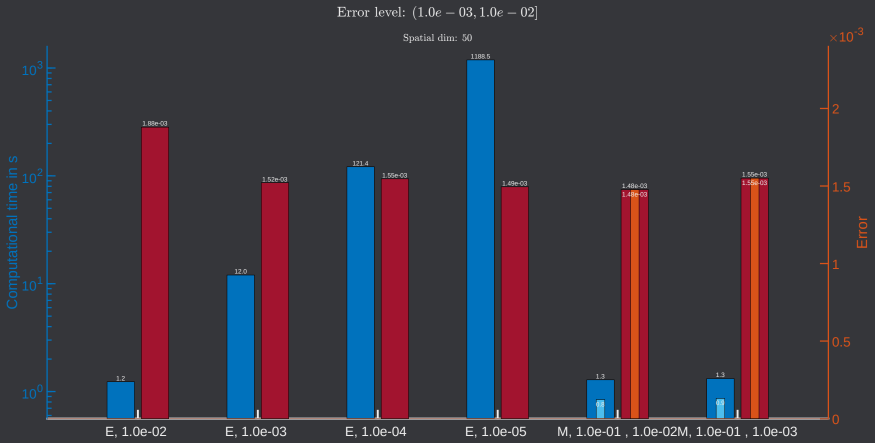


Absolute Errors

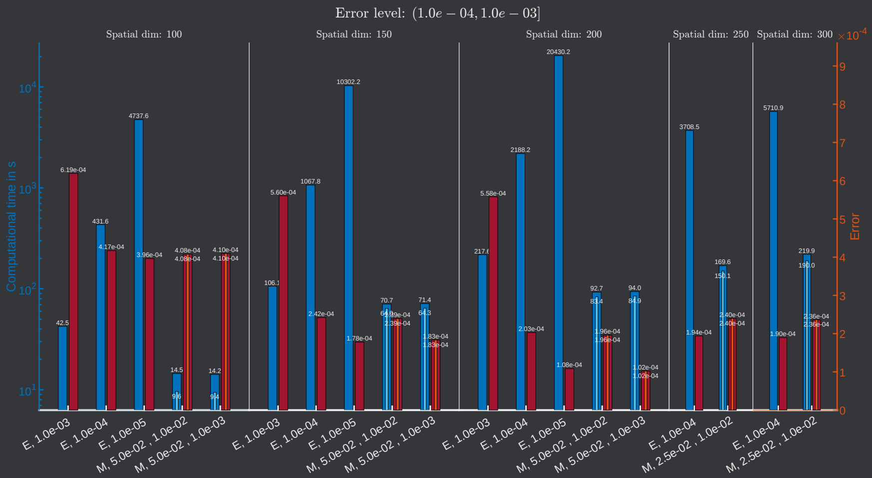


In the case $d = 300$ and $\Delta = 2.5e-2$ on $[-4,4] \times [-4,4]$

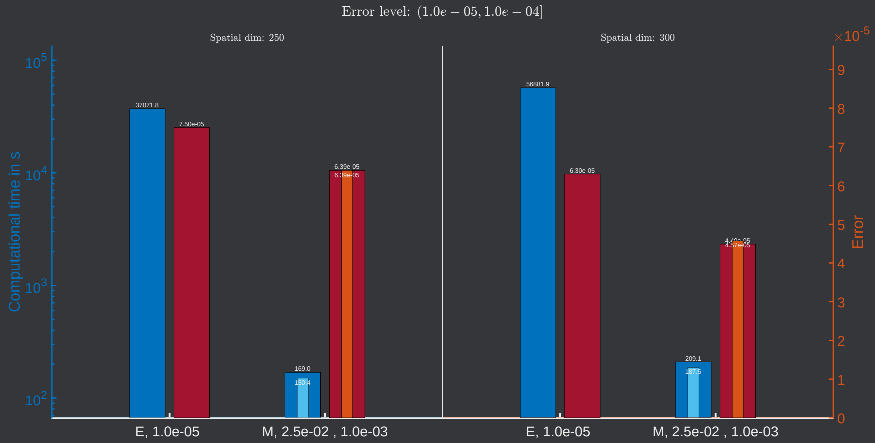
Computational times vs Error level



Computational times vs Error level



Computational times vs Error level



Natural step-size control

The iterated Magnus scheme has the possibility for a natural time step-size control.

$$|\text{order } 2 - \text{order } 3| \leq \text{tol} \quad \Rightarrow \quad \text{Step-size small enough}$$

Computational effort:

1. Magnus logarithm of order 2 can be re-used for order 3
2. two matrix-vector exponentials per step-size reduction for each trajectory
   - Maybe select random smaller batch-size to determine step-size

Conclusion

We have seen how to derive the Itô-stochastic Magnus expansion for SDEs with constant matrices and used it to solve two-dimensional SPDEs with a given initial datum numerically. The scheme has an excellent accuracy and its advantage in terms of computational effort excels for higher spatial resolution, e.g. to have the same accuracy for $d = 400$ we need an Euler scheme with $\Delta = 1e - 5$ taking 35.8 hours, while Magnus order 3 takes only 462 seconds using $\Delta = 1e - 2$ with $M = 100$ trajectories.
This is a speed-up by a factor 280 just using one GPU while sparsity ensures an almost equal memory demand.

Thank you for your attention!

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 813261 and is part of the ABC-EU-XVA project.