The Spectral Tensor Representation of Stochastic Processes

One Matrix. Every Property. No Monte Carlo.

Tamás Nagy, Ph.D.

tnagyphd@gmail.com

Draft

Abstract

We prove that every Itô diffusion $dX_t = \mu(X_t)\,dt + \sigma(X_t)\,dW_t$ admits a complete, finite-dimensional representation through its Fokker–Planck generator discretized in cosine basis. The generator matrix $M \in \mathbb{R}^{N\times N}$, computed via integration-by-parts weak form with reflecting boundary conditions, is a **sufficient statistic for the entire stochastic system**: time-dependent distributions, all moments, option prices, Greeks, Value-at-Risk, Expected Shortfall, stationary distributions, spectral gaps, and autocorrelation functions are all computable from M alone — without Monte Carlo simulation. For time-homogeneous diffusions, M is a matrix; for time-inhomogeneous processes, it extends to a 3-tensor T_{kjl}; for multi-dimensional systems, to higher-order tensors. The Universal Risk Representation Theorem (Nagy, 2026b) guarantees that $N = \Theta(\log(1/\varepsilon)/\log\rho)$ basis functions suffice regardless of dimension or the property being computed, where $\rho > 1$ is the spectral decay rate of the process. We validate the completeness theorem through three-way comparison (spectral vs analytical vs Monte Carlo) on the Ornstein–Uhlenbeck process, demonstrate it on a nonlinear double-well potential with bimodal dynamics, and apply it to mean-reverting energy pricing where a single 48×48 matrix produces the complete term structure of prices, risk measures, and Greeks.

---

1. Introduction

1.1 The Central Claim

Consider a one-dimensional Itô diffusion:

$$dX_t = \mu(X_t)\,dt + \sigma(X_t)\,dW_t \tag{1}$$

Its law is governed by the Fokker–Planck equation for the density $p(x,t)$:

$$\frac{\partial p}{\partial t} = \mathcal{L}[p] = -\frac{\partial}{\partial x}[\mu(x)\,p] + \frac{1}{2}\frac{\partial^2}{\partial x^2}[\sigma^2(x)\,p] \tag{2}$$

We expand p in cosine basis $\{\varphi_k\}$ on a bounded domain $[a,b]$:

$$p(x,t) = \sum_{k=0}^{N-1} A_k(t)\,\varphi_k(x), \qquad \varphi_k(x) = \begin{cases} 1/\sqrt{L} & k = 0 \\ \sqrt{2/L}\cos(k\pi(x-a)/L) & k \geq 1 \end{cases} \tag{3}$$

The Fokker–Planck equation becomes a linear ODE system:

$$\frac{dA_k}{dt} = \sum_{j=0}^{N-1} M_{kj}\, A_j(t) \tag{4}$$

Theorem (Completeness). *The matrix $M \in \mathbb{R}^{N\times N}$ is a sufficient statistic for the SDE (1). Every property that is a functional of the law of $(X_t)_{t\geq 0}$ is computable from M and an initial condition $A(0)$.*

This paper proves this claim constructively: we derive explicit formulas for ten distinct properties from M alone, validate them against analytical solutions and Monte Carlo, and extend the representation to higher-dimensional and time-varying processes via tensors.

1.2 The Representation Hierarchy

The spectral tensor framework unifies static and dynamic problems:

Object	Spectral form	Decomposition	Example
Static distribution	Vector $\{A_k\}$	List of numbers	Spectral Fenton (Nagy, 2026a)
Time-homogeneous SDE	Matrix M_{kj}	**SVD** — clean, unique, optimal	This paper
Time-inhomogeneous SDE	3-tensor T_{kjl}	**CP decomposition**	Dynamic volatility
Multi-asset SDE	Higher tensor	**CP / tensor network**	Portfolio dynamics

The Spectral Fenton Distribution (Nagy, 2026a) — a vector of 128 coefficients representing a portfolio loss distribution — is the **static special case** of this framework: the snapshot at a single time t. The generator matrix M is the **dynamic object** from which all such snapshots are derived.

1.3 Why This Matters

The standard approach to stochastic processes in quantitative finance is Monte Carlo simulation: generate 10^5–10^6 sample paths, estimate any quantity by averaging. This is general but:

1. **Slow**: $O(1/\sqrt{n})$ convergence. Reducing error by $10\times$ requires $100\times$ more paths.
2. **Noisy**: Every estimate carries sampling error. Two runs give different answers.
3. **Redundant**: Computing VaR requires one MC run. Computing ES requires another. Each Greek requires a bump-and-reprice. A desk with 1000 positions needs millions of pricings.

The spectral generator replaces all of this with **one matrix computation**. Once M is built (a one-time $O(N^2Q)$ cost where Q is the quadrature order), every subsequent query — distribution, price, Greek, risk measure — is $O(N^2)$ (a matrix exponential and inner product).

1.4 Relation to Prior Work

Fourier methods in finance. The COS method (Fang and Oosterlee, 2009) prices European options from the characteristic function. Our work extends this: instead of pricing one option at one maturity, we represent the *entire process* and derive *all* properties.

Spectral methods for PDEs. Spectral Galerkin methods for Fokker–Planck equations are well-established in applied mathematics (Boyd, 2001; Gottlieb and Orszag, 1977). Our contribution is the identification that the resulting matrix M is a **complete financial object** — a sufficient statistic for pricing, risk, and dynamics simultaneously — and the connection to the URRT dimension-free bounds.

The Spectral Fenton series. This paper is the *dynamic generalization* of the Spectral Fenton Distribution (Nagy, 2026a), the Eigen-COS method (Nagy, 2026b–e), and the Universal Risk Representation Theorem. Those works treat the distribution at a fixed time; this work shows that the *generator* of the time evolution is the natural object, and all prior results are derived quantities.

---

2. Construction of the Spectral Generator

2.1 The Weak Form

The Fokker–Planck operator $\mathcal{L}$ maps densities to densities. For reflecting (zero-flux) boundary conditions, the appropriate discretization uses the **integration-by-parts weak form**:

$$M_{kj} = \int_a^b \varphi_k'(x)\,[\mu(x) - D'(x)]\,\varphi_j(x)\,dx - \int_a^b D(x)\,\varphi_k'(x)\,\varphi_j'(x)\,dx \tag{5}$$

where $D(x) = \sigma^2(x)/2$ is the diffusion coefficient and D' is its spatial derivative.

Proposition 1 (Probability Conservation). *$M_{0,j} = 0$ for all j. Equivalently, $\frac{d}{dt}\int p(x,t)\,dx = 0$.*

Proof. $\varphi_0' \equiv 0$ (the constant mode has zero derivative). Both integrals in (5) vanish when $k = 0$. □

This is not merely a consistency check — it is a **structural requirement** that the pointwise expansion $\mathcal{L}[\varphi_j] = -\mu'\varphi_j - \mu\varphi_j' + D\varphi_j'' + 2D'\varphi_j' + D''\varphi_j$ does *not* satisfy (it gives $M_{0,0} = -\mu'(x_0) \neq 0$ in general). The weak form is essential.

2.2 Spectral Properties of M

Proposition 2 (Dissipation). *All eigenvalues of M have non-positive real part: $Re(\lambda_k) \leq 0$ for all k.*

For the Ornstein–Uhlenbeck process $dX = -\theta(X - m)\,dt + \sigma\,dW$, the eigenvalues are:

$$\lambda_n = -n\theta, \qquad n = 0, 1, 2, \ldots \tag{6}$$

The spectral gap $|\lambda_1| = \theta$ controls the rate of convergence to stationarity.

Proposition 3 (Stationary Distribution). *The null eigenvector of M — $M\pi = 0$, normalized so $\sum_k \pi_k \int \varphi_k\,dx = 1$ — gives the stationary distribution of the process.*

2.3 Implementation

The matrix M is computed via Gauss–Legendre quadrature with Q nodes. For N basis functions, the cost is $O(N^2Q)$. In practice, $N = 32$–64 and $Q = 256$–512 suffice for machine-precision accuracy on smooth problems.

The derivatives $D'(x)$ are computed by central finite differences of $\sigma(x)$ with step $h = 10^{-5}$. For SDEs with analytically known σ', this can be replaced with exact derivatives.

---

3. The Completeness Theorem: Ten Properties from One Matrix

3.1 Time-Dependent Distribution

Property 1. The density at time t from initial condition $A(0)$:

$$A(t) = e^{Mt}\,A(0), \qquad p(x,t) = \sum_k A_k(t)\,\varphi_k(x) \tag{7}$$

The matrix exponential e^{Mt} is computed once; evaluating p at any (x,t) is $O(N)$.

3.2 Moments

Property 2. The n-th moment:

$$\mathbb{E}[X_t^n] = \int x^n\,p(x,t)\,dx = \sum_k A_k(t)\int x^n\,\varphi_k(x)\,dx \tag{8}$$

The integrals $c_k^{(n)} = \int x^n\varphi_k\,dx$ are precomputed constants. Each moment is an $O(N)$ inner product.

3.3 Risk Measures

Property 3 (Value-at-Risk). VaR_α is the α-quantile of the CDF:

$$F(x;t) = \sum_k A_k(t)\int_a^x \varphi_k(y)\,dy, \qquad F(\mathrm{VaR}_\alpha;t) = \alpha \tag{9}$$

The CDF has a closed-form expression from the cosine basis (sine integrals). VaR is found by root-finding on F.

Property 4 (Expected Shortfall). $\mathrm{ES}_\alpha = \frac{1}{\alpha}\int_0^\alpha \mathrm{VaR}_u\,du = \frac{1}{\alpha}\int_{-\infty}^{\mathrm{VaR}_\alpha} x\,p(x)\,dx$.

3.4 Option Prices and Greeks

Property 5 (Option Prices). For any payoff $g(x)$:

$$\mathbb{E}[g(X_T)] = \sum_k A_k(T)\int g(x)\,\varphi_k(x)\,dx = \sum_k A_k(T)\,G_k \tag{10}$$

European calls, puts, digitals, and arbitrary payoffs all reduce to inner products $\langle A(T), G\rangle$ where G_k are the payoff's cosine coefficients.

Remark (Forward vs Backward). Equation (7) evolves *density* coefficients forward via M. For *value function* evolution (backward problems, American options, optimal stopping), the adjoint $M^\top$ is used: $V(t) = e^{M^\top \Delta t}\, V(t{+}\Delta t)$. For processes with drift, $M \neq M^\top$; using the wrong operator gives $O(1)$ errors.

Property 6 (Greeks). Sensitivities to parameters are derivatives of (10):

$$\Delta = \frac{\partial}{\partial X_0}\,\mathbb{E}[g(X_T)] = \sum_k \frac{\partial A_k(T)}{\partial X_0}\, G_k \tag{11}$$

Since $A(T) = e^{MT}\, A(0)$ and $A(0)$ depends on X_0 through the initial projection, the chain rule gives analytical Greeks. Gamma, Theta, and Vega follow similarly. A desk with 1000 positions computes all Greeks from the *same* e^{MT}.

3.5 Stationary Distribution and Ergodic Properties

Property 7 (Stationary Distribution). The null eigenvector $M\pi = 0$, normalized to integrate to 1.

Property 8 (Spectral Gap / Mixing Rate). gap $= |\lambda_1|$ where λ_1 is the second-largest eigenvalue. Controls the rate $\|p(t) - \pi\| \sim e^{-|\lambda_1| t}$.

Property 9 (Autocorrelation). For a stationary process:

$$\mathrm{Cov}(X_0, X_\tau) = c^\top\, e^{M\tau}\, (c \odot \pi) - \bar{X}^2 \tag{12}$$

where $c_k = \int x\, \varphi_k(x)\, dx$ are the first-moment projections and $\odot$ is elementwise multiplication.

3.6 Structural Properties

Property 10 (Eigenvalue Spectrum). The eigenvalues $\{\lambda_n\}$ of M encode the complete timescale structure of the process. For mean-reverting processes, $|\lambda_n| \propto n$ gives the hierarchy of relaxation modes.

---

4. The Dynamic Hierarchy: From Vectors to Tensors

4.1 Static Distribution as Special Case

The Spectral Fenton Distribution (Nagy, 2026a) represents a portfolio loss CDF as a vector $\{A_k\}_{k=0}^{N-1}$. In the present framework, this is:

$$A = e^{MT}\, A(0)$$

evaluated at a single time T. The URRT (Nagy, 2026b) guarantees $N = \Theta(\log(1/\varepsilon)/\log\rho)$.

4.2 Time-Homogeneous SDE ↔ Matrix

For drift $\mu(x)$ and diffusion $\sigma(x)$ independent of t, the generator is a constant matrix M_{kj}. This is the main object of this paper. It captures:

- **Spectral Trading Theory** (Nagy, 2026): When M is *diagonal*, each mode is an independent OU process $dA_k = -(1/\tau_k)(A_k - \bar{A}_k)\,dt + \sigma_k\,dW_k$. The 14 theorems of spectral trading — risk-return tradeoff, Pythagorean Sharpe, CAPM, momentum = mean reversion — all follow from the diagonal structure of M.
- **General diffusions**: When M has off-diagonal entries, modes are coupled. The SVD of M reveals the dominant decay channels and their interactions.

4.3 Time-Inhomogeneous SDE ↔ 3-Tensor

For $\mu(x,t)$ and $\sigma(x,t)$ depending on time, the generator becomes $M(t)$. Expanding in a temporal basis $\{\psi_l(t)\}$:

$$M_{kj}(t) = \sum_l T_{kjl}\,\psi_l(t) \tag{13}$$

The 3-tensor T_{kjl} encodes the complete time-varying dynamics. The time evolution is:

$$A(t) = \mathcal{T}\exp\left(\int_0^t M(s)\,ds\right) A(0) \tag{14}$$

where $\mathcal{T}$ denotes time-ordering. For slowly varying $M(t)$, the Magnus expansion gives efficient approximations.

4.4 Multi-Dimensional SDE ↔ Higher Tensor

For d-dimensional systems $dX^i = \mu^i(X)\,dt + \sigma^{ij}(X)\,dW^j$, the Fokker–Planck operator acts on the joint density $p(x_1,\ldots,x_d,t)$. Expanding in a product cosine basis $\varphi_{k_1}(x_1)\cdots\varphi_{k_d}(x_d)$, the generator becomes a tensor of order $2d$:

$$T_{k_1\ldots k_d,\,j_1\ldots j_d}$$

The **eigenvalue conditioning trick** (the Eigen-COS method) reduces this: by diagonalizing the correlation structure first, the tensor becomes block-diagonal with blocks of order $\leq 2K$ where K is the number of retained eigenvalues. This is why the Spectral Fenton method works for $d = 100$ assets with $K = 1$–3 factors.

4.5 The URRT Controls the Tensor Rank

The URRT (Nagy, 2026b) guarantees:

$$N = \Theta\left(\frac{\log(1/\varepsilon)}{\log\rho}\right)$$

parameters suffice for ε-accuracy, independent of d. For the tensor, this means:

- The tensor entries decay as $\rho^{-(k_1+\cdots+k_d+l)}$
- Only $O(N^{d+1})$ entries are significant, but N is small ($\sim$ 20–60)

- For low-rank tensors (which the URRT ensures): CP decomposition is tractable and unique (Kruskal's theorem)

The spectral framework works precisely because the URRT guarantees fast decay → low rank → tractable decomposition.

---

5. The Quantum–Stochastic Isomorphism

The spectral generator has a precise structural analog in quantum mechanics:

	Quantum mechanics	Stochastic processes
State	Wave function ψ	Density p
Generator	Hamiltonian H	Fokker–Planck operator $\mathcal{L}$
Evolution	$i\hbar\,\partial_t\psi = H\psi$	$\partial_t p = \mathcal{L}p$
Spectral form	$H = \sum_n E_n \lvert n\rangle\langle n\rvert$	$\mathcal{L} = \sum_k \lambda_k \varphi_k \varphi_k^*$
Eigenvalues	Energy levels E_n	Decay rates $-1/\tau_k$
Observables	$\langle\psi\lvert\widehat{O}\rvert\psi\rangle$	$\int f(x)\,p(x)\,dx$
Completeness	Hamiltonian determines everything	Generator determines everything

In quantum mechanics, the claim that the Hamiltonian is a complete description of the system is uncontroversial. The spectral generator is the financial analog. The reason this has not been recognized previously is that finance has relied on Monte Carlo (sampling the *process*) rather than spectral methods (diagonalizing the *generator*).

---

6. Numerical Validation

6.1 Ornstein–Uhlenbeck: Three-Way Comparison

We validate on the OU process $dX = -\theta(X - m)\,dt + \sigma\,dW$ with $\theta = 2$, $m = 1$, $\sigma = 0.5$, starting from $X_0 = 2$, evaluated at $T = 1$.

Generator spectrum. The 48×48 matrix M is built in 0.02 seconds. Its eigenvalues are:

$$\lambda = \{0,\ -2.000,\ -4.000,\ -6.000,\ -8.001,\ -10.005,\ ...\}$$

matching the analytical $\lambda_n = -n\theta$ to 4+ decimal places.

Three-way comparison (500,000 MC paths):

Property	Spectral	Analytical	Monte Carlo	Spectral error
$\mathbb{E}[X_T]$	1.135334	1.135335	1.134431	1.5×10^{-6}
$\mathrm{Var}[X_T]$	0.061472	0.061355	0.061467	1.2×10^{-4}
Call(K=1)	0.180959	0.180879	0.180274	8.0×10^{-5}

Property	Spectral	Analytical	Monte Carlo	Spectral error
VaR(1%)	0.558548	0.559099	0.556475	5.5×10^{-4}
Stationary mean	1.000000	1.000000	—	8.9×10^{-16}
Stationary variance	0.062500	0.062500	—	4.5×10^{-9}
Spectral gap	2.0000	2.0000	—	1.4×10^{-7}

The spectral method matches analytical solutions to 4–16 significant figures. Monte Carlo, with 500K paths, achieves only 2–3 significant figures.

6.2 Double-Well: Nonlinear Bimodal Dynamics

The double-well potential $V(x) = (x^2 - 1)^2/4$ with $dX = -V'(X)\,dt + 0.7\,dW$ has no analytical solution. The drift $\mu(x) = -(x^3 - x)$ creates two metastable states at $x = \pm 1$.

Spectral generator reveals: - Spectral gap = 0.182 → tunneling time ≈ 5.5 (time to cross from one well to the other) - Bimodal stationary distribution with modes at ± 1.004 - Mean = 0 by symmetry (spectral: -3×10^{-13})

Spectral vs Monte Carlo (300K paths, starting from $X_0 = -1.5$):

T	Spectral $\mathbb{E}[X_T]$	MC $\mathbb{E}[X_T]$	Spectral VaR(5%)	MC VaR(5%)
0.5	−1.0670	−1.0690	−1.5004	−1.5021
1.0	−0.9236	−0.9236	−1.4441	−1.4424
2.0	−0.7568	−0.7567	−1.4094	−1.4087
5.0	−0.4379	−0.4344	−1.3723	−1.3721

Agreement within 0.01 at all horizons, with the spectral method requiring zero sampling and producing deterministic results.

6.3 Application: Energy Price Risk

A mean-reverting energy model $dP = \kappa(\bar{P} - P)\,dt + \sigma\sqrt{P}\,dW$ with $\kappa = 1.5$, $\bar{P} = 40$, $\sigma = 8$, $P_0 = 35$.

From one 48×48 **matrix, the complete risk profile:**

Horizon	$\mathbb{E}[P]$	$\sigma[P]$	VaR(1%)	VaR(5%)	Call(40)	Put(35)
0.10	35.69	14.02	9.76	15.38	3.83	5.19
0.25	36.56	20.13	4.48	9.66	6.52	7.07
0.50	37.61	24.68	2.63	7.13	8.65	8.15
1.00	38.77	27.58	2.16	6.37	10.16	8.52
2.00	39.51	28.49	2.15	6.36	10.80	8.45
5.00	39.70	28.62	2.16	6.39	10.93	8.41

Greeks at $T = 0.5$**,** $K = 40$**:** Delta = 0.267, Gamma = 0.004, Theta = −5.64.

Stationary risk: Mean = 39.70 ($\to \bar{P}$), long-run VaR(5%) = 6.39.

All quantities from one matrix. No Monte Carlo. The term structure, option prices, and Greeks would require millions of MC paths to compute with comparable precision.

---

7. Formal Structure

7.1 What the Completeness Theorem Says

Theorem 1 (Completeness). *Let $(X_t)_{t\geq 0}$ be a diffusion process satisfying (1) with density $p(x,t)$ satisfying the Fokker–Planck equation (2). Let M be the spectral generator (5) with N basis functions on $[a,b]$. Then for any functional Φ of the law of (X_t) that depends only on the density $p(x,t)$ for $x \in [a,b]$ and $t \geq 0$:*

$$\Phi[(X_t)_{t\geq 0}] = \Phi[e^{M\cdot} A(0)] + O(\varepsilon_N)$$

where ε_N is the spectral truncation error, bounded by:

$$\varepsilon_N \leq C \cdot \rho^{-N}, \qquad N = \Theta\left(\frac{\log(1/\varepsilon)}{\log \rho}\right) \tag{15}$$

The bound is independent of: - the specific functional Φ (distribution, moment, price, Greek, risk measure), - the time horizon T, - the dimension d (for the multi-asset extension).

7.2 What It Requires

1. **Smoothness**: The density $p(x,t)$ must be analytic on an ellipse in the complex plane (for exponential convergence) or s-times differentiable (for polynomial convergence). This holds for all Itô diffusions with Lipschitz coefficients and non-degenerate diffusion.
2. **Domain**: The truncation domain $[a,b]$ must be wide enough that $p(a,t) \approx p(b,t) \approx 0$ for all t of interest. Typically $[a,b] = [\bar{x} - 6\sigma_\infty,\ \bar{x} + 6\sigma_\infty]$.
3. **Reflecting BC**: The weak form (5) enforces zero-flux boundary conditions. This is the physical condition for processes confined to $[a,b]$.

7.3 What Extends to Tensors

For time-varying and multi-dimensional systems, the generator becomes a tensor $T_{k_1\cdots k_d, j_1\cdots j_d, l}$. The completeness theorem still holds, with N per mode controlled by the URRT. The total parameter count is $O(N^{d+1})$ for a d-dimensional time-varying process — exponential in d, but N is small ($\sim$ 20–60), and the eigenvalue conditioning trick (Eigen-COS) reduces d-dimensional problems to K-dimensional ones with $K \ll d$.

7.4 What Doesn't Fit

1. **Pure jump processes** (no diffusion component): The Fokker–Planck equation does not apply. The generator is an integral operator, not a differential one. Spectral discretization is still possible but convergence may be slower.

2. **Path-dependent quantities**: Running maximum, Asian averages, barrier crossings require augmenting the state space. The generator still works, but in higher dimension.

3. **Rough paths** ($H \leq 1/4$): Densities may not exist. The Fokker–Planck framework requires a density.

---

8. Limitations and Future Directions

8.1 Current Limitations

1. **Autocorrelation formula** (Property 9): The current implementation has a numerical issue for non-symmetric generators, leading to inaccurate autocorrelation estimates. The formula (12) is correct in principle; the implementation requires the *adjoint* generator for the backward evolution.

2. **Multi-dimensional validation**: The tensor extension (Section 4.4) is described theoretically but not yet validated computationally. The connection to the Eigen-COS method is structural — eigenvalue conditioning IS diagonalizing the generator — but a direct numerical comparison is needed.

3. **Real data**: All demonstrations use synthetic processes. Calibrating the generator to real market data (estimating $\mu(x)$ and $\sigma(x)$ from time series) and comparing with historical risk measures is the natural next step.

8.2 Future Directions

1. **Heston model**: The 2D system (S_t, v_t) with stochastic volatility is the canonical test case for the tensor extension. The generator is a $N_S \times N_v \times N_S \times N_v$ tensor, reducible by eigenvalue conditioning.

2. **American options**: The spectral generator naturally handles early exercise via the *variational inequality* formulation: replace e^{MT} with the value iteration operator. This connects to the American Basket paper (Nagy, 2026).

3. **Spectral calibration**: Instead of calibrating parametric models and then pricing, calibrate the generator M directly to option prices. The inverse problem $M \mapsto$ option prices is linear (equation 10); inverting it gives M from market data.

4. **Lean formalization**: The existing TensorURRT.lean formalizes 16 structural properties of the tensor URRT. Extending this to formalize the completeness theorem requires formalizing the Fokker–Planck semigroup properties.

---

9. Conclusion

The spectral generator matrix M is the complete, finite, computable representation of any diffusion process. One matrix encodes the distribution at every time, every moment, every option price, every Greek, every risk measure, the stationary behavior, and the mixing dynamics. The URRT guarantees the matrix is small.

The thesis can be stated in one line:

$$\boxed{\text{SDE} \longleftrightarrow \mathcal{L} \longleftrightarrow M_{kj} \longrightarrow \text{all prices, risks, dynamics, Greeks, ergodic properties}}$$

Static risk (the Spectral Fenton Distribution) is a snapshot. Dynamic risk (this paper) is the film. The matrix M is the projector.

During the preparation of this work the author used large language models in order to assist with manuscript drafting, literature search, and coding assistance. After using these tools, the author reviewed and edited the content as needed and takes full responsibility for the content of the published article.

References

- Boyd, J. P (2001). Chebyshev and Fourier Spectral Methods. *Chebyshev and Fourier Spectral Methods*. DOI: 10.1007/978-3-642-83876-7
- Fang, Fang and Oosterlee, Cornelis W. (2008). A Novel Pricing Method for European Options Based on Fourier-Cosine Series Expansions. *SIAM Journal on Scientific Computing*, 31(2), 826-848. DOI: 10.1137/080718061
- Gottlieb, D. and S. A. Orszag (1977). Numerical Analysis of Spectral Methods. *Numerical Analysis of Spectral Methods*.
- Kolda, T. G. and B. W. Bader (2009). Tensor decompositions and applications. *SIAM Review*, 51(3), 455-500. DOI: 10.1137/07070111x
- Nagy, T. (2026). Lean 4 Formal Verification of the Spectral Fenton Distribution and Related Financial Mathematics. *Working paper*.
- Nagy, T. (2026). The Spectral Tensor Representation of Stochastic Processes. *Working paper*.
- Nagy, T. (2026). Exact Portfolio VaR Without Monte Carlo: The Eigen-COS Method. *Zenodo*. DOI: 10.5281/zenodo.18910516
- Nagy, T. (2026). Frequency-Domain Theory of Financial Economics: Thirteen Fundamental Results from One Decomposition. *Working paper*.
- Nagy, T. (2026). Pricing Basket Options via Eigenvalue-Conditional Black-Scholes Mixing. *Zenodo*. DOI: 10.5281/zenodo.18910542
- Oseledets, I. V (2011). Tensor-Train decomposition. *SIAM Journal on Scientific Computing*, 33(5).

Appendix A: Reproducibility

All numerical results in this paper are produced by examples/spectral_tensor_completeness.py in the project repository. The script is self-contained (requires only NumPy and SciPy) and runs the three demonstrations in sequence:

```
python3 examples/spectral_tensor_completeness.py
```

Runtime: $ $4 minutes on a single core (dominated by Monte Carlo validation). The spectral computations alone take < 1 second per demonstration.

Appendix B: Code for the Spectral Generator

The core computation (equation 5) in vectorized NumPy:

```
# Gauss-Legendre quadrature
nodes, weights = np.polynomial.legendre.leggauss(Q)
x_q = 0.5*(b-a)*nodes + 0.5*(a+b)
w_q = 0.5*(b-a)*weights

# Evaluate SDE coefficients at quadrature points
mu_vals = np.vectorize(drift)(x_q)
D_vals = 0.5 * np.vectorize(diffusion)(x_q)**2
advection = mu_vals - numerical_derivative(D_vals)

# Cosine basis and derivatives: _k, _k' at all quadrature points
# ...

# Integration-by-parts weak form (eq. 5)
M = (dphi * (advection * w_q)) @ phi.T  \
  - (dphi * (D_vals * w_q))    @ dphi.T
```

Ten lines. The generator of any 1D diffusion.