AGH University of Science and Technology

Noise, Precision, and Stochastic Differential Equations: From Numerical Analysis to Modern AI Systems

Paweł M. Morkisz

Aigorithmics & AGH UST, Krakow, Poland

Based in part on joint work with Paweł Przybyłowicz and Martyna Wiącek

COST Stochastica Workshop, University College Cork

April 27th, 2026

The main tension

Numerical analysis usually assumes exact information.

Classical view

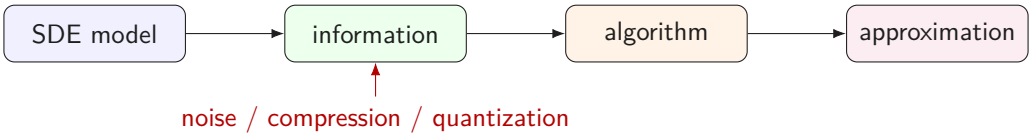
The algorithm evaluates

$$a(t,x), \qquad b(t,x), \qquad W(t)$$

exactly, and the main design parameter is the time step.

Modern computational reality

The algorithm may only see compressed, quantized, sparse, or otherwise noisy versions of the objects it needs.



Talk map: 30 minutes

Roadmap

1. motivation: exact $\rightarrow$ noisy information,
2. representative upper bounds for Euler and Milstein,
3. lower bounds and the induced error floor,
4. AI systems: precision, sparsity, compression,
5. diffusion models as an SDE-based modern example.

Goal: To show that noisy information is not only a technical nuisance in SDE numerics: it is a natural language for discussing the limits of modern low-information computation.

Starting point: strong approximation of an SDE

Let

$$\begin{cases} \mathrm{d}X(t) = a(t, X(t))\,\mathrm{d}t + b(t, X(t))\,\mathrm{d}W(t), \quad t \in [0, T], \\ X(0) = \eta. \end{cases}$$

Classical numerical question

Given access to the coefficients and the driving Wiener process, approximate $X(T)$ in a strong sense:

$$\left\| X(T) - \widehat{X}(T) \right\|_r = \left(\mathbb{E} |X(T) - \widehat{X}(T)|^r \right)^{1/r}.$$

Question for today

What changes if the algorithm does *not* receive exact information about a, b, and W?

Classical schemes: Euler and Milstein

For a uniform grid $t_i = ih$, $h = T/n$:

Euler type step

$$\bar{X}_{i+1} = \bar{X}_i + a(t_i, \bar{X}_i)h + b(t_i, \bar{X}_i)\Delta W_i.$$

Milstein type step

$$\bar{X}_{i+1} = \bar{X}_i + a(t_i, \bar{X}_i)h + b(t_i, \bar{X}_i)\Delta W_i + L_1 b(t_i, \bar{X}_i) I_{i,i+1}(W, W),$$

where

$$L_1 b = b\,\partial_y b, \qquad I_{i,i+1}(W, W) = \frac{1}{2}\left((\Delta W_i)^2 - h\right).$$

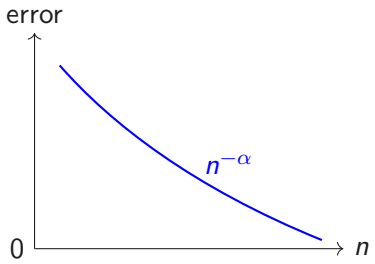
Classical message

Discretization is the main error source

Typical strong rates are of the form

$$\left\| X(T) - \bar{X}_n(T) \right\|_r \lesssim n^{-\alpha},$$

where α depends on the scheme and regularity assumptions.



Hidden assumption

The right-hand side reflects only discretization. It assumes the information supplied to the algorithm is exact.

Noisy information model

Instead of exact objects, the algorithm observes perturbed ones:

$$\widetilde{a}(t,y) = a(t,y) + \delta_1 p_a(t,y), \qquad \widetilde{b}(t,y) = b(t,y) + \delta_2 p_b(t,y),$$

$$\widetilde{W}(t) = W(t) + \delta_3 p_W(t, W(t)).$$

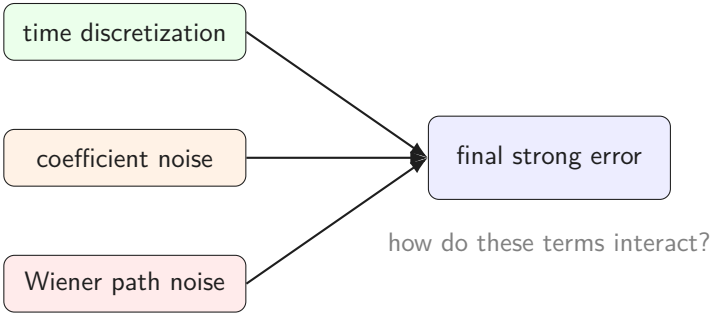
Interpretation

- δ_1: drift information noise,
- δ_2: diffusion information noise,
- δ_3: driving noise / path observation noise.

Worst-case viewpoint

Perturbations are not necessarily random in a friendly way. They may be adversarial within a prescribed class.

Three errors, one algorithm



$$\text{total error} \approx \text{discretization error} + \text{information error}$$

But the exact form depends on the numerical scheme and on the regularity of the perturbations.

Randomized Euler under noisy information

A typical noisy randomized Euler step is

$$\bar{X}_{i+1}^{RE} = \bar{X}_i^{RE} + \widetilde{a}(\xi_i, \bar{X}_i^{RE})h + \widetilde{b}(t_i, \bar{X}_i^{RE})\Delta\widetilde{W}_i,$$

where $\xi_i \sim \mathrm{Unif}[t_i, t_{i+1}]$.

Representative upper bound

For standard coefficient classes and smooth path perturbations,

$$\left\| X(T) - \bar{X}_n^{RE}(T) \right\|_r \leq C \left(n^{-\min\{\varrho, 1/2\}} + \delta_1 + \delta_2 + \delta_3 \right).$$

Message

Even if $n \to \infty$, fixed information noise creates a non-vanishing error floor.

Milstein and noisy information: what becomes harder?

Milstein improves the discretization rate, but it also uses more structure:

$$L_1 b(t,y) = b(t,y)\partial_y b(t,y), \qquad I_{i,i+1}(W,W) = \frac{1}{2}((\Delta W_i)^2 - h).$$

Two extra difficulties

1. If $\widetilde{b} = b + \delta_2 p_b$ and p_b is merely measurable, then $\partial_y \widetilde{b}$ may not exist.
2. The iterated integral becomes nonlinear in the noisy path:

$$I(\widetilde{W},\widetilde{W}) = I(W,W) + \delta_3 \Delta W \Delta Z + \frac{1}{2}\delta_3^2 (\Delta Z)^2.$$

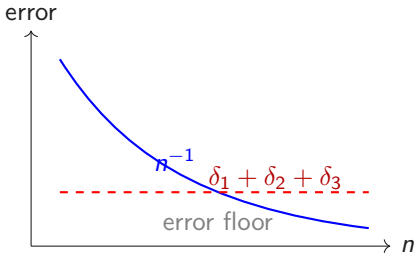
Practical conclusion

Higher order methods can be more accurate, but they need more reliable information.

Milstein: representative upper bound

stronger regularity assumptions and smooth Wiener perturbations

$$\left\| X(T) - \bar{X}_n^{RM}(T) \right\|_2 \leq C \left(n^{-1} + \delta_1 + \delta_2 + \delta_3 \right).$$



A better time discretization rate does not remove the information barrier.

Lower bounds: noise creates a fundamental limit

Let $e_n^{(r)}$ denote the minimal worst-case error over algorithms using at most n information evaluations.

Representative optimality statement

For standard problem classes and admissible perturbation models,

$$e_n^{(r)} \asymp n^{-\alpha} + \delta_1 + \delta_2 + \delta_3,$$

up to constants, with α determined by the regularity assumptions and the algorithmic class.

No algorithm can beat the noise level.

This turns noisy information from an implementation detail into a complexity-theoretic obstruction.

Why the trend toward lower precision?

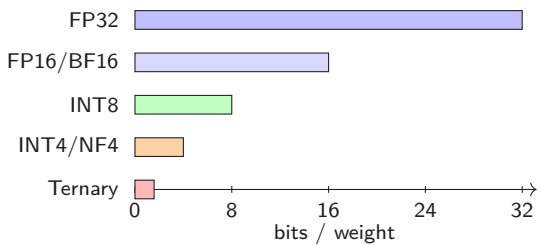
Main engineering reasons

- smaller memory footprint,
- less bandwidth and memory traffic,
- higher throughput and lower energy cost.

Representative energy figures

32-bit float multiply	≈ 3.7 pJ
8-bit integer multiply	≈ 0.2 pJ
32-bit DRAM access	≈ 640 pJ

Moving fewer bits is often the easiest way to save both time and energy.

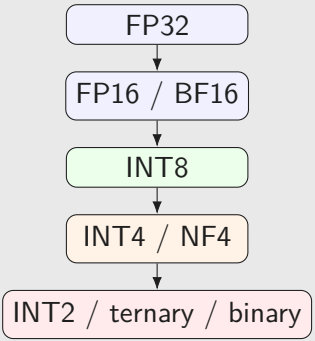


The trend is not aesthetic: lower precision is a direct route to more computation per watt.

Energy figures adapted from Horowitz (ISSCC 2014).

Modern AI: exact arithmetic is not the default

Dense low precision



Numerical-analysis interpretation

Low precision changes the information available to the algorithm.

- 4-bit quantized models are now standard in memory-efficient fine-tuning.
- Ternary-weight LLMs use values in $\{-1, 0, +1\}$, i.e. $\log_2 3 \approx 1.58$ bits per dense weight.
- Not only storage: arithmetic itself becomes structured and lossy.

Examples: QLoRA (Dettmers et al., 2023), BitNet b1.58 (Ma et al., 2024).

Beyond quantization: sparsity

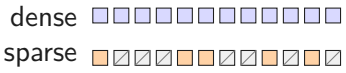
- **Dense quantization:** keep every coordinate, but store it with fewer bits,

$$x_i \mapsto Q_b(x_i).$$

- **Sparse quantization:** remove many coordinates and quantize the survivors,

$$x \mapsto M \odot Q_b(x), \qquad M_i \in \{0, 1\}.$$

- **Information viewpoint:** count bits per coordinate of the original dense object, not only bits per stored value.



Effective sub-bit representations

A simple sparse ternary example:

$$x_i \in \{-1, 0, +1\}, \qquad \mathbb{P}(x_i = 0) = 0.9, \quad \mathbb{P}(x_i = 1) = \mathbb{P}(x_i = -1) = 0.05.$$

The entropy per original coordinate is

$$H(0.9, 0.05, 0.05) = -0.9 \log_2(0.9) - 2 \cdot 0.05 \log_2(0.05) \approx 0.57 \text{ bits}.$$

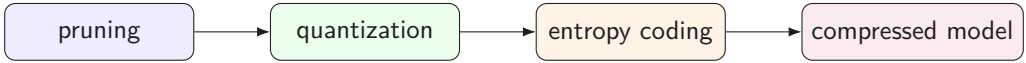
Message

With sparsity, the effective information content per dense coordinate can be below one bit.

Caveat

Actual storage depends on mask/index overhead. Structured sparsity or entropy coding can reduce this overhead substantially.

Compression pipelines already combine these ideas



Examples

- Deep Compression: pruning + trained quantization + Huffman coding; reported $35\times$-$49\times$ storage reduction.
- SparseGPT: one-shot pruning of large GPT-family models to at least 50% sparsity, with compatibility with quantization.

Numerical-analysis translation

The algorithm operates on compressed information, not on the original dense object.

Examples: Deep Compression (Han, Mao, Dally, 2015); SparseGPT (Frantar, Alistarh, 2023).

From sparse arithmetic to noisy information

Classical noisy information often looks additive:

$$\tilde{a} = a + \delta_1 p_a, \qquad \tilde{b} = b + \delta_2 p_b.$$

Modern compressed computation may look more like

$$\tilde{a} = \mathcal{C}(a), \qquad \tilde{b} = \mathcal{C}(b), \qquad \mathcal{C} = \text{sparsification} \circ \text{quantization}.$$

Additive bounded noise

- clean to analyze,
- natural in IBC,
- useful first-order model.

Compressed computation

- nonlinear,
- biased,
- structured,
- often state-dependent.

Diffusion models: from pure noise to a sample

Forward process

Start from clean data and gradually inject noise:

$$x_0 \to x_t \to x_T \approx \mathcal{N}(0, I).$$

After many steps the sample is almost Gaussian.

Reverse process

Generation runs in the opposite direction:

$$x_T \sim \mathcal{N}(0, I) \to x_{t_k} \to \cdots \to x_0.$$

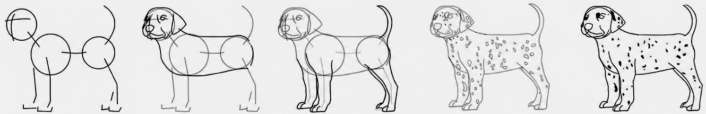
The model repeatedly predicts how to denoise.

pure noise → less noise → structure → generated image

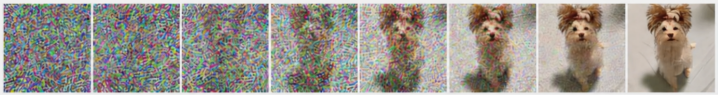
In continuous time, this iterative denoising can be viewed through a reverse-time SDE (or a related ODE).

Meme intuition: how humans draw a dog vs how AI draws a dog

How humans draw a dog



How AI draws a dog



Diffusion models: SDEs are already in the loop

Score-based generative modeling can be formulated through a forward SDE that adds noise and a reverse-time SDE that generates samples.

Schematic reverse-time dynamics

$$\mathrm{d}X_t = \Big[f(t, X_t) - g(t)^2 \, s_\theta(t, X_t) \Big] \, \mathrm{d}t + g(t) \, \mathrm{d}\bar{W}_t,$$

where s_θ is a learned score approximation.

Where numerical error and information error meet

- time discretization of the reverse dynamics,
- approximation error in the learned score,
- low-precision or sparse evaluation of the network,
- stochastic sampling noise.

AI bridge: compressed score = compressed drift information

In diffusion models, the learned score modifies the drift. If the network is quantized or sparse, then the solver effectively sees

$$s_\theta(t,x) \quad \rightsquigarrow \quad \mathcal{C}(s_\theta)(t,x),$$

and hence

$$f(t,x) - g(t)^2 s_\theta(t,x) \quad \rightsquigarrow \quad f(t,x) - g(t)^2 \mathcal{C}(s_\theta)(t,x).$$

Interpretation

Low precision and sparsity can be viewed as perturbations of the drift information available to the SDE solver.

Open direction

Develop numerical analysis for SDE solvers where coefficient information is quantized, sparse, and learned.

What should we take away?

1. **Noise changes numerical analysis.**
   The error is no longer only a discretization error.

2. **There are provable limits.**
   For standard noisy information models, upper and lower bounds match up to constants.

3. **Modern AI makes this timely.**
   Low precision, sparsity, and compression change the information available to algorithms.

From low precision to low information.

Open questions

Numerical analysis

- How should we model quantized and sparse coefficient information?
- Which perturbation models give sharp upper and lower bounds?
- Can higher-order schemes remain stable under structured computational noise?

AI systems

- How does low precision affect SDE samplers in diffusion models?
- When does compression act like harmless regularization, and when does it create an error floor?
- Can IBC-style lower bounds inform architecture and precision choices?

Thank you!

Questions?

References for SDE numerics and noisy information

- M. Baranek, A. Kałuża, P. M. Morkisz, P. Przybyłowicz, M. Sobieraj,
  On the randomized Euler algorithm under inexact information,
  Journal of Computational and Applied Mathematics, 476, 117070, 2026.
- P. M. Morkisz, P. Przybyłowicz, M. Wiącek,
  Optimality and error of randomized Milstein algorithms for SDEs under noisy information,
  manuscript in preparation.

References for the AI context

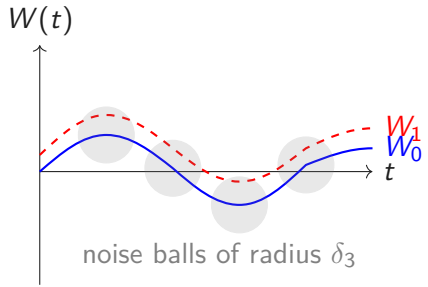
- Y. Song et al., *Score-Based Generative Modeling through Stochastic Differential Equations*, 2020.
- J. Ho, A. Jain, P. Abbeel, *Denoising Diffusion Probabilistic Models*, 2020.
- M. Horowitz, *Computing's Energy Problem (and What We Can Do About It)*, ISSCC 2014.
- T. Dettmers et al., *QLoRA: Efficient Finetuning of Quantized LLMs*, 2023.
- S. Ma et al., *The Era of 1-bit LLMs: All Large Language Models are in 1.58 Bits*, 2024.
- E. Frantar, D. Alistarh, *SparseGPT*, 2023.
- S. Han, H. Mao, W. J. Dally, *Deep Compression*, 2015.

Backup: Lower bound proof idea – indistinguishable worlds

Two-point lower bound

Construct two inputs that lead to different solutions, but whose noisy observations are hard to distinguish.

- Le Cam-type argument,
- total variation / KL control,
- if observations overlap, any algorithm must fail on one input.



Three regimes

Regime	Dominant term	Interpretation
Classical	$n^{-\alpha} \gg \delta$	refine the grid
Balanced	$n^{-\alpha} \approx \delta$	optimal computational budget
Noise dominated	$n^{-\alpha} \ll \delta$	more time steps do not help

Design implication

The step size is no longer the only design parameter. One should balance discretization cost against information quality.

One-line takeaway

Increasing n past the noise floor buys accuracy only on paper.

Backup: noisy Milstein expansion

Let $\widetilde{W} = W + \delta_3 Z$, where $Z(t) = p_W(t, W(t))$. Then

$$I_{i,i+1}(\widetilde{W}, \widetilde{W}) = \frac{1}{2}\left((\Delta W_i + \delta_3 \Delta Z_i)^2 - h\right).$$

Therefore

$$I_{i,i+1}(\widetilde{W}, \widetilde{W}) = I_{i,i+1}(W, W) + \delta_3 \Delta W_i \Delta Z_i + \frac{1}{2}\delta_3^2 (\Delta Z_i)^2.$$

Why this is useful as a backup slide

If someone asks about the proof, this is the key place where noisy path information produces genuinely new terms.

Backup: lower bound toolbox

Le Cam two-point principle

Choose two inputs u_0, u_1 such that the corresponding target values differ by Δ, but the distributions of noisy observations are close.

Pinsker inequality

$$\|\mathbb{P}_0 - \mathbb{P}_1\|_{TV} \leq \sqrt{\frac{1}{2} D_{KL}(\mathbb{P}_0 \| \mathbb{P}_1)}.$$

Consequence

If the observations cannot distinguish the two inputs, any algorithm has worst-case error of order Δ, which is chosen proportional to the relevant noise level.

Backup: a possible refined model for compressed coefficients

Instead of additive perturbations, consider an operator model

$$\mathcal{C}_{\delta,s}(f)(t,x) = M_s(t,x) \odot Q_\delta(f(t,x)),$$

where Q_δ is a quantizer and M_s is a sparsity mask.

Questions

- What regularity of $\mathcal{C}_{\delta,s}(f)$ is preserved?
- Is the perturbation biased or unbiased?
- Does the mask depend on (t,x), on f, or on the hardware kernel?
- What is the correct information budget: evaluations, bits, entropy, or energy?