Computation of the diffusion equation path integrals via tensor train decomposition

Mikhail S. Litsarev, Ivan V. Oseledets

Skolkovo Institute of Science and Technology

August 25, Dubna, MPAMCS 2014

Let us consider a one-dimensional diffusion equation with the initial distribution $f(x) : \mathbb{R} \to \mathbb{R}$ and a constant diffusion coefficient σ

$$\begin{cases} \dfrac{\partial u(x,t)}{\partial t} = \sigma \, \dfrac{\partial^2 u(x,t)}{\partial x^2} - \nu(x,t) u(x,t), \\ u(x,0) = f(x). \end{cases}$$

This equation describes a Brownian particle motion, where $\nu(x,t) : \mathbb{R} \times [0,t] \to [0,\infty)$ means a probability density of annihilation of Brownian particles, $f(x)$, $\nu(x,t)$ are continuous and bounded for all x in their domain of definition.

According to the Feynman-Kac formula the solution of equation (2) is written in the path integral form

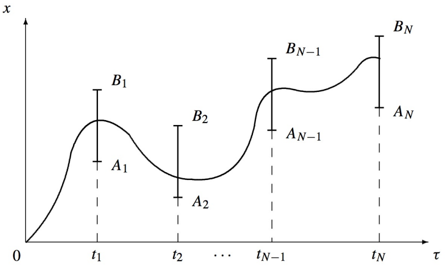
$$u(x,t) = \int_{\mathcal{C}\{x,0;t\}} \left[f(\xi(t)) \exp\left(- \int_0^t \nu(\xi(\tau), t-\tau) d\tau \right) \right] \mathcal{D}_\xi(\tau)$$

where the integration is done over a set of all possible trajectories starting at $\xi(0) = x$ with arbitrary endpoints at time t, $\mathcal{D}_\xi(\tau)$ is the Wiener measure and $\xi(t)$ is the Wiener process.

$$\int_{\mathcal{C}\{x,0;t\}} f(\xi(t))\mathcal{D}_\xi(\tau) \approx \int_{(n)} f(\xi^{(n)})\mathcal{D}_\xi^{(n)} =$$

$$= \int_{-\infty}^{\infty} d\xi_1 \frac{\exp\left(-\frac{\xi_1^2}{4\sigma\delta t}\right)}{\sqrt{4\pi\sigma\delta t}} \int_{-\infty}^{\infty} d\xi_2 \frac{\exp\left(-\frac{\xi_2^2}{4\sigma\delta t}\right)}{\sqrt{4\pi\sigma\delta t}} \times \dots$$

$$\dots \times \int_{-\infty}^{\infty} d\xi_n \frac{\exp\left(-\frac{(\xi_n)^2}{4\sigma\delta t}\right)}{\sqrt{4\pi\sigma\delta t}} f(x + \xi_1 + \dots + \xi_n)$$



$$\xi^{(k)} = \xi(\tau_k) = x + \xi_1 + \xi_2 + \dots + \xi_k,$$

$$\tau_k = k \cdot \delta t, \qquad 0 \le k \le n,$$

$$n : \tau_n = t.$$

Consider a particle on a one-dimensional infinite chain moving from the node i to node j with probability

$$W(ia - ja, \varepsilon) = \begin{cases} \frac{1}{2} & \text{if} \quad |i-j| = 1, \\ 0, & \text{otherwise} \end{cases} \qquad i, j \in \mathbb{Z}$$

After n steps

$$W(ia{-}ja, n\varepsilon) = \begin{cases} 0 & \text{if} \quad |i-j| > n \quad \text{or} \quad i-j+n \quad \text{is odd,} \\ \frac{1}{2^n}\binom{n}{\frac{n+i-j}{2}} & \text{if} \quad |i-j| \le n \quad \text{and} \quad i-j+n \quad \text{is even} \end{cases}$$

<u>The Brownian motion:</u>

– homogenious in space $(i-j)$

– homogenious in time

– isotropic $(i, j) \to (-i, -j)$

$$W(ia - ja, n) \to w(ia, n\varepsilon) = w_i(n)$$

$$w_i(n) = \begin{cases} 0 & \text{if} \quad |i| > n \quad \text{or} \qquad i+n \quad \text{is odd,} \\ \dfrac{1}{2^n}\dbinom{n}{\frac{n+i}{2}} & \text{if} \quad |i| \leq n \quad \text{and} \quad i+n \quad \text{is even} \end{cases} \tag{1}$$

$$\binom{n+1}{k} = \binom{n}{k} + \binom{n}{k-1} \tag{2}$$

$$w(x, t+\varepsilon) = \frac{1}{2}w(x+a, t) + \frac{1}{2}w(x-a, t) \tag{3}$$

$$x = ia, \qquad t = n\varepsilon \tag{4}$$

$$\frac{w(x,t+\varepsilon)-w(x,t)}{\varepsilon}=\frac{a^2}{2\varepsilon}\frac{w(x+a,t)-2w(x,t)+w(x-a,t)}{a^2} \tag{5}$$

$$\sigma=\frac{a^2}{2\varepsilon} \tag{6}$$

$$\frac{\partial}{\partial t}w(x,t)=\sigma\frac{\partial^2}{\partial x^2}w(x,t) \tag{7}$$

$$w(x,t)=\frac{1}{\sqrt{4\pi\sigma t}}\exp\left(-\frac{x^2}{4\sigma t}\right), \qquad f(x)=\delta(x) \tag{8}$$

$$w(x,t)=\frac{1}{\sqrt{4\pi\sigma t}}\int_{-\infty}^{\infty}f(y)\exp\left(-\frac{(x-y)^2}{4\sigma t}\right)dy \tag{9}$$

Exact solution:

$$u(x,t) = \int_{\mathcal{C}\{x,0;t\}} \left[f(\xi(t)) \exp\left(- \int_0^t \nu(\xi(\tau), t-\tau) d\tau \right) \right] \mathcal{D}_\xi(\tau)$$

corresponding discrete version:

$$u^{(n)}(x,t) = \int_{-\infty}^{\infty} \mathcal{D}_\xi^{(n)}\, f\left(\xi^{(n)}\right) \prod_{i=0}^{n} \exp\left(-\eta_i \nu(\xi^{(i)}, \tau_{n-i}) \delta t \right),$$

Time discretization in the exponent

$$\int_0^t \nu(\xi(\tau), t-\tau) d\tau \approx \sum_{i=0}^{n} \eta_i \nu(\xi^{(i)}, \tau_{n-i}) \delta t$$

Gaussian-weighted integral

$$\Phi_{\{\mu\}}[F] = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \prod_{k=1}^{n} dx_k \exp(-\mu_k x_k^2) F(x_1, \dots, x_n)$$

can be approximated by the Gauss-Hermite quadrature with weights $\{\lambda_k(i_k)\}_{i_k=1}^{l_k}$ and nodes $\{p_k(i_k)\}_{i_k=1}^{l_k}$ in each dimension $1 \leq k \leq n$. $p_k(i_k)$ – roots of Hermite polynomial $H_{l_k}(x)$, $i_k = 1, 2, \dots, l_k$ $\lambda_k(i_k) = \frac{2^{l_k - 1} l_k! \sqrt{\pi}}{l_k^2 \left[H_{l_k - 1}(x_{i_k})\right]^2}$

$$\Phi_{\{\mu\}}[F] \approx \sum_{i_1, i_2, \dots i_n} \lambda_1(i_1)\lambda_2(i_2) \dots \lambda_n(i_n) F(p_1(i_1), p_2(i_2), \dots, p_n(i_n)).$$

Summation is done over a multidimensional mesh, which is a Kronecker product of one-dimensional meshes $\mathbf{p}_1 \otimes \mathbf{p}_2 \otimes \dots \otimes \mathbf{p}_n$.

The idea of separation of variables

$$F(x_1, x_2, \dots, x_n) \approx \sum_{\alpha=1}^{r} u_1(x_1, \alpha) u_2(x_2, \alpha) \dots u_n(x_n, \alpha). \tag{10}$$

Minimal number r in sum is called *separation rank*.

The advantage of the decomposition (10) may be clearly seen from the following equality

$$\Phi_{\{\mu\}}[f] \approx \sum_{\alpha=1}^{r} \int_{-\infty}^{\infty} u_1(x_1, \alpha) \exp(-\mu_1 x_1^2)\, dx_1 \dots \int_{-\infty}^{\infty} u(x_n, \alpha) \exp(-\mu_n x_n^2)\, dx_n$$

In case of $r \ll n$ the dimensionality of the problem is significantly reduced.

In discrete case, when the function is defined on the multidimensional mesh indexed by i_k, $1 \leq k \leq n$, the separated representation turns out to be *canonical decomposition* of tensor **F**

$$F(i_1, i_2, \dots, i_n) \approx \sum_{\alpha=1}^{r} U_1(i_1, \alpha) U_2(i_2, \alpha) \dots U_n(i_n, \alpha),$$

the number r is called *canonical rank*, $U_k(i_k, \alpha)$ are real or complex matrices of size $I_k \times r$.

The advantage: complexity of the problem is reduced from $\mathcal{O}(n^d)$ to $\mathcal{O}(rnd)$, where n is the mesh size, d - dimension, r – rank.

A tensor **F** is sad to be in the TT-format if its elements satisfy the relation

$$F(i_1, i_2, \ldots, i_n) = G_1(i_1) G_2(i_2) \ldots G_n(i_n),$$

$G_k(i_k)$ are $r_{k-1} \times r_k$ matrices for each fixed i_k, $r_0 = r_n = 1$. The numbers r_k are called *TT-ranks* The parameter-depended matrices $G_k(i_k)$ can be considered as three-dimensional tensors $G_k(\alpha_{k-1}, i_k, \alpha_k)$, where α_k vary from 1 to r_k.
For a given tensor and fixed ranks r_k there exists the best possible approximation in the TT-format.

$$\Phi_{\{\mu\}}[F] \approx S_1 S_2 \ldots S_n,$$

$$S_k = \sum_{i_k=1}^{I_k} G_k(i_k),$$

Python toolbox `ttpy`: `tt.cross()`, `tt.sum()`, `tt.multifuncrs()`, etc.
`https://github.com/oseledets/ttpy`
I.V. Oseledets, T. Saluev, S. Dolgov, and D. Savostyanov.

Test example:

$$V(x,t) = -\frac{1}{t+1} - \frac{1}{x^2+1} + \frac{4x^2}{(x^2+1)^2},$$

$$f(x) = \frac{1}{\pi}\frac{1}{x^2+1},$$

with exact solution

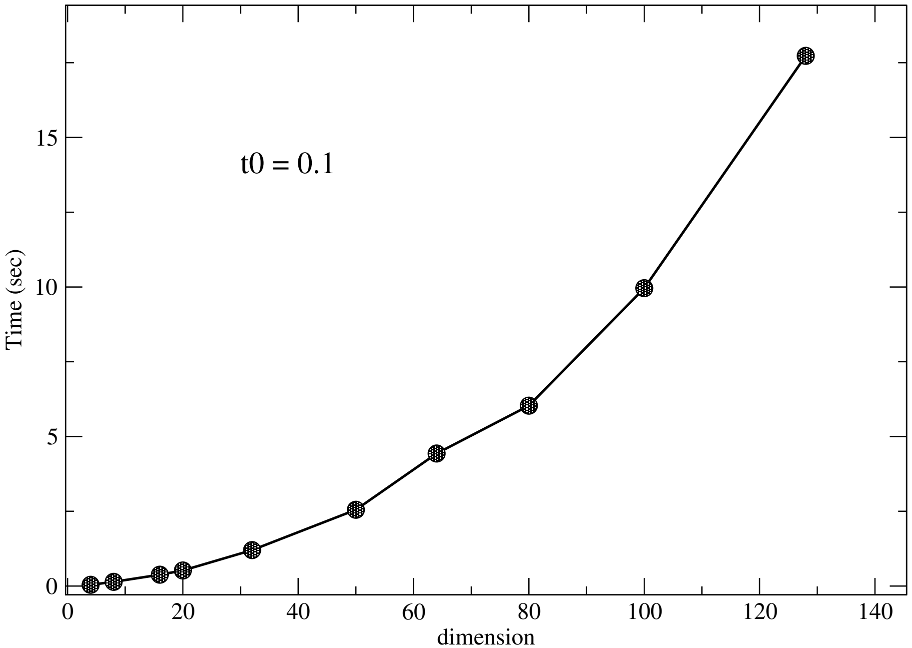
$$u_v(x,t) = \frac{1}{\pi}\frac{t+1}{x^2+1}.$$

Table : Path integral for $t_0 = 0.1$, $x_0 = 2.1$, cross accuracy $\varepsilon = 10^{-6}$, number of the mesh nodes $m = 10$

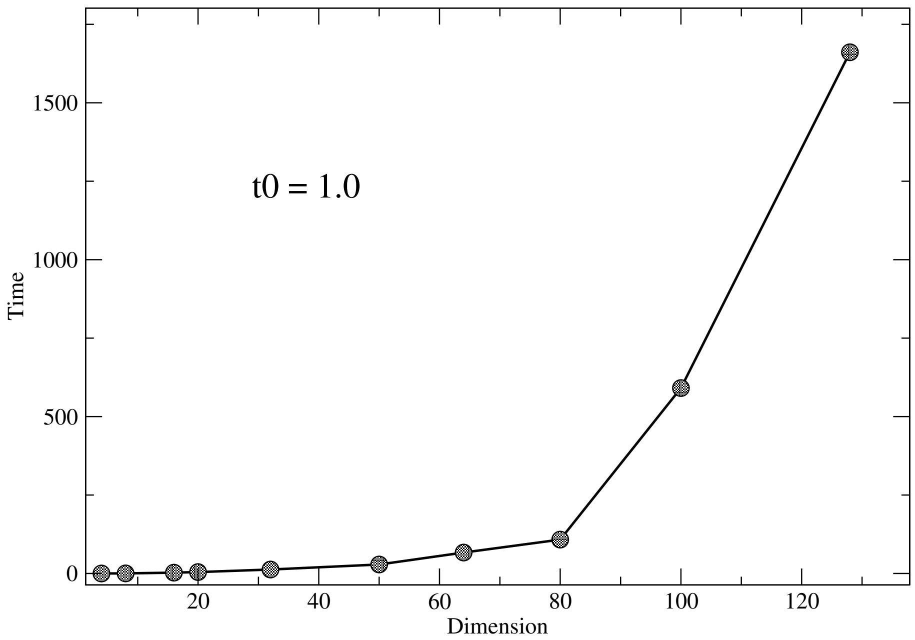
n	δt	Value	Rel. Error	T (sec.)	N calls
		0.06472105^{ex}			
128	$7.81 \cdot 10^{-4}$	0.06470589	$2.34 \cdot 10^{-4}$	17.73	1 100 600
100	$1.00 \cdot 10^{-3}$	0.06470165	$3.00 \cdot 10^{-4}$	9.96	884 740
80	$1.25 \cdot 10^{-3}$	0.06469680	$3.75 \cdot 10^{-4}$	6.03	701 150
64	$1.56 \cdot 10^{-3}$	0.06469074	$4.68 \cdot 10^{-4}$	4.43	571 660
50	$2.00 \cdot 10^{-3}$	0.06468227	$5.99 \cdot 10^{-4}$	2.55	440 480
32	$3.13 \cdot 10^{-3}$	0.06466049	$9.36 \cdot 10^{-4}$	1.20	273 680
20	$5.00 \cdot 10^{-3}$	0.06462425	$1.50 \cdot 10^{-3}$	0.52	171 510
16	$6.25 \cdot 10^{-3}$	0.06460014	$1.87 \cdot 10^{-3}$	0.38	130 520
8	$1.25 \cdot 10^{-2}$	0.06448006	$3.72 \cdot 10^{-3}$	0.14	49 510
4	$2.50 \cdot 10^{-2}$	0.06424237	$7.40 \cdot 10^{-3}$	0.04	11 930

Table : Path integral for $t_0 = 1.0$, $x_0 = 2.1$, cross accuracy $\varepsilon = 10^{-6}$, number of the mesh nodes $m = 10$

n	δt	Value	Rel. Error	T (sec.)	N calls
		0.11767463^{ex}			
128	$7.81 \cdot 10^{-3}$	0.11752589	$1.26 \cdot 10^{-3}$	1661	66 403 870
100	$1.00 \cdot 10^{-2}$	0.11748495	$1.61 \cdot 10^{-3}$	591	38 318 670
80	$1.25 \cdot 10^{-2}$	0.11743854	$2.01 \cdot 10^{-3}$	108	10 909 020
64	$1.56 \cdot 10^{-2}$	0.11738108	$2.49 \cdot 10^{-3}$	66.9	8 251 230
50	$2.00 \cdot 10^{-2}$	0.11730169	$3.17 \cdot 10^{-3}$	28.6	5 000 380
32	$3.13 \cdot 10^{-2}$	0.11710311	$4.86 \cdot 10^{-3}$	12.6	3 057 670
20	$5.00 \cdot 10^{-2}$	0.11678969	$7.52 \cdot 10^{-3}$	4.54	1 695 880
16	$6.25 \cdot 10^{-2}$	0.11659266	$9.19 \cdot 10^{-3}$	2.62	1 226 620
8	$1.25 \cdot 10^{-1}$	0.11573941	$1.64 \cdot 10^{-2}$	0.38	219 120
4	$2.50 \cdot 10^{-1}$	0.11451109	$2.69 \cdot 10^{-2}$	0.14	33 770



t0 = 0.1
Time (sec)
dimension
0
5
10
15
0
20
40
60
80
100
120
140


t0 = 1.0
Time
Dimension
0
500
1000
1500
20
40
60
80
100
120

Iteration of initial distribution

– Time evolution step Δt is fixed

$$u_{k+1}^{(n)}(x) = u^{(n)}(x, \Delta t) = \int_{-\infty}^{\infty} \mathcal{D}_{\xi}^{(n)}\, g_k\left(\xi^{(n)}\right) \prod_{i=0}^{n} \exp\left(-\eta_i \nu(\xi^{(i)}, \tau_{n-i})\delta t\right),$$

$$g_k(x) = u_k^{(n)}(x), \qquad g_0(x) = f(x)$$

1) linear growing of the definition domain $\sim n \max(|\sum_k \xi_k|)$
2)The error grows up proportional to the number of iterations $\varepsilon \cdot n$

Thank you for your attention!