Quantum Variables in Finance

Lester Ingber
Physical Studies Institute LLC
Hillsboro, Oregon

ABSTRACT

Background: A path-integral algorithm, PATHINT used previously for several systems, has been generalized from 1 dimension to N dimensions, and from Classical to Quantum systems into qPATHINT. Previous publications applied qPATHINT to two systems developed by the author, intercortical interactions and financial options. Classical PATHINT also has been published demonstrating development of Eurodollar options in industrial applications.

Objective: Historical Volatility (HstVol) or Implied Volatility (ImpVol) (preferred by traders) from Strike data are used and are required to price financial options. This proposal creates tables of ImpVol, first for Classical and Quantum old (circa 2000) Eurodollars which benchmarks all codes to previous publications, then for Quantum Bitcoin which is among the most volatile currencies. Only Classical (super)computers are required here for all calculations.

Method: The system is developed using mathematical-physics methods of path integrals in Quantum spaces. Supercomputer pilot studies using XSEDE.org and StonyBrook.edu Ookami resources tested various dimensions for their scaling limits. For this study, ImpVols and all traded Greeks are calculated for options in Quantum-money spaces, including realistic shocks (dividends, etc.).

Results: The mathematical-physics and computer parts of the study are successful for both systems. A 3-dimensional path-integral propagation of qPATHINT for both systems is within normal computational bounds on supercomputers, but here only 1-dimensional path-integrals are required.

Conclusion: Each of the two systems considered have contributed insight into applications of qPATHINT to the other system, leading to new algorithms presenting time-dependent propagation of interacting Quantum and Classical scales.

Keywords: Nonlinear Stochastic Systems, Quantum Systems, Importance-Sampling Optimization, Financial Markets

1. Executive Summary

Quantum computing is here, and in the near future this is expected to include financial products with blockchains. It not far-fetched to assume that soon derivatives will be developed on these products. As is the case in Classical real spaces with PATHTREE and PATHINT, qPATHTREE and qPATHINT are poised to calculate derivatives in Quantum complex spaces. This is beyond using Quantum computation of derivatives, since the space of the dependent variables themselves may live in Quantum worlds (Baaquie et al., 2002; Piotrowski et al., 2005; Accardi & Boukas, 2007; Meyer, 2009; Aaronson & Christiano, 2012; Jogenfors, 2016). Quantum and Classical are capitalized here if not in References.

This proposal puts equations in appendices: Section 2 Appendix A is Previous Benchmarks. Section 3 Appendix B is Quantum Money. Section 4 Appendix C is Path Integrals. Section 5 Appendix D is Adaptive Simulated Annealing (ASA).
1.1. Path Integral Applications


1.2. Statistical Mechanics of Neocortical Interactions (SMNI)

Path integrals have been used by the author primarily in Statistical Mechanics of Neocortical Interactions (SMNI) studies. A StonyBrook.edu Ookami project, "Statistical Mechanics of Neocortical Interactions: Quantum influences on neocortical information processing," ongoing since 2012, has produced several relevant papers discussing the path-integral algorithm (Ingber, 2022b; Ingber, 2021b), including papers published at XSEDE.org under grants dating from February 2013 through December 2021 which contain more SMNI references (Ingber, 2018; Ingber, 2019; Ingber, 2021b; Ingber, 2021c; Ingber, 2022b). This work included the author’s closed-form Quantum path-integral solutions for tripartite (neuron-astrocyte-neuron) interactions.

1.3. Statistical Mechanics of Financial Markets (SMFM)

Another project since 2017, using similar mathematical-physics tools, was ongoing at XSEDE.org until 2021, "Quantum Options," under Grant TG-MCB140110 (Ingber, 2017c; Ingber, 2021a).

The codes developed enable calculation of wave functions $\psi$ and moments of physical operators $O$, $\langle \psi^* | O | \psi \rangle$. To numerically calculate the path integral, especially for serial changes in time — not approachable with standard Monte Carlo techniques — PATHINT was developed.

Note that the financial applications yet have no real-world applications, and likely will not for a decade when Quantum computers will become mainstream, and then Quantum money will become a reality, if no other reason than Quantum states cannot be cloned/counterfeited (Aaronson & Christiano, 2012; Jogenfors, 2016; Ingber, 2017b). At that time, to control volatility, options on Quantum money will be required, and the calculations performed by qPATHINT and qPATHTREE to calculate options present all Greeks necessary for trading.

Historical Volatility (HstVol) or Implied volatility (ImpVol) (preferred by traders) from Strike data are essential for trading options, i.e., setting fair prices for options (Ingber, Chen et al, 2001). This requires searching for minima in the nonlinear volatility variable fit to Strike data, which is most influential in options pricing. Since cryptocurrencies currently exhibit the largest swings in volatility (Zulfiqar & Gulzar, 2021) they will be fit after baselining the code to Eurodollars. Probabilities will be based on fits to seasonal real data, the expressions will be quite nonlinear in volatility, and so the author’s ASA (under a public license and available to all without any registration at ingber.com) is used (Ingber, 1993a).

However, it is likely that not every fit requires the full power (and time) of ASA. As pointed in other publications (Ingber, 2020), after a full ASA fit for the first of a given market, the Simplex code that comes with the ASA code (used efficiently to get the last few decimal places) can be used to modify the original fitted parameter(s).

The Classical counterparts to Quantum options have used path-integral methods successfully to calculate options with nonlinear and time-dependent multivariate options (Ingber, 2000).

1.4. Approximate State of Quantum Hybrid Computing

As of 2020 (Ingber, 2021c), there are several companies now offering commercial-grade Hybrid Classical-Quantum computers, e.g., Rigetti, D-Wave, Google, Microsoft, and IBM; see

https://www.rigetti.com/what
https://www.ibm.com/it-infrastructure/z/capabilities/hybrid-cloud
Hybrid computers typically offer Classical computers to optimize parameters in systems that are described by Quantum variables using Quantum computers (Benedetti et al., 2019). Other studies show Quantum computing still out of reach for many systems, even with Classical optimizers (Chakrabarti et al., 2020). Software for Quantum states also has been developed, e.g., Tensorflow, the popular end-to-end open-source tool for machine learning, also has been adapted for Hybrid Classical-Quantum computing:

https://quantumzeitgeist.com/tensorflow-for-quantum-hits-first-birthday/
https://www.tensorflow.org/quantum

It has been noted that SMNI can offer more detail to many of these neural-net projects (Ingber, 2022c).

The PI has accounts on D-Wave and Rigetti computers, but has not yet ported SMFM code to these Quantum computers for Hybrid computations.

This project merges codes generated for two previous XSEDE grant proposals, “Electroencephalographic field influence on calcium momentum waves” (Ingber, 2018) and “Quantum path-integral qPATHTREE and qPATHINT algorithms” (Ingber, 2017a; Ingber, 2017b). These codes both run on Classical computers and thereby define a Hybrid Classical-Quantum system that can be calculated solely on a Classical computer.

2. Appendix A: Previous Benchmarks

The Ookami SMNI neuroscience benchmarks are fully in the context of the proposed finance project, since similar codes, e.g., ASA and [q]PATHINT are to be used here as well.

On the author’s Ookami Testbed account benchmarks were performed on two codes using all 4 compilers available, with and without vectorization. The optimization code Adaptive Simulated Annealing (ASA) has hooks for OpenMP. 100000 iterations/cost functions are required for a complete run.

Each time was obtained from ‘time make run’ after using ‘make compile’. Below are the time estimates. Based on the data below, time for each full run is obtained as 100000 * (0.025 + 10 * 0.07) = 72500s or about 20 hrs/run for 1 subject for one set of data.

Each full run uses 12 subjects, with 2 runs per subject (Train/Test, and switched) = 24 array entries per run. On TACC, multiple instances of a serial job on a single node are run, as described on https://portal.tacc.utexas.edu/software/launcher.

100 such runs for debugging and production would give an estimate required of 48000 hrs. If required, the author’s ASA code can limit each run to less than any number of hrs, saving all data required for continued runs.

```
project path_arm_noVec
12.170u 0.114s 0:12.56 97.7% 0+0k 0+56960io 0pf+0w
```

```
project path_arm_Vec
12.198u 0.069s 0:13.27 92.3% 0+0k 0+57088io 0pf+0w
```

```
project path_cc_noVec
9.224u 0.122s 0:09.50 98.3% 0+0k 0+56960io 0pf+0w
```

```
project path_cc_Vec
9.164u 0.135s 0:09.48 97.9% 0+0k 768+56960io 0pf+0w
```

```
project path_fcc_noVec
9.251u 0.097s 0:10.13 92.2% 0+0k 0+56960io 0pf+0w
```

```
project path_fcc_Vec
9.265u 0.067s 0:09.52 97.8% 0+0k 2432+56960io 10pf+0w
```

```
project path_gcc_noVec
8.740u 0.107s 0:08.94 98.8% 0+0k 0+57088io 0pf+0w
```

```
project path_gcc_Vec
8.858u 0.067s 0:09.18 97.0% 0+0k 128+56960io 0pf+0w
```

```
project smni_arm_noVec
34.161u 0.495s 0:35.54 97.4% 0+0k 0+33408io 0pf+0w
```

```
project smni_arm_Vec
34.066u 0.576s 0:36.13 95.8% 0+0k 128+33152io 0pf+0w
```
The results sorted on secs yield
pathfcc_Vec 0.067s
pathgcc_Vec 0.067s
patharm_Vec 0.069s
smni_arm_Vec_openmp 0.242s
smni_cc_noVec
5.213u 0.242s 0:05.57 97.8% 0+0k 0+27264io 0pf+0w
0+0k 0+33408io 0pf+0w
0+128+33408io 0pf+0w
0+768+27264io 0pf+0w
0+0k 128+33408io 0pf+0w
0+0k 0+33280io 0pf+0w
0+0k 0+33152io 0pf+0w
0+0k 11520+33152io 0pf+0w
0+256+27520io 0pf+0w
 GCC performed very well and was used thereafter.

3. Appendix B: Quantum Money
It is reasonable to assume that soon there will be financial derivatives developed on all products. qPATHTREE and qPATHINT can calculate financial derivatives in these spaces. This is beyond using Quantum computation of Classical financial derivatives, since the space of the dependent variables themselves may live in Quantum worlds (Baaquie et al., 2002; Piotrowski et al., 2005; Accardi & Boukas, 2007; Meyer, 2009; Aaronson & Christiano, 2012; Jogenfors, 2016).

3.1. Previous Applications — PATHINT
Options $V$ are generally described by a portfolio $\Pi$ over an underlying asset $S$. The real-world probability distribution of $S$ often is critical to numerical details for trading. The asset is often hedged by trading the option $V$ and a quantity $\Delta$ of the asset $S$.

$$d\Pi = \sigma \left( \frac{\partial V}{\partial S} - \Delta \right) dX + \left( \mu S \frac{\partial V}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + \frac{\partial V}{\partial t} - \mu S \right) dt$$

$$\Gamma = \frac{\partial^2 \Pi}{\partial S^2}, \quad \Theta = \frac{\partial \Pi}{\partial t}, \quad \Xi = \frac{\partial \Pi}{\partial \sigma}, \quad \rho = \frac{\partial \Pi}{\partial \sigma}$$

This defines the “Greeks” used in trading. The portfolio $\Pi$ to be hedged is often considered to be “risk-neutral,” if $\Delta$ is chosen such that $\Delta = \frac{\partial V}{\partial S}$.

Quite a few closed-form solutions exist for European options (Black & Scholes, 1973) where there is not early exercise. For American options — among the most popular options — there is no general closed
form, and numerical calculations must be performed (Hull, 2000): First the probability “tree” for $S$ is propagated forward in time until the expiration date $T$, branching out as extended $S$ values develop; then, marching back in time, at each time-node calculations of the Greeks above are developed, inserting changes in dividends, interest rates, changes in cheapest-to-deliver for baskets of bonds for options on bond futures, etc (Ingber & Wilson, 1999; Ingber, 2000).

At each node a calculation is performed, comparing the strike price $X$ to the price $S$ at that node, and a decision is made, e.g., whether to exercise the option at that node, which determines the fair value of the option price $V$. To obtain the Greeks, extended derivatives of these derivatives are calculated numerically by using values across branches and notes (Ingber, 2000).

3.2. Volatility of Volatility of American Options

An example of a two-dimensional options model processed by PATHINT developed the volatility of volatility of options on old (circa 2000) Eurodollars, using 2-factor model developed by the author (Ingber, 2000):

$$dS = \mu S dt + \sigma F(S, S_0, S_\infty, x, y) d\zeta_S$$
$$d\sigma = \nu dt + \epsilon d\zeta_{\sigma}$$

$$F(S, S_0, S_\infty, x, y) = S, \ S < S_0$$
$$F(S, S_0, S_\infty, x, y) = S^x S_0^{1-x}, \ S_0 \leq S \leq S_\infty$$

$$F(S, S_0, S_\infty, x, y) = S^y S_0^{1-y} S_\infty^{1-x}, \ S > S_\infty$$ (2)

where $S_0$ and $S_\infty$ are selected to lie outside the data region used to fit the other parameters, e.g., $S_0 = 1/2$ and $S_\infty = 20$ for fits to Eurodollar futures.

3.3. SMFM Example of 2-Factor qPATHINT

qPATHINT was used to perform the similar calculations as were previously performed by PATHINT (Ingber, 2000), except that whereas PATHINT propagates the probability, qPATHINT propagates the wave-function $\psi$ and the kernels are normalized with $\psi^* \psi$. Since real variables were input, the output is the same for test sections of the code as previously reported using one-dimensional closed-form Black-Scholes (Black & Scholes, 1973) and numerical Cox-Ross-Rubenstein (CRR) codes (Cox et al., 1979). The output differs for the 2-dimensional volatility of volatility calculations, as $\sigma$ requires averaging. This was another test for qPATHINT, illustrating that if complex variables were input (including real and imaginary non-zero values), then this can be processed by this code.

3.4. qPATHINT: Inclusion of Quantum Scales

PATHINT was developed to numerically calculate the path integral, especially for serial changes in time, usually not approachable with standard Monte Carlo techniques.

qPATHINT was developed from the PATHINT C code of about 7500 lines of code. The C code run with the GCC C-compiler uses double complex variables instead of double variables. The code is written for arbitrary N dimensions. The outline of the code is described here for Classical or Quantum systems, using generic coordinates $q$ and $x$ (Ingber, 2016; Ingber, 2017a; Ingber, 2017b):

This histogram procedure recognizes that the distribution (probabilities for Classical systems, wave-functions for Quantum systems) can be numerically approximated by sums of rectangles of height $P_i$ and width $\Delta q^i$ at points $q^i$.

3.5. Shocks

Many real-world systems propagate in the presence of continual “shocks” (Ingber, 2017b; Ingber, 2017a):

future dividends
changes in interest rates
changes in asset distributions used in American options algorithms

3.6. PATHINT/qPATHINT Histograms

In the prepoint Ito discretization, using a one-dimensional system in variable \(x\), the path-integral is written in terms of the kernel \(G\), for each of its intermediate integrals, as

\[
P(x; t + \Delta t) = \int dx'[g^{1/2}(2\pi\Delta t)^{-1/2} \exp(-L \Delta t)]P(x'; t) = \int dx'G(x, x'; \Delta t)P(x'; t)
\]

\[
P(x; t) = \sum_{i=1}^{N} \pi(x - x^i)P_i(t)
\]

\[
\pi(x - x^i) = 1, \quad (x^i - \frac{1}{2} \Delta x^{i-1}) \leq x \leq (x^i + \frac{1}{2} \Delta x^i) ; 0, \quad \text{otherwise}
\]  
(3)

This yields

\[
P_i(t + \Delta t) = T_{ij}(\Delta t)P_j(t)
\]

\[
T_{ij}(\Delta t) = \frac{2}{\Delta x^{i-1} + \Delta x^i} \int_{x_i - \Delta x^{i-1/2}}^{x_i + \Delta x^{i-1/2}} dx \int_{x_{i+\Delta x^i/2}}^{x_i + \Delta x^{i/2}} dx'G(x, x'; \Delta t)
\]  
(4)

\(T_{ij}\) is a banded matrix representing the Gaussian nature of the short-time probability centered about the (possibly time-dependent) drift.

Several projects have used this algorithm (Wehner & Wolfer, 1983a; Wehner & Wolfer, 1983b; Wehner & Wolfer, 1987; Ingber & Nunez, 1995; Ingber, Srinivasan & Nunez, 1996; Ingber & Wilson, 1999). 2-dimensional codes were developed for Statistical Mechanics of Combat (SMC), Statistical Mechanics of Neocortical Interactions (SMNI) and Statistical Mechanics of Financial Markets (SMFM) (Ingber, Fujio & Wehner, 1991; Ingber & Nunez, 1995; Ingber, 2000).

3.7. Meshes For [q]PATHINT

Explicit dependence of \(L\) on time \(t\) also can be included. The mesh \(\Delta q^i\) is strongly dependent on diagonal elements of the diffusion matrix, e.g.,

\[
\Delta q^i = (\Delta g^{[ii]}|^i)^{1/2}
\]  
(5)

The covariances develop a rectangular-ish underlying mesh. Fitting the data with integrals over the short-time distribution permits the coarser meshes than the corresponding stochastic differential equation(s) (Ingber, 1990; Wehner & Wolfer, 1983a). Here, vertical bars on \([ii]\) imply no summation.

3.8. Broad-Banded Kernels Required

SMNI qPATHINT requires broad-banded kernels for oscillatory Quantum states.

SMFM PATHTREE and qPATHTREE are different options codes based on path-integral studies permitting a new very fast binary calculation applied to nonlinear time-dependent systems (Ingber, Chen et al, 2001). However, currently an SMFM [q]PATHTREE is only a binary tree with \(J = 1\) and it cannot be effectively applied to Quantum oscillatory systems (Ingber, 2016; Ingber, 2017a; Ingber, 2017b).

3.9. SMFM qPATHINT With Serial Shocks

Some N-dim qPATHINT runs for SMFM used a contrived N-factor model with the same 1-dimensional system cloned in all dimensions (each unit is a "double complex"). The kernel size is \((IJ)^N\), where \(I = \text{imxall}, J = \text{jmxall} (= \text{kernel band width})\), and kernel size = ijkcnt. This spatial mesh might change at each time slice.

\[
D=1:\text{imxall: 27, jmxall: 7, ijkcnt: 189}
\]

\[
D=2:\text{imxall: 729, jmxall: 49, ijkcnt: 35721}
\]
A full set of Adaptive Simulated Annealing (ASA) (Ingber, 1993a) fits of Classical SMNI to EEG data takes about 1000 hours of supercomputer CPUs; many such sets of runs are required to develop good models. Cost functions including Quantum processes take even longer.

3.10. Applications

3.10.1. Quantum Money and Blockchains

Soon financial derivatives will be developed on these products. qPATHINT can calculate financial derivatives in these spaces. The marketplace will determine traded variables.

3.10.2. Enhanced Security/Verification

Quantum currencies cannot be cloned. Such currencies are candidates for efficient blockchains. Each “coin” has a unique identity (Meyer, 2009; Aaronson & Christiano, 2012; Bartkiewicz et al, 2016; Jogenfors, 2016). There are issues about the decoherence time of such currency that are being addressed, e.g., by using topological qubits (Quantum bits).

4. Appendix C: Path Integrals

4.1. Path Integral in Stratonovich (Midpoint) Representation

Most suitable for examining discretization issues in time-dependent nonlinear systems, the path integral in the Feynman (midpoint) representation is (Langouche et al, 1979; Schulman, 1981; Langouche et al, 1982) (N.b. $g^3$ in $DM$ implies a prepoint evaluation.) Unless stated otherwise, the Einstein summation convention is used for repeated indices to signify summation.

$$P[M_i|M_{i_0}]dM(t) = \int \cdots \int DM \exp \left( -\min_{t_i} \int_{t_0}^t dt' L \right) \delta(M(t_0) = M_0) \delta(M(t) = M_i)$$

$$DM = \lim_{u \to \infty} \lim_{\rho \to 0} \prod_{\rho=1}^{u+1} g^{+1/2} (2\pi \theta)^{-1/2} dM^G_{\rho}$$

$$L(\dot{M}^G, M^G, t) = \frac{1}{2} (\dot{M}^G - h^G) g_{GG} (\dot{M}^G - h^G) + \frac{1}{2} h^G \cdot :G + R/6 - V$$

$$\dot{M}^G (t) \rightarrow M^G_{\rho+1} - M^G_{\rho} , M^G(t) \rightarrow \frac{1}{2} (M^G_{\rho+1} + M^G_{\rho}) \cdot [\cdot \cdot]_G = \frac{\partial [\cdot \cdot \cdot]}{\partial M^G}$$

$$h^G = g^G - \frac{1}{2} g^{-1/2} (g^{1/2} g_{GG}^G) , h^G \cdot :G = h^G + \Gamma^F_{GF} h^G = g^{-1/2} (g^{1/2} h^G) , g_{GG} = \det(g_{GG}^G)$$

$$\Gamma^F_{JK} \equiv g_{JK}^{LF} [JL, L] = g_{JKL}^{LF} (g_{JL,K} + g_{KL,J} - g_{JK,L})$$

$$R = g^{JL} R_{JL} = g^{JL} g^{JK} R_{FJKL}$$

$$R_{FJKL} = \frac{1}{2} (g_{FK,JL} - g_{FK,FL} - g_{FL,JK} + g_{JL,FK}) + g_{MN} (\Gamma^M_{FK} \Gamma^N_{JL} - \Gamma^M_{FL} \Gamma^N_{JK})$$
4.2. Path Integral in Ito (Prepoint) Representation

For probability distributions or for wave functions the Ito development in the prepoint representation:

\[ P[M_t|M_{t_0}]dM(t) = \int \cdots \int DM \exp(-\min_{t_0} \int_{t_0}^t dt'L)\delta(M(t_0) = M_0)\delta(M(t) = M_t) \]

\[ DM = \lim_{u \to \infty} \prod_{\rho=1}^{u+1} g^{1/2} \prod_{\rho} (2\pi \Delta t)^{-1/2} dM_{\rho}^G \]

\[ L(\dot{M}^G, M^G, t) = \frac{1}{2} (\dot{M}^G - g^G)g_{GG'}(\dot{M}^G - g^G') + R/6 \]

\[ \dot{M}^G(t) \to M_{\rho+1}^G - M_{\rho}^G, \quad M^G(t) \to M_{\rho}^G \]

\[ g_{GG'} = (g^{GG'})^{-1}, \quad g = \det(g_{GG'}) \]

The diagonal diffusion terms are \( g^{[GG]} \) and the drift terms are \( g^G \); Here, vertical bars on \( |GG| \) imply no summation. No-constant diffusions terms add terms to the drifts. A Riemannian-curvature potential \( R/6 \) arises for dimension \( \geq 1 \) in the midpoint Stratonovich/Feynman discretization for nonlinear systems.

4.3. Path-Integral Riemannian Geometry

The midpoint derivation derives a Riemannian geometry with a metric defined by the inverse of the covariance matrix

\[ g_{GG'} = (g^{GG'})^{-1} \]

and where \( R \) is the Riemannian curvature

\[ R = g^{IL}R_{JL} = g^{IL}g^{JK}R_{FJKL} \]

The Ito prepoint discretization gives a simpler algebraic form,

\[ M(t_s) = M(t_s) \]

\[ L = \frac{1}{2} (dM^G/dt - g^G)g_{GG'}(dM^G/dt - g^G') - V \]

but the Lagrangian \( L \) is useful only in the weak-noise limit. This means that finer meshes are often required for the prepoint case.

4.4. Three Approaches Are Mathematically Equivalent

Three basic different approaches are mathematically equivalent:

(a) Fokker-Planck/Chapman-Kolmogorov partial-differential equations
(b) Langevin coupled stochastic-differential equations
(c) Lagrangian or Hamiltonian path-integrals, as defined above

The path-integral approach is particularly useful to precisely define intuitive physical variables from the Lagrangian \( L \) in terms of its underlying variables \( M^G \):

Momentum: \( \Pi^G = \frac{\partial L}{\partial (\partial M^G/\partial t)} \)

Mass: \( g_{GG'} = \frac{\partial L}{\partial (\partial M^G/\partial t)\partial (\partial M^G/\partial t)} \)

Force: \( \frac{\partial L}{\partial M^G} \)
\[ F = ma: \quad \delta L = 0 = \frac{\partial L}{\partial M^G} - \frac{\partial}{\partial t} \left( \frac{\partial L}{\partial (\partial M^G/\partial t)} \right) \]  

(11)

The Laplacian also can be considered as an “average” over the Lagrangian (Ingber, 2022a).
Differentiation of noisy systems can introduce more noise, whereas integration is inherently a smoothing process and the path-integral often gives superior numerical performance to other representations below.

4.4.1. Stochastic Differential Equation (SDE)

The Stratonovich (midpoint discretized) Langevin equations can be analyzed in terms of the Wiener process \( dW \), which can be rewritten in terms of Gaussian noise \( \eta^i = dW^i / dt \) if care is taken in the limit.

\[
dM^G = f^G(t, M(t)) dt + \hat{g}^G(t, M(t)) dW^i
\]

\[
\dot{M}^G(t) = f^G(t, M(t)) + \hat{g}^G(t, M(t)) \eta^i(t)
\]

\[
dW^i \rightarrow \eta^i dt
\]

\[ M = \{ M^G, G = 1, \ldots, \Lambda \} \]

\[ \eta = \{ \eta^i; i = 1, \ldots, N \} \]

\[ \dot{M}^G = dM^G / dt \]

\[ \langle \eta^i(t) \rangle_\eta = 0, \quad \langle \eta^i(t), \eta^j(t') \rangle_\eta = \delta^i_j \delta(t - t') \]  

(12)

\( \eta^i \) represents Gaussian white noise.

4.4.2. Partial Differential Equation (PDE)

The Fokker-Planck, or Chapman-Kolmogorov, partial differential equation is:

\[ P_{,G} = \frac{1}{2} (g^{GG'} P)_{,GG'} - (g^G P)_{,G} + VP \]

\[ P = \langle P \eta \rangle_\eta \]

\[ g^G = f^G + \frac{1}{2} \hat{g}^G \hat{g}^G_{,G} \]

\[ g^{GG'} = \hat{g}^G \hat{g}^G_{,G} \]

\[ (\cdots)_{,G} = \partial (\cdots) / \partial M^G \]  

(13)

\( g^G \) replaces \( f^G \) in the SDE if the Ito (prepoint discretized) calculus is used to define that equation. If boundary conditions are added as Lagrange multipliers, these enter as a “potential” \( V \), creating a Schrodinger-type equation.

4.5. Applications

5. Appendix D: Adaptive Simulated Annealing (ASA)

5.1. Importance Sampling

Nonlinear and/or stochastic systems often require importance-sampling algorithms to scan or to fit parameters. Methods of simulated annealing (SA) are often used. Proper annealing (not “quenching”) possesses a proof of finding the deepest minimum in searches.

The ASA code is open-source software, and can be downloaded and used without any cost or registration at https://www.ingber.com/#ASA (Ingber, 1993a; Ingber, 2012a).

This algorithm fits empirical data to a cost function over a $D$-dimensional parameter space, adapting for varying sensitivities of parameters during the fit.

This ASA algorithm, with a true exponential annealing schedule, is faster than fast Cauchy annealing, which has schedule $T_i = T_0/k$, and much faster than Boltzmann annealing, which has schedule $T_i = T_0/\ln k$ (Ingber, 1989).

5.2. Outline of ASA Algorithm

For parameters

$$\alpha_i^k \in [A_i, B_i]$$

sampling with the random variable $x^i$

$$x^i \in [-1, 1]$$

$$\alpha_i^{k+1} = \alpha_i^k + x^i (B_i - A_i)$$

the default generating function is

$$g_T(x) = \prod_{i=1}^{D} \frac{1}{2 \ln(1 + 1/T_i) (|x^i| + T_i)} \equiv \prod_{i=1}^{D} g_i^T(x^i)$$

in terms of parameter “temperatures”

$$T_i = T_{i0} \exp(-c_i k^{1/D})$$

(14)

The default ASA uses the same type of annealing schedule for the acceptance function $h$ as used for the generating function $g$.

All default functions in ASA can be overridden with user-defined functions.

5.3. ASA Applications

The ASA code (Ingber, 1993a) and the original Very Fast Simulated Reannealing (VFSR) code (Ingber, 1989) have been used by many researchers, including the author, in several disciplines:

- chaotic systems (Ingber, Srinivasan & Nunez, 1996)
- combat simulations (Ingber, 1993c; Ingber, 1998a)

Acknowledgements

The author thanks both XSEDE.org (now ACCESS-CL.org) for supercomputer resources from Feb 2013 through 2021, and the Ookami supercomputer at StonyBrook.edu from Jun 2021 through May 2023.
REFERENCES


$Id: https://www.ingber.com/markets22_quantum_options.pdf 1.55 2022/12/17 15:58:32 ingber Exp ingber$