# PROBABILITY LOADING VIA INVERSE TRANSFORMATION METHOD

DEKUAN DONG

## Contents



## 1. Background

The work in [15] presented a quantum algorithm for the Monte Carlo pricing of financial derivatives. The quantum amplitude estimation algorithm was applied to achieve a quadratic quantum speedup compared with classical Monte Carlo method.

### 1.1. Monte Carlo pricing.

Options are usually nonlinear functions applied to the outcomes of one or multiple underlying assets. The option payoff depends on the asset prices at specific time instances or is based on the paths of the asset prices.

Under the Black-Scholes-Merton (BSM) model, if the nonlinear function is piece-wise linear, the price can be priced analytically. If there are multiple independent Brownian processes underlying the dynamics, the price can often also be determined analytically. **The need for Monte Carlo arises if the payoff function is nonlinear beyond piece-wise linear or, for example, in cases when different asset prices are assume to be correlated.** Another class of options, called American options are also priced using Monte Carlo methods. Asian options depend on the average asset price during a time intervals and, if the averaging is arithmetic, may also require Monte Carlo.

Monte Carlo pricing of financial derivatives proceeds in the following way. Assume that the risk-neutral probability distribution is known, or can be obtained from calibrating to market variables. Sample from this risk-neutral probability distribution a market outcome, compute

*Date*: May 26, 2024.

the asset prices given that market outcome, then compute the payoff given the asset prices. Averaging the payoff over multiple samples obtains an approximation of the derivative price.

Assume a European option on a single benchmark asset and let the true option price be $\Pi$ and $\widehat{\Pi}$ be the approximation obtained from $k$ sample. Assume that the random variable of the payoff $f(S_T)$ is bounded in variance, i.e. $\mathrm{Var}[f(S_T)] \leq \lambda^2$. Then the probability that the price estimation $\widehat{\Pi}$ is $\epsilon$ away from the true price is determined by Chebyshev's inequality.

$$\Pr\left[|\widehat{\Pi} - \Pi| \geq \epsilon\right] \leq \frac{\lambda^2}{k\epsilon^2}.$$

For a constant success probability, we thus require

$$k = \mathcal{O}\left(\frac{\lambda^2}{\epsilon^2}\right)$$

samples to estimate to additive error $\epsilon$. The task of the quantum algorithm will be to improve the $\epsilon$ dependence from $\epsilon^2$ to $\epsilon$, hence providing a quadratic speedup for a given error.

The quantum Monte Carlo Integration based on the following theorem:

**Theorem 1.1** (Amplitude Estimation [12])**.** *There is a quantum algorithm called amplitude estimation which takes as input: one copy of a quantum state $|\chi\rangle$, a unitary transformation $\mathcal{U} = \mathcal{I} - 2|\chi\rangle\langle\chi|$, a unitary transformation $\mathcal{V} = \mathcal{I} - 2P$ for some projector $P$, and an integer $t$. The algorithm outputs $\widehat{a}$, an estimation of $a = \langle\chi|P|\chi\rangle$, such that*

$$|\widehat{a} - a| \leq 2\pi\frac{\sqrt{a(1-a)}}{t} + \frac{\pi^2}{t^2}$$

*with probability at least $8/\pi^2$, using $\mathcal{U}$ and $\mathcal{V}$ $t$ times each.*

1.1.1. *European Options.* For the BSM model with a single Brownian motion we have

$$\begin{aligned}\Pi &= e^{-rT}\mathbb{E}_{\mathbb{Q}}\left[v(W_T)\right],\\ v(x) &= \max\left\{0, S_0 e^{\sigma x + (r - \frac{1}{2}\sigma^2)T} - K\right\},\end{aligned}$$

where $\mathbb{Q}$ denotes the risk-neutral measure and $W_T$ is the Brownian motion at time $T$, and thus $W_T \sim \mathcal{N}(0, T)$. The probability density for this random variable is given by

$$p_T(x) = \frac{1}{\sqrt{2\pi T}} e^{-\frac{x^2}{2T}}.$$

Following the scheme of QMC proposed in [12], we need to prepare an approximate superposition of these probabilities. At first, we choose an $x_{\max} = \mathcal{O}(\sqrt{T})$ and discretise the density $p_T(x)$ on the interval $[-x_{max}, x_{max}]$, as a few standard deviations are usually enough to capture the normal distribution and reliably estimate the option price, and define

$$\Delta x = \frac{2x_{max}}{2^n - 1}, \quad x_j = -x_{max} + j \cdot \Delta x, \quad j = 0, 1, \ldots, 2^n - 1.$$

Define the probability $p_j = \frac{1}{C} p_T(x_j)$, with normalization $C = \sum_{j=0}^{2^n-1} p_T(x_j)$. Suppose we can efficiently prepare the quantum state

$$\sum_{j=0}^{2^n-1} \sqrt{p_j}|j\rangle. \tag{1.1}$$

Use the fact that $\mathrm{Var}_{\mathbb{Q}}[v(W_T)]$ can be bounded by some $\lambda^2 = \mathcal{O}\left(\mathrm{poly}(S_0, e^{rT}, e^{\sigma^2 T}, K)\right)$, we can use the algorithm proposed in [12] to compute the quantity

$$\sum_{j=0}^{2^n-1} p_j v(x_j),$$

which approximates the value of $\mathbb{E}_{\mathbb{Q}}[v(W_T)]$. Here, the question is to prepare the quantum state (1.1).

1.1.2. *Asian Option.*

**Definition 1.2.** *The Asian call option payoff is defined as*

$$f(A_T) = \max\{0, A_T - K\},$$

*where $K$ is the strike price and $T$ the maturity date. The arithmetic mean option value is defined via*

$$A_T^{arith} = \frac{1}{L}\sum_{l=1}^{L} S_{t_l},$$

*and the geometric mean option is defined via*

$$A_T^{geo} = \exp\frac{1}{L}\sum_{l=1}^{L} \log S_{t_l},$$

*for pre-defined time points $0 < t_1 < \cdots < t_L \leq T$. with $L \geq 1$.*

In the BSM framework, the geometric mean Asian option can be priced analytically, while such a solution is not known for the arithmetic Asian option. Assume for this discussion that all adjacent time points are separated by the interval $\Delta t$, i.e. $t_{l+1} - t_l = \Delta t = T/L$ for all $l = 1, \ldots, L-1$. Analogously to the European case, we can efficiently prepare via the Grover-Rudolph algorithm a state that corresponds to the Gaussian normal distribution with variance $\Delta t$

$$|p_{\Delta t}\rangle := \mathcal{G}|0^m\rangle = \sum_{j=0}^{2^m-1} \sqrt{p_{\Delta t}(x_j)}|j\rangle. \tag{1.2}$$

This state uses $m$ qubits and takes $\mathcal{O}(m)$ steps to prepare. Then we prepare the product state with $L$ such states, i.e.,

$$|p\rangle := |p_{\Delta t}\rangle \cdots |p_{\Delta t}\rangle.$$

This state uses $Lm$ qubits and takes $\mathcal{O}(Lm)$ steps to prepare.

In addition, we require the operation

$$|j_1, \ldots, j_L\rangle|0\rangle = |j_1, \ldots, j_L\rangle|A(S_{t_1}(x_{j_1}), \ldots, S_{t_L}(x_{j_L}))\rangle. \tag{1.3}$$

Here, $A(S_{t_1}(x_{j_1}), \ldots, S_{t_L}(x_{j_L}))$ is the average stock price corresponding to the Brownian path $x_{j_1}, \ldots, x_{j_L}$. This operation is easily computable. Each index $j$ is mapped to its corresponding point $x_j$ via $x_j = -x_{max} + j\Delta x$ as before. Then start at the known $S_0$ and use

$$S_{t_{l+1}}(x) = S_{t_l} \exp\left(\sigma x + \left(r - \frac{\sigma^2}{2}\right)\Delta t\right)$$

to obtain the stock price at the next time point. Moreover, the average (both arithmetic and geometric) can be computed in a sequential manner, since we can implement the step

$$\begin{aligned}&|j_1,\dots,j_L\rangle|S_{t_l}(x_{j_l})\rangle|A(S_{t_1}(x_{j_1}),\dots,S_{t_l}(x_{j_l}))\rangle\\ \longrightarrow&|j_1,\dots,j_L\rangle|S_{t_{l+1}}(x_{j_{l+1}})\rangle|A(S_{t_1}(x_{j_1}),\dots,S_{t_{l+1}}(x_{j_{l+1}}))\rangle.\end{aligned}$$

The steps are performed until the final time $t_L$ is reached and $A(S_{t_1}(x_{j_1}),\dots,S_{t_L}(x_L))$ is stored in a register of qubits. Reversibility of the quantum arithmetic operations guarantees that registers storing the intermediate steps can be uncomputed.

Applying operation (1.3) to the product state (1.2) obtains

$$\sum_{j_1,\dots,j_L=0}^{2^m-1}\sqrt{p_{j_1,\dots,j_L}}\rangle|A(S_{t_1}(x_{j_1}),\dots,S_{t_L}(x_{j_L}))\rangle,$$

with $\sqrt{p_{j_1,\dots,j_L}}:=\sqrt{p_{\Delta t}(x_{j_1})}\cdots\sqrt{p_{\Delta t}(x_{j_L})}$. Thus, we can use a standard QMC procedure to estimate the price of the Asian option.

1.2. **Loading probabilities in the Superposition.** To perform Quantum Monte Carlo, it is desired to create a quantum superposition of states which encode a discrete approximation of corresponding probability density functions. The most cited method to create such a superposition state is proposed by Grover and Rudolph [3] in 2002. This paper focused on the log-concave probability density functions, which are efficiently integrable by classical monte carlo methods. The Grover-Rudolph method takes $\Theta(n)$ steps to prepare a discrete distribution with $N=2^n$ points, and thus is efficient in this sense (with the important caveat that $p(x)$ need to be efficiently integrable).

However, in [5], Herbert showed that the Grover-Rudolph state preparation is not sufficiently efficient to preserve the quadratic quantum advantage in quantum Monte-Carlo. This result gives a motivation to find novel method to loading probability [16, 7].

1.2.1. *Grover-Rudolph Method.* Consider a probability density $p(x)$ of a single variable $x$. Assume we have discretized the probability over some interval, such that for some integer $n$, we have $\{p_j\}$ for $j=0,\dots,2^n-1$. Assume that $\sum_j p_j=1$. The task is to show an algorithm $\mathcal{G}$ without measurements such that

$$\mathcal{G}|0^n\rangle=\sum_{j=0}^{2^n-1}\sqrt{p_j}|j\rangle.$$

Further assume that there exists a shallow classical circuit that can efficiently compute the sums

$$\sum_{j=a}^{b}p_j\approx\int_{x_a}^{x_b}p(x)dx,$$

for any $a\le b\in\{0,\cdots,2^n-1\}$. Thus, for $m=1,\dots,n$ we can efficiently compute the probabilities

$$p_k^{(m)}=\sum_{j=k2^{n-m}}^{(k+1)2^{n-m}-1}p_j,$$

with $k=0,\dots,2^m-1$.

For $m < n$, assume we have prepared the state

$$|\psi^{(m)}\rangle = \sum_{k=0}^{2^m-1} \sqrt{p_k^{(m)}}|k\rangle.$$

We would like to show by induction that we can prepare the state

$$|\psi^{(m+1)}\rangle = \sum_{k=0}^{2^{m+1}-1} \sqrt{p_k^{(m+1)}}|k\rangle.$$

Define the quantities

$$f(k,m) = \frac{p_{2k}^{(m+1)}}{p_k^{(m)}}.$$

This quantity allows to go up one level of discretization to $m+1$. Also define $\theta_k^{(m)} = \arccos\sqrt{f(k,m)}$. The operation

$$|k\rangle|0\rangle \mapsto |k\rangle|\theta_k^{(m)}\rangle$$

is enabled by the efficient computability of $f(k,m)$. Now proceed

$$\begin{aligned}
|\psi^{(m)}\rangle|0\rangle|0\rangle &\mapsto \sum_{k=0}^{2^m-1} \sqrt{p_k^{(m)}}|k\rangle|\theta_k^{(m)}\rangle|0\rangle \\
&\mapsto \sum_{k=0}^{2^m-1} \sqrt{p_k^{(m)}}|k\rangle|\theta_k^{(m)}\rangle \left(\cos\theta_k^{(m)}|0\rangle + \sin\theta_k^{(m)}|1\rangle\right) \\
&\equiv |\psi^{(m+1)}\rangle.
\end{aligned}$$

In the second step the register containing $|\theta_k^{(m)}\rangle$ was uncomputed.

1.2.2. *Problem with G-R method for QMI.* For simplicity, we assume that the distribution of interest has been discretised over $N = 2^n$ points and shifted and scaled such that its domain is $\{1, 2, \ldots, 2^n - 1\}$. We let $\mu$ be the mean of $p(x)$ and $\widehat{\mu}$ be an estimator of $\mu$. The natural measure of error for Monte-Carlo integration is the RMSE, defined:

$$\widehat{\epsilon} = \sqrt{\mathbb{E}\left[(\widehat{\mu} - \mu)^2\right]}.$$

**Theorem 1.3** ([5])**.** *To achieve a RMSE of $\widehat{\epsilon}$ using an unbiased quantum Monte-Carlo estimation method requires $\widetilde{\Omega}\left(\frac{1}{\widehat{\epsilon}^2}\right)$ operations when the Grover-Rudolph method is used to prepare some log-concave distribution as a quantum state.*

*Proof.* To assess the convergence rate of quantum Monte-Carlo, we make the conservative assumption that error is only incurred in the first Grover-Rudolph iteration. Formally, let

$$q_l = \sum_{i=0}^{2^{n-1}-1} p_i$$

be the probability mass that is distributed over the left-hand half of the distribution, and $\epsilon_l$ be the error incurred in the first Grover-Rudolph iteration. Under these assumptions, rather

than preparing state $\sum_{i=0}^{2^n-1}\sqrt{p_i}|i\rangle$, we instead prepare the erroneous state:

$$|\psi'\rangle = \sum_{i=0}^{2^{n-1}-1}\sqrt{\frac{p_i(q_l+\epsilon_l)}{q_l}}|i\rangle + \sum_{i=2^{n-1}}^{2^n-1}\sqrt{\frac{p_i(1-q_l-\epsilon_l)}{1-q_l}}|i\rangle.$$

We can now evaluate the mean, $\mu'$, of the erroneous distribution that has been encoded in the state $|\psi'\rangle$:

$$\begin{aligned}\mu' &= \sum_{i=0}^{2^{n-1}-1}\frac{p_i(q_l+\epsilon_l)}{q_l}i + \sum_{i=2^{n-1}}^{2^n-1}\frac{p_i(1-q_l-\epsilon_l)}{1-q_l}i \\ &= \sum_{i=0}^{2^n-1} ip_i + \epsilon_l\left(\sum_{i=0}^{2^{n-1}-1}\frac{p_i}{q_l}i - \sum_{i=2^{n-1}}^{2^n-1}\frac{p_i}{1-q_l}i\right) \\ &= \mu - k\epsilon_l,\end{aligned}$$

where $k$ is a non-zero constant that depends only on the distribution being encoded and not the sampling process. It follows that when performing the quantum Monte-Carlo, we are actually sampling from an erroneous distribution whose mean errs from the mean of the actual distribution of interest by a factor proportional to $\epsilon_l$. Thus, we can consider the overall mean squared error, which is equal to:

$$\begin{aligned}\widehat{\epsilon}^2 &= \mathbb{E}\left[(\mu - \widehat{\mu}(N_q))^2\right] \\ &= \mathbb{E}\left[(\mu' + k\epsilon_l - \widehat{\mu}(N_q))^2\right] \\ &= \mathbb{E}\left[(\mu' - \widehat{\mu}(N_q))^2\right] + k^2\mathbb{E}[\epsilon_l^2] + 2k\mathbb{E}\left[\epsilon_l\mathbb{E}\left[\mu' - \widehat{\mu}(N_q)|\epsilon_l\right]\right] \\ &= \mathbb{E}\left[(\mu' - \widehat{\mu}(N_q))^2\right] + k^2\mathbb{E}[\epsilon_l^2],\end{aligned}$$

where $\widehat{\mu}(N_q)$ is the estimate of the mean obtained by performing quantum Monte-Carlo such that the state preparation circuit is queries $N_q$ times. The analysis of QAE gives $\mathbb{E}\left[(\mu' - \widehat{\mu}(N_q))^2\right] = \Theta(1/N_q^2)$. However, the error $\epsilon_l$ is due to the classical Monte Carlo method, and thus

$$\mathbb{E}[\epsilon_l^2] = \frac{q_l(1-q_l)}{N_c},$$

where $N_c$ is the number of samples of classical Monte Carlo method. Finally, we can see that

$$\widehat{\epsilon}^2 = \Theta\left(\frac{1}{N_q^2} + \frac{1}{N_c}\right).$$

We can see that the classical Monte-Carlo sampling required to prepare the state sufficiently accurately is therefore the bottlenech, thus giving us an overall computational complexity of

$$\widetilde{\Omega}\left(\frac{1}{\widehat{\epsilon}^2}\right)$$

where $\Omega$ is used because this is a lower-bound: we have only considered only the error in the first Grover-Rudolph iteration. □

## 2. Some Novel Approach to Load Distribution for QMCI

In the above section, we see that the Grover-Rudolph method is not efficient enough for Quantum Monte Carlo Integration. Here we find some novel method to load the probability more efficient.

2.1. **Efficient State Preparation for QAE.** In [16], they discussed QAE from a numerical integration point of view. Here is their method: suppose a random variable $X$ with a corresponding probability density function $f : \mathbb{R} \mapsto \mathbb{R}_{\geq 0}$, an we are going to approximate the expectation $\mathbb{E}_f[g(X)] = \int g(x)f(x)dx$. To perform computation in quantum state's amplitude, we assume the range of $f$ and $g$ are normalized to $[0, 1]$, and thus we can prepare operators $\mathcal{F}$ and $\mathcal{G}$ such that

$$\mathcal{F} : |i\rangle_n|0\rangle|j\rangle \mapsto |i\rangle_n \left(\sqrt{1 - f(x_i)}|0\rangle + \sqrt{f(x_i)}|1\rangle\right)|j\rangle,$$
$$\mathcal{G} : |i\rangle_n|j\rangle|0\rangle \mapsto |i\rangle_n|j\rangle \left(\sqrt{1 - g(x_i)}|0\rangle + \sqrt{g(x_i)}|1\rangle\right),$$

where $\mathcal{F}$ prepares the first ancilla qubit and $\mathcal{G}$ prepares the second one, and $x_i = a \cdot i + b$ is a set of grids in $\mathbb{R}$. We now apply Hadamard gates to the first $n$ qubits of $|0\rangle_n|00\rangle$ followed by $\mathcal{F}$ and $\mathcal{G}$, which leads to the state

$$\frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle_n \sqrt{1 - f(x_i)}\sqrt{1 - g(x_i)}|00\rangle + \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle_n \sqrt{1 - f(x_i)}\sqrt{g(x_i)}|01\rangle$$
$$+\frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle_n \sqrt{f(x_i)}\sqrt{1 - g(x_i)}|10\rangle + \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle_n \sqrt{f(x_i)}\sqrt{g(x_i)}|11\rangle.$$

Subsequently, the probability of measuring $|11\rangle$ for the last two qubits is given by

$$\frac{1}{2^n} \sum_{i=0}^{2^n-1} f(x_i)g(x_i),$$

which is equal to a Riemann sum and approximates the expected value $\mathbb{E}_f[g(X)]$.

It is straightforward to extend this approach to multivariate problems as well. Suppose a stochastic process given by the initial probability distribution $f_0(x^0)$ and the transition probabilities $f_t(x^t|x^{t-1})$, i.e., the probability to reach state $x^t$ at time $t$ given the state $x^{t-1}$ at time $t - 1$. To simplify the notation, we assume Markov processes. Furthermore, assume $n_t$ qubits to represent the state $x^t$ at time $t$, for $t = 0, \ldots, T$, add $T + 1$ ancilla qubits. Then, as before, we can construct an operator $\mathcal{F}_0$ corresponding to $f_0$ to prepare the first ancilla qubit, and we can construct the operators

$$\mathcal{F}_t : |i\rangle_{n_{t-1}}|j\rangle_{n_t}|0\rangle \mapsto |i\rangle_{n_{t-1}}|j\rangle_{n_t} \left(\sqrt{1 - f(x_j^t|x_i^{t-1})}|0\rangle + \sqrt{f(x_j^t|x_i^{t-1})}|1\rangle\right).$$

If we first apply Hadamard gates to all state qubits and then apply $\mathcal{F}_0, \ldots, \mathcal{F}_T$ to the corresponding qubit registers and ancilla qubits, we can construct the $(n + T + 1)$-qubit state

$$\frac{1}{\sqrt{2^n}} \sum_{i_0=0}^{2^{n_0}-1} \cdots \sum_{i_T=0}^{2^{n_T}-1} |i_0\rangle_{n_0} \cdots |i_T\rangle_{n_T} \left[\cdots + \sqrt{f_0(x_{i_0}^0) \prod_{t=1}^{T} f(x_{i_t}^t|x_{i_{t-1}}^{t-1})}|1 \cdots 1\rangle_{T+1}\right],$$

where we drop the terms without $|1 \cdots 1\rangle_{T+1}$ in the ancilla qubits.

Given an objective function $g : \mathbb{R}^{T+1} \mapsto [0,1]$ and the corresponding operator $\mathcal{G}$, we can add another ancilla qubit and apply $\mathcal{G}$. The resulting probability of measuring $|1\cdots 1\rangle_{T+2}$ in all ancilla qubits is then given by

$$\frac{1}{2^n}\sum_{i_0,\ldots,i_T} f_0(x_{i_0}^0)\prod_{t=1}^{T} f(x_{i_t}^t|x_{i_{t-1}}^{t-1})g\left(x_{i_0}^0,\ldots,x_{i_T}^T\right),$$

which approximates the expectation value $\mathbb{E}[g(X)]$ where $X = \left(x^0,\ldots,x^T\right)$ represents the possible paths of the stochastic process defined by the $f_t$.

2.1.1. *Error Analysis and Limitations.* The error comes from the truncation and discretization of the original integral to a Riemann sum, the quantum arithmetic and the quantum amplitude estimation. In addition, since we have to scale the density function $f$, transition density function $f_t(\cdot|\cdot)$, and the payoff function $g$ to the range $[0,1]$, the result we get is proportional to the desired result. For instance, we want to evaluate the Riemann sum

$$\frac{1}{2^n}\sum_{i_0,\ldots,i_T} f_0(x_{i_0}^0)\prod_{t=1}^{T} f(x_{i_t}^t|x_{i_{t-1}}^{t-1})g\left(x_{i_0}^0,\ldots,x_{i_T}^T\right),$$

where the value of $f(\cdot|\cdot)$ is out of the range $[0,1]$, and for simplicity assume $f_0, g \in [0,1]$. We need to find a constant $C > 1$ such that

$$\widetilde{f}(\cdot|\cdot) = \frac{1}{C}f(\cdot|\cdot) \in [0,1],$$

thus we can use quantum algorithm to evaluate the modified summation

$$\frac{1}{2^n}\sum_{i_0,\ldots,i_T} f_0(x_{i_0}^0)\prod_{t=1}^{T} \widetilde{f}(x_{i_t}^t|x_{i_{t-1}}^{t-1})g\left(x_{i_0}^0,\ldots,x_{i_T}^T\right).$$

To retrieval the desired result, we need to multiply the modified summation a factor $C^T$. It means that the error becomes $C^T$ times larger, and $C^T$ grows exponentially with respect to $T$. This would limit our grid density when discretizing a given stochastic process.

2.2. **Clustering Method.** The main bottleneck of Grover-Rudolph method is that we have to perform different rotation angles for different sub-intervals. Ref. [10] propose a method to cluster the rotation angles and perform an uniform rotation angle to the quantum state.

Recall that the method of Grover and Rudolph, describes a constructive protocol to load into an $n$-qubit quantum state the discretized version $\{f_i\}$ of certain integrable density function $f : [x_{min}, x_{max}] \subset \mathbb{R} \mapsto \mathbb{R}_+$ as

$$|\Psi(f)\rangle_n = \sum_{i=0}^{2^n-1}\sqrt{f_i}|i\rangle.$$

Without using ancillas, this protocol provides a constructive algorithm which applies a sequence of operation blocks, $F_k^{(k-1)}(\boldsymbol{\theta}^{(k-1)})$, to the initial state $|0\rangle^{\otimes n}$, where

$$F_k^{(k-1)}(\boldsymbol{\theta}^{(k-1)}) = \sum_{l=0}^{2^{k-1}-1}|l\rangle\langle l|\otimes R_y\left(\theta_l^{(k-1)}\right),$$

and

$$\theta_l^{(k-1)} = 2\arccos\left(\sqrt{\frac{\int_{x_{min}+l\delta_k}^{x_{min}+(l+1/2)\delta_k} f(x)dx}{\int_{x_{min}+l\delta_k}^{x_{min}+(l+1)\delta_k} f(x)dx}}\right).$$

Here, $\delta_k = \frac{x_{max}-x_{min}}{2^{k-1}}$ and $l \in \left\{0,\dots,2^{k-1}-1\right\}$ is the index corresponding to the $(l+1)$-th subinterval of the $2^{k-1}$ partition of the interval $[x_{min}, x_{max}]$.

The key observation of [10] is the following lemma.

**Lemma 2.1** (Lemma 1 of [10]). *Let $f$ be a continuous function such that $f:[0,1]\mapsto\mathbb{R}_+$ and consider a block comprising a uniformly-controlled rotation of $k$ qubits. Then, the difference between two contiguous angles is bounded by the second derivative of the logarithm of $f$ in the following way:*

$$|\theta_{l+1}^{(k-1)} - \theta_l^{(k-1)}| \leq \frac{\theta_k^2}{4} \max_{y'\in[l\delta_k,(l+1)\delta_k]} \left|\left(\partial_y^2 \log f(y)\right)\Big|_{y=y'}\right|,$$

*where $\delta_k = \frac{1}{2^{k-1}}, l \in \left\{0,\dots,2^{k-1}-2\right\}$.*

Consequently, if $|\partial_y^2 \log f(y)| \leq \eta$, the for each block comprising a uniformly-controlled rotation of $k$ qubits, the difference between any two angles is bounded by

$$|\theta_l^{(k-1)} - \theta_{l'}^{(k-1)}| \leq \frac{\delta_k}{4}\eta, \quad \forall l,l' \in \left\{0,\dots,2^{k-1}-1\right\}.$$

This result allows us to cluster angles of each block in which the above condition is fulfilled. Thus, we define a cluster representative angle, $\widetilde{\theta}^{(k-1)}$, as

$$\left|\widetilde{\theta}^{(k-1)} - \theta_l^{(k-1)}\right| \leq \frac{\delta_k}{8}\eta := \eta_k, \quad \forall l \in \left\{0,\dots,2^{k-1}-1\right\}.$$

Then, the following lemma bound the error if we replace $\theta_l^{(k-1)}$'s by $\widetilde{\theta}^{(k-1)}$. Before the statement of the lemma, we need some notation. Let $\mathfrak{U}_n = \mathcal{U}_{n-1}\left(\boldsymbol{\theta}^{(n-1)}\right)\cdots\mathcal{U}_0\left(\boldsymbol{\theta}^{(0)}\right)$, where we define

$$\mathcal{U}_{k-1}\left(\boldsymbol{\theta}^{(k-1)}\right) := F_k^{(k-1)}\left(\boldsymbol{\theta}^{(k-1)}\right)\otimes\mathbb{I}^{\otimes(n-k)}.$$

Let $\widetilde{\mathfrak{U}}_n$ denote the operation $\mathfrak{U}_n$ when the rotation angles corresponding to each block $k$ are replaced by a representative, $\widetilde{\theta}^{(k-1)}$, such that its difference with any angle of the block is at most $\eta_k$, i.e. $|\theta_l^{(k-1)} - \widetilde{\theta}^{(k-1)}| \leq \eta_k$ for $l = 0,\dots,2^{k-1}-1$ and $k = 1,\dots,n$. Then the following lemma are proposed in [10].

**Lemma 2.2** (Lemma 2 of [10]). *Consider a system of $n$ qubits and an error $\eta_k$ between any angle of the $k$-th block and its representative such that $\eta_k \leq \pi$, with $k = 1,\dots,n$. Then, the fidelity between the final states with and without clustering, $F = \left|\langle 0|^{\otimes n}\mathfrak{U}_n^\dagger\widetilde{\mathfrak{U}}_n|0\rangle^{\otimes n}\right|^2$, satisfies*

$$F \geq \prod_{k=1}^{n}\cos^2(\eta_k/2).$$

Assuming that we cluster angles of the blocks for $k \geq k_0+1 > 2$, then

$$F \geq \prod_{k=k_0+1}^{n}\cos^2\left(\frac{\eta}{8\cdot 2^k}\right) \geq e^{-\frac{\eta^2}{96}(4^{-k_0}-4^{-n})} := F_{k_0},$$

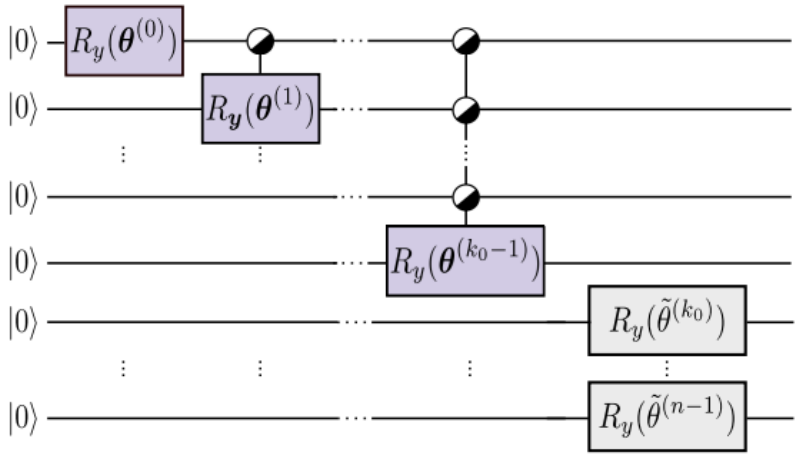
since $\cos(x) \geq e^{-x^2}$ for $x \lessapprox \pi/2$. Therefore, if an infidelity $\epsilon = 1 - F_{k_0}$ is allowed and the angles of all blocks comprised of more than $k_0(\epsilon)$ qubits are clustered, with $k_0$ given by

$$k_0(\epsilon) = \max\left\{\lceil -\frac{1}{2}\log_2\left(4^{-n} - \frac{96}{\eta^2}\log(1-\epsilon)\right)\rceil, 2\right\},$$

then the fidelity satisfies $F \geq 1 - \epsilon$. In the asymptotic limit of $n \to \infty$, we have that $k_0(\epsilon)$ tends to

$$\max\left\{\lceil -\frac{1}{2}\log_2\left(-\frac{96}{\eta^2}\log(1-\epsilon)\right)\rceil, 2\right\}$$

which is independent of the system size, $n$. The corresponding quantum circuit is given by




2.2.1. *Problems.* It seems that we can use a constant number of quantum gates to prepare an arbitrary number of qubits state, however, there is a question here.

> Does the rotations $R_y\left(\widetilde{\theta}^{(k)}\right)$ $(k > k_0)$ really make the quantum state a better discretization to the function $f$?

Unfortunately, the answer shall be no, at least for the loading of normal distribution. Suppose, we use the above method to load the square root of standard normal density, i.e.

$$f(x) = \sqrt{\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x^2}{2}\right)}.$$

Moreover, suppose that we discretize the function in the interval $[-w, w]$ by $2^n$ equally spaced grids. Then, after $k_0$ steps, we are going to perform rotations with clustered angles. Notice that $f$ is a even function, and thus if there is a $l$ such that

$$\theta_l^{(k-1)} = 2\arccos\left(\sqrt{\frac{\int_{-w+l\delta_k}^{-w+(l+1/2)\delta_k} f(x)dx}{\int_{-w+l\delta_k}^{-w+(l+1)\delta_k} f(x)dx}}\right) \equiv 2\arccos\left(\sqrt{\alpha}\right)$$

then there is another $l' = 2^{k-1} - l - 1$ such that

$$\theta_{l'}^{(k-1)} = 2\arccos\left(\sqrt{\frac{\int_{-w+l'\delta_k}^{-w+(l'+1/2)\delta_k} f(x)dx}{\int_{-w+l'\delta_k}^{-w+(l'+1)\delta_k} f(x)dx}}\right) = 2\arccos\left(\sqrt{\frac{\int_{w-(l+1/2)\delta_k}^{w-(l+1)\delta_k} f(x)dx}{\int_{w-(l+1)\delta_k}^{w-l\delta_k} f(x)dx}}\right)$$
$$= 2\arccos\left(\sqrt{\frac{\int_{-w+(l+1/2)\delta_k}^{-w+(l+1)\delta_k} f(x)dx}{\int_{-w+l\delta_k}^{-w+(l+1)\delta_k} f(x)dx}}\right) = 2\arccos\left(\sqrt{1-\alpha^2}\right).$$

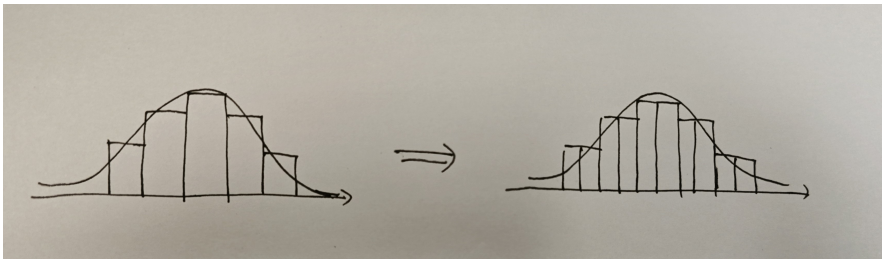
We can see that

$$\frac{\theta_l^{(k-1)} + \theta_{l'}^{(k-1)}}{2} = \frac{\pi}{2},$$

and thus $\theta_l^{(k-1)}$'s are symmetrically distributed about $\pi/2$, which means that we can choose $\widetilde{\theta}^{(k-1)} = \pi/2$ as the representative. However,

$$R_y(\pi/2) = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}.$$

The effect of $R_y(\pi/2)$ can be illustrated by the following figure:



It is easy to show that the denser grids does not give a better approximate to $f$.

2.3. **Other Methods.** In this subsection, I list some other method to achieve the probability loading task:

- QFT (Quantum Eigenvalue Transformation) based method [11];
- MPS (Matrix Product State) based method [7, 6];
- Variational method [2]. Several materials [9] suggest these methods have difficulties in avoiding local minima;
- Random walk method [14]. The main problem of this method is that it is not a reversible process, but its idea is interesting.

2.4. **What is a good initialization method?** Ref.[9] summarizes several properties that a good initialization method should have:

- being efficient, i.e., scaling polynomially in the number of qubits;
- being robust to noise;
- having controllable precision;
- being deterministic, i.e., not having any post-selection;
- having efficient circuit construction, i.e. that the compilation of the circuit is efficient.

Moreover, in the same reference [9], they summarized eight categories of initialization method but none of them fulfill all the requirements at once.

## 3. My Ideas

3.1. **Inverse Method to Prepare Distribution.** Suppose we want prepare a quantum state to approximately encode a distribution with cumulative function $F$. Inspired by the inverse transformation method, we may use the following steps to prepare such a quantum state.

$$|0^n\rangle|0^m\rangle \mapsto \sum_{x\in\{0,1\}^n} \frac{1}{\sqrt{2^n}}|x\rangle|0^m\rangle \mapsto \sum_{x\in\{0,1\}^n} \frac{1}{\sqrt{2^n}}|x\rangle|\widehat{F}_b^{-1}(x)\rangle,$$

where $\widehat{F}_b^{-1}:\{0,1\}^n \mapsto \{0,1\}^m$ is an proxy of $F^{-1}$ in the sense that

$$\begin{cases} -\frac{1}{2^{d+1}} < F^{-1}\left(\frac{x}{2^n}\right) - \left(\frac{\widehat{F}_b^{-1}(x)}{2^d} - M\right) \le \frac{1}{2^{d+1}}, \quad \forall x = 1,\dots,2^n-1, \\ \widehat{F}_b^{-1}(0) = 0. \end{cases} \tag{3.1}$$

Then, we rearrange the final state in terms of a summation about the second register and get

$$\sum_{x\in\{0,1\}^n} \frac{1}{\sqrt{2^n}}|x\rangle|\widehat{F}_b^{-1}(x)\rangle = \sum_{y\in\{0,1\}^m} \left(\sum_{x:\widehat{F}_b^{-1}(x)=y} \frac{1}{\sqrt{2^n}}|x\rangle\right)|y\rangle. \tag{3.2}$$

Note that

$$\begin{aligned} \left\{x\in\{0,1\}^n : \widehat{F}_b^{-1}(x) = y\right\} &= \left\{x\in\{0,1\}^n : -\frac{1}{2^{d+1}} < F^{-1}\left(\frac{x}{2^n}\right) - \left(\frac{y}{2^d} - M\right) \le \frac{1}{2^{d+1}}\right\} \\ &= \left\{x\in\{0,1\}^n : \frac{y-2^{-1}}{2^d} - M < F^{-1}\left(\frac{x}{2^n}\right) \le \frac{y+2^{-1}}{2^d} - M\right\} \\ &= \left\{x\in\{0,1\}^n : F\left(\frac{y-2^{-1}}{2^d} - M\right) < \frac{x}{2^n} \le F\left(\frac{y+2^{-1}}{2^d} - M\right)\right\}, \end{aligned}$$

where the last equality due to the definition that $F^{-1}(u) = \inf\{x : F(x) \ge u\}$, and thus $\{F^{-1}(u) \le y\} = \{u \le F(y)\}$. Let $N_y$ denote the number of elements in this set, then we have

$$\frac{F\left(\frac{y+2^{-1}}{2^d} - M\right) - F\left(\frac{y-2^{-1}}{2^d} - M\right)}{\frac{1}{2^n}} - 2 < N_y < \frac{F\left(\frac{y+2^{-1}}{2^d} - M\right) - F\left(\frac{y-2^{-1}}{2^d} - M\right)}{\frac{1}{2^n}}. \tag{3.3}$$

When we measure the second register with computational basis, the probability to get an outcome $y$ is

$$\frac{N_y}{2^n} \in \left(\widetilde{p}_y - 2^{-n+1}, \widetilde{p}_y\right),$$

where

$$\begin{aligned} \widetilde{p}_y &= F\left(\frac{y+2^{-1}}{2^d} - M\right) - F\left(\frac{y-2^{-1}}{2^d} - M\right) \\ &\equiv F\left(\overline{y}\right) - F\left(\underline{y}\right). \end{aligned}$$

Suppose $Y \sim F$, then we have

$$\widetilde{p}_y = \Pr\left[Y \in (\underline{y}, \overline{y}]\right] = \Pr\left[-2^{-(d+1)} < Y - B_y \le 2^{-(d+1)}\right],$$

where $B_y = \frac{y}{2^d} - M$. We can see that $\{B_y : y \in \{0,1\}^m\}$ is a set of points distributed in the interval $[-M, 2^{m-d} - M - 2^{-d}]$ with equal space.

**Question 3.1.** *We can write the (3.2) as*

$$\sum_{y\in\{0,1\}^m}\left(\sum_{x:\widehat{F}_b^{-1}(x)=y}\frac{1}{\sqrt{2^n}}|x\rangle\right)|y\rangle \equiv \sum_{y\in\{0,1\}^m}\sqrt{p_y}|\psi_y\rangle|y\rangle,$$

*where* $p_y=\frac{N_y}{2^n}$. *Moreover, suppose the density corresponding to the distribution* $F$ *is* $f$ *and*

$$p(x)=\begin{cases}2^d\cdot p_y & \exists y\in\{0,1,\dots,2^m-1\} \quad s.t. \quad x\in\left(\underline{y},\overline{y}\right],\\ 0 & otherwise.\end{cases}$$

*Then we may regard* $p(x)$ *as an approximation to* $f(x)$. *To achieve the error bound* $\|f-p\|_\infty\le\epsilon$, *what values of* $M,m,n,$ *and* $d$ *should be set?*

*Solution.* For $x\in(\underline{y},\overline{y}]$, we have

$$\begin{aligned}|p(x)-f(x)| &= \left|\frac{2^d\cdot N_y}{2^n}-f(x)\right|\\ &\le \left|\frac{2^d\cdot N_y}{2^n}-2^d\cdot\widetilde{p}_y\right|+\left|2^d\cdot\widetilde{p}_y-f(x)\right|\\ &\le \frac{2^d}{2^{n-1}}+2^d\int_{\underline{y}}^{\overline{y}}|f(z)-f(x)|\,dz\\ &\le \frac{2^d}{2^{n-1}}+\frac{1}{2^d}\cdot\max_{z\in[\underline{y},\overline{y}]}|f'(z)|\\ &\le \frac{1}{2^{n-d-1}}+\frac{C}{2^d},\end{aligned}$$

with the assumption that $f'(z)$ is uniformly bounded by a constant $C$ in the real line. For $x\le-\frac{1}{2^{d+1}}-M$ or $x>2^{m-d}-M-2^{-(d+1)}$, we have $p(x)=0$ and thus

$$|p(x)-f(x)|=f(x).$$

In summary, we should choose $M,m,n,d$ such that

$$\begin{cases}\frac{1}{2^{n-d-1}}+\frac{C}{2^d}\le\epsilon,\\ f(x)\le\epsilon, \quad \forall x\in(-\infty,-\frac{1}{2^{d+1}}-M]\cup(2^{m-d}-M-2^{-(d+1)},+\infty).\end{cases}\tag{3.4}$$

For the case of standard Normal density

$$f(x)=\frac{1}{\sqrt{2\pi}}e^{-x^2/2},$$

it is reasonable to construct a symmetric set of grids w.r.t. the original point, and hence we choose

$$\frac{2M}{2^m}=\frac{1}{2^d}\Longrightarrow M=2^{m-d-1}.$$

Moreover, we can set $C=\frac{1}{\sqrt{\pi}e^{1/4}}$. Solving (3.4), we get the condition

$$\begin{cases}\frac{1}{2^{n-d-1}}+\frac{C}{2^d}\le\epsilon,\\ 2^{m-d-1}=M\ge 2^{-(d+1)}+\sqrt{2\log\frac{1}{\sqrt{2\pi}\epsilon}}.\end{cases}$$

We can choose

$$\begin{aligned} d &= 1 + \log_2 \frac{C}{\epsilon}, \\ n &= d + 2 + \log_2 \frac{1}{\epsilon} = 3 + 2\log_2 \frac{\sqrt{C}}{\epsilon}, \\ m &= d + \frac{5}{2} + \frac{1}{2}\log_2 \log \frac{1}{\sqrt{2\pi}\epsilon}, \end{aligned}$$

where $C = \frac{1}{\sqrt{2\pi e}}$ for standard normal distribution. □

**Remark 3.2.** *In the following application, we use the state* $\sum_y \sqrt{p_y}|y\rangle$ *to approximate some expectation* $\int g(z)f(z)dz$ *by*

$$\sum_y p_y g(y).$$

*The error is*

$$\begin{aligned} &\left|\sum_y p_y g(y) - \int g(z)f(z)dz\right| \\ \le &\left|\sum_y p_y g(y) - \sum_y \widetilde{p}_y g(y)\right| + \left|\sum_y \widetilde{p}_y g(y) - \int g(z)f(z)dz\right| \\ \le &\sum_y |g(y)| \cdot |p_y - \widetilde{p}_y| + \sum_y \int_{\underline{y}}^{\overline{y}} |g(y) - g(z)| f(z)dz \\ &+ \int_{-\infty}^{-M-2^{-d-1}} |g(z)|f(z)dz + \int_{2^{m-d}-2^{-d-1}-M}^{+\infty} |g(z)|f(z)dz. \end{aligned}$$

*We can see the last three terms are independent of the parameter* $n$*. Therefore, to determine the parameter* $n$*, it is much more reasonable to consider the error term*

$$\sum_y |g(y)| \cdot |p_y - \widetilde{p}_y| \le 2^{-n+1} \sum_y |g(y)|.$$

*For bounded valued* $g$*, we need to set* $n = 2m$ *approximately, which is consistent to the above result.*

In many cases, the formula of the inverse CDF is not given explicitly and we need to choose suitable approximation. The next question is how to implement a function $\widehat{F}_b^{-1}$ satisfying (3.1) based on some approximation $\widehat{F}^{-1}$ to $F^{-1}$.

**Question 3.3.** *Suppose* $\widehat{F}^{-1} : \mathbb{R} \mapsto [0,1]$ *is an approximation to* $F^{-1}$ *in the sense that*

$$\left|\widehat{F}^{-1}\left(\frac{x}{2^n}\right) - F^{-1}\left(\frac{x}{2^n}\right)\right| \le \delta, \quad \forall x \in \{1, \dots, 2^n - 1\},$$

*and* $\widehat{F}^{-1}(0) = -\infty$*. Can we implement* $\widehat{F}_b^{-1}$ *satisfying (3.1) based on* $\widehat{F}^{-1}$*, and what is the requirement to* $\delta$*? If the condition (3.1) can not be satisfied, how to get a reasonable approximation?*

*Solution.* Unfortunately, I think (3.1) can not be implemented by $\widehat{F}^{-1}$. For a given $x \in \{1, \dots, 2^n - 1\}$, we use $B_x$ to denote the binary representation of $\widehat{F}^{-1}\left(\frac{x}{2^n}\right)$,

$$B_x = s_{m-d-1} \cdots s_1 s_0 . s_{-1} s_{-2} \cdots .$$

Here, we assume $\left|\widehat{F}^{-1}\left(\frac{x}{2^n}\right)\right| < 2^{m-d}$, and the most significant position $s_{m-d-1}$ denote the sign of the number, 0 for negative and 1 for positive. Thus, the scheme of $B_x$ can represent numbers in the interval $[-2^{m-d-1}, 2^{m-d-1}]$. Let $\mathtt{rd}_{m,d}(\cdot)$ denote the following operation:

$$\mathtt{rd}_{m,d}(B_x) = s_{m-d-1} \cdots s_1 s_0 . s_{-1} \cdots (s_{-d} + s_{-d-1}).$$

With this truncation, the numbers can be represented are in the interval $[-2^{m-d-1}+2^{-d}, 2^{m-d-1}-2^{-d}]$ and of the form $N \cdot 2^{-d}$ for some integer $N$. For convenience, we define the set

$$S := [-2^{m-d-1} + 2^{-d}, 2^{m-d-1} - 2^{-d}] \cap \left\{N \cdot 2^{-d} : N \in \mathbb{Z}\right\}.$$

Then, it can be shown that $|\mathtt{rd}_{m,d}(B_x) - B_x| \le 2^{-d-1}$:

- If $s_{-d-1} = 0$, then

$$\begin{aligned} B_x - \mathtt{rd}_{m,d}(B_x) &= 0.0 \cdots 0 s_{-d-1} s_{-d-2} \cdots \\ &= 0.0 \cdots 00 s_{-d-2} \cdots \\ &\le 0.0 \cdots 0011 \cdots = 2^{-d-1}. \end{aligned}$$

- If $s_{-d-1} = 1$, then

$$\begin{aligned} B_x - \mathtt{rd}_{m,d}(B_x) &= -2^{-d} + 0.0 \cdots 0 s_{-d-1} s_{-d-2} \cdots \\ &= -2^{-d} + 0.0 \cdots 01 s_{-d-2} \cdots \\ &\ge -2^{-d} + 0.0 \cdots 010 \cdots = -2^{-d-1}. \end{aligned}$$

Then $\widehat{F}_b^{-1}$ can be defined by

$$\widehat{F}_b^{-1}(x) := 2^d \cdot (\mathtt{rd}_{m,d}(B_x) + M).$$

Thus, in the case $\mathtt{rd}_{m,d}(B_x) + M \in S$ we have

$$\frac{\widehat{F}_b^{-1}(x)}{2^d} - M = \mathtt{rd}_{m,d}(B_x).$$

With triangular inequality, for $x \in \{1, \dots, 2^n - 1\}$, we have

$$\begin{aligned} \left|F^{-1}\left(\frac{x}{2^n}\right) - \left(\frac{\widehat{F}_b^{-1}(x)}{2^d} - M\right)\right| &\le \left|F^{-1}\left(\frac{x}{2^n}\right) - \widehat{F}^{-1}\left(\frac{x}{2^n}\right)\right| + \left|\widehat{F}^{-1}\left(\frac{x}{2^n}\right) - \left(\frac{\widehat{F}_b^{-1}(x)}{2^d} - M\right)\right| \\ &= \left|F^{-1}\left(\frac{x}{2^n}\right) - \widehat{F}^{-1}\left(\frac{x}{2^n}\right)\right| + |B_x - \mathtt{rd}_{m,d}(B_x)| \\ &\le \delta + 2^{-d-1}. \end{aligned}$$

Thus, we can see (3.1) is not achievable in realistic since the round off to a $d$-accurate register causes an magnitude $2^{-d-1}$ error and the approximately computing of $F^{-1}$ causes extra error $\delta$. □

**Remark 3.4.** *From the above definition of* $\widehat{F}_b^{-1}$*, we see that*

$$\min_{x\in\{1,\dots,2^n-1\}} \widehat{F}_b^{-1}(x) = \widehat{F}_b^{-1}(x) := 2^d\left(\mathtt{rd}_{m,d}(B_1)+M\right).$$

*In order to take the full advantage of the register storing* $\widehat{F}_b^{-1}$*, the value of* $M$ *should be chosen such that* $\widehat{F}_b^{-1}(1)=0$*.*

**Question 3.5.** *What happens if we relax the condition* (3.1) *to*

$$\begin{cases} -\gamma < F^{-1}\left(\frac{x}{2^n}\right) - \left(\frac{\widehat{F}_b^{-1}(x)}{2^d} - M\right) \le \gamma, \quad \forall x = 1,\dots,2^n-1, \\ \widehat{F}_b^{-1}(0) = 0, \end{cases} \tag{3.5}$$

*with* $\gamma \in (2^{-d-1}, 2^{-d})$*? We assume that* $F$ *is continuous over the real line, and has density function* $f$*.*

*Sol.* At first, we investigate the case $\gamma < 2^{-d-1}$, which will give us some insight to case $\gamma > 2^{-d-1}$.

- In the case $\gamma < 2^{-d-1}$, we have

$$\left\{x\in\{0,1\}^n : \widehat{F}_b^{-1}(x) = y\right\} = \left\{x\in\{0,1\}^n : -\gamma < F^{-1}\left(\frac{x}{2^n}\right) - \left(\frac{y}{2^d} - M\right) \le \gamma\right\}.$$

  For one direction, if $x$ is in the first set, then $y = \widehat{F}_b^{-1}(x)$ satisfying the condition (3.5), and thus $x$ belongs to the second set also. For another direction, if $x\in\{0,1\}^n$ satisfies

$$-\gamma < F^{-1}\left(\frac{x}{2^n}\right) - \left(\frac{y}{2^d} - M\right) \le \gamma$$

  for a given $y$, but $\widehat{F}_b^{-1}(x) \neq y$. Then we have $\left|\widehat{F}_b^{-1}(x) - y\right| \ge 1$, and thus

$$\begin{aligned} \left|F^{-1}\left(\frac{x}{2^n}\right) - \left(\frac{\widehat{F}_b^{-1}(x)}{2^d} - M\right)\right| &\ge \left|\frac{y - \widehat{F}_b^{-1}(x)}{2^d}\right| - \left|F^{-1}\left(\frac{x}{2^n}\right) - \left(\frac{y}{2^d} - M\right)\right| \\ &\ge 2^{-d} - \gamma \ge 2^{-d-1} > \gamma, \end{aligned}$$

  which is contradict to the condition (3.5).
- In the case $\gamma \ge 2^{-d-1}$ we can not argue $2^{-d} - \gamma \ge 2^{-d-1} > \gamma$ again for the second direction, but the first direction still holds true:

$$\left\{x\in\{0,1\}^n : \widehat{F}_b^{-1}(x) = y\right\} \subset \left\{x\in\{0,1\}^n : -\gamma < F^{-1}\left(\frac{x}{2^n}\right) - \left(\frac{y}{2^d} - M\right) \le \gamma\right\}.$$

  For the second direction, we define the quantity $\bar{\gamma} = 2^{-d} - \gamma$, if $x\in\{0,1\}^n$ satisfying

$$-\bar{\gamma} < F^{-1}\left(\frac{x}{2^n}\right) - \left(\frac{y}{2^d} - M\right) < \bar{\gamma}$$

  for a given $y$. Then we have $\widehat{F}_b^{-1}(x) = y$, or we have $\left|\widehat{F}_b^{-1}(x) - y\right| \ge 1$ and thus

$$\begin{aligned} \left|F^{-1}\left(\frac{x}{2^n}\right) - \left(\frac{\widehat{F}_b^{-1}(x)}{2^d} - M\right)\right| &\ge \left|\frac{y - \widehat{F}_b^{-1}(x)}{2^d}\right| - \left|F^{-1}\left(\frac{x}{2^n}\right) - \left(\frac{y}{2^d} - M\right)\right| \\ &> 2^{-d} - \bar{\gamma} = \gamma, \end{aligned}$$

which is contradict to the condition (3.5). In conclusion, we have

$$\left\{x\in\{0,1\}^n:\widehat{F}_b^{-1}(x)=y\right\}\subset\left\{x\in\{0,1\}^n:-\gamma<F^{-1}\left(\frac{x}{2^n}\right)-\left(\frac{y}{2^d}-M\right)\leq\gamma\right\},$$
$$\left\{x\in\{0,1\}^n:\widehat{F}_b^{-1}(x)=y\right\}\supset\left\{x\in\{0,1\}^n:-\bar{\gamma}<F^{-1}\left(\frac{x}{2^n}\right)-\left(\frac{y}{2^d}-M\right)<\bar{\gamma}\right\}.$$

Let

$$\overline{p_y}=\int_{-\gamma+\frac{y}{2^d}-M}^{\gamma+\frac{y}{2^d}-M}f(x)dx,\quad \underline{p_y}=\int_{-\bar{\gamma}+\frac{y}{2^d}-M}^{\bar{\gamma}+\frac{y}{2^d}-M}f(x)dx,$$

and $N_y$ be the number of elements in $\left\{x\in\{0,1\}^n:\widehat{F}_b^{-1}(x)=y\right\}$, then we have

$$\frac{\underline{p_y}}{2^{-n}}-2<N_y<\frac{\overline{p_y}}{2^{-n}}.$$

Then, we get

$$\frac{N_y}{2^n}\in\left(\underline{p_y}-2^{-n+1},\overline{p_y}\right).$$

The desired quantity $p_y$ is

$$p_y=\int_{-M+\frac{y}{2^d}-\frac{1}{2^{d+1}}}^{-M+\frac{y}{2^d}+\frac{1}{2^{d+1}}}f(x)dx=\int_{\underline{y}}^{\overline{y}}f(x)dx,$$

where we write $\underline{y}\equiv-M+\frac{y}{2^d}-\frac{1}{2^{d+1}}$ and $\overline{y}\equiv-M+\frac{y}{2^d}+\frac{1}{2^{d+1}}$ for simplicity. Then we have

$$\left|\frac{N_y}{2^n}-p_y\right|\leq\max\left\{\overline{p_y}-p_y,p_y+2^{-n+1}-\underline{p_y}\right\},$$

where

$$\begin{aligned}\overline{p_y}-p_y&=\int_{-\delta+\underline{y}}^{\delta+\overline{y}}f(x)dx-\int_{\underline{y}}^{\overline{y}}f(x)dx=\int_{-\delta+\underline{y}}^{\underline{y}}f(x)dx+\int_{\overline{y}}^{\delta+\overline{y}}f(x)dx\\&\leq 2\delta\cdot\max_{x\in\mathbb{R}}f(x),\\p_y-\underline{p_y}&=\int_{\underline{y}}^{\overline{y}}f(x)dx-\int_{\delta+\underline{y}}^{-\delta+\overline{y}}f(x)dx=\int_{\underline{y}}^{\delta+\underline{y}}f(x)dx+\int_{-\delta+\overline{y}}^{\overline{y}}f(x)dx\\&\leq 2\delta\cdot\max_{x\in\mathbb{R}}f(x).\end{aligned}$$

Here, we write $\gamma=2^{-d-1}+\delta$ for some $\delta\in(0,2^{-d-1})$. In conclusion, we have

$$\left|\frac{N_y}{2^n}-p_y\right|\leq 2^{-n+1}+2\delta\cdot\max_{x\in\mathbb{R}}f(x).$$

□

3.2. **Implement the Inverse Transformation for Standard Normal Distribution.** Here we consider the gate complexity to implement the function $\Phi^{-1}$, where

$$\Phi(x)=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x}e^{-\frac{z^2}{2}}dz.$$

To preserve the quadratic speedup of QMCI, we should use some approximations to $\Phi^{-1}$ with reasonable accuracy.

There is a plenty of works on the topic of approximating the inverse Normal CDF. For example in [8], there are some simple approximations

$$\Phi^{-1}(p) \approx 1.724 - 2.22t + 0.157t^2, \quad t = \sqrt{-\log p},$$
$$\Phi^{-1}(p) \approx 1.758 - 2.257t + 0.1661t^2, \quad t = \sqrt{-\log p}.$$

For more accurate approximations, there are

$$\begin{aligned} &y = -\log\left(2(1-p)\right), \\ &\Phi^{-1}(p) \approx \sqrt{\frac{\left((4y+100)\,y+205\right)y^2}{\left((2y+56)\,y+192\right)y+131}} = y\cdot\sqrt{\frac{(4y+100)\,y+205}{\left((2y+56)\,y+192\right)y+131}} \end{aligned} \tag{3.6}$$

for $\frac{1}{2} < p < 1 - 10^{-7}$. This method is recorded in [13], and is claimed to achieve the approximate error within $1.3\times10^{-4}$ in the interval $(0.5, 1-10^{-7})$. There is another interesting approximation in [13], say

$$\Phi^{-1}(p) \approx \frac{p^{0.135} - (1-p)^{0.135}}{0.1975}, \quad 0 < p < 1. \tag{3.7}$$

In the case $|\Phi^{-1}(p)| \le 2$, this approximation achieve an error within 0.0093. These approximations are not that accurate, but I think they are enough for the present quantum hardware. For more accurate approximation, please refer to [8].

To implement such approximation in quantum circuit, I refered to [4] for the fixed point addition and multiplication. The quantum circuit implementation of $\log(\cdot)$ and $\sqrt{\cdot}$ should be considered further, or we use the approximation (3.7) with an implementation of $(\cdot)^{0.135}$. Since the more accurate approximation usually involves square root and logarithm, I think it is worth to talk about them.

**Question 3.6.** *What is the advantage of this method compared to the classical Grover-Rudolph method?*

3.3. **Box-Muller Method.** Suppose $U_1, U_2 \overset{\text{i.i.d.}}{\sim} \text{Unif}[0,1]$, and

$$\begin{cases} Z_1 = \sqrt{-2\log U_1}\cos\left(2\pi U_2\right), \\ Z_2 = \sqrt{-2\log U_1}\sin\left(2\pi U_2\right). \end{cases}$$

Then we have $Z_1, Z_2 \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$. To avoid the computation of sine and cosine, we can use the following procedure

$$\begin{aligned} &U_1 \leftarrow 2U_1 - 1, \quad U_2 \leftarrow 2U_2 - 1, \\ &X = U_1^2 + U_2^2, \quad Y = \sqrt{-2\log X/X}, \\ &Z_1 = U_1 Y, \quad Z_2 = U_2 Y. \end{aligned}$$

Moreover, to generate random vectors satisfying

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}\right),$$

we can compute

$$\begin{cases} X_1 = Z_1, \\ X_2 = \rho Z_1 + \sqrt{1-\rho^2}Z_2. \end{cases}$$

3.4. **Can we reduce the required qubits?** Figure 1 is a simple illustration to the inverse transformation method. Here we set $n = 5, m = 3, M = 1.75, d = 1$, and try to approximate the standard normal distribution. There are $2^n$ black points on the horizontal axis equally spaced by $2^{-n}$, and $2^m$ red points on the vertical axis equally spaced by $2^{-d}$. Then, we map the red points to the horizontal axis by the function $F(\cdot)$.

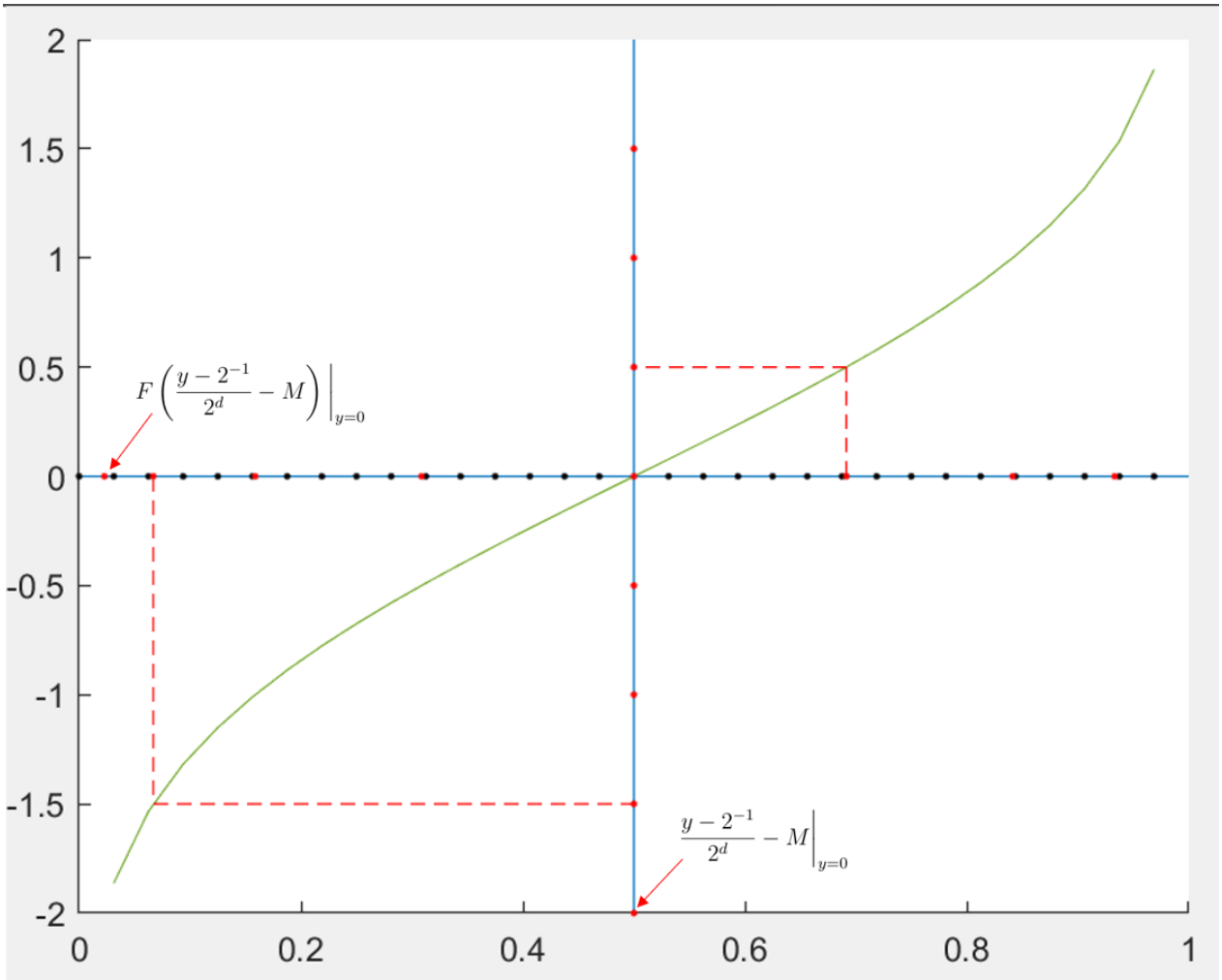



FIGURE 1. A Simple Illustration to The Inverse Method.

Finally, we can count the number of black points between two adjacent red points as the quantity $N_y$ and get an approximate distribution to the standard normal distribution.

In this case, if we consider the corresponding quantum computation process, that is

$$\begin{aligned}
|0^5\rangle|0^3\rangle &\mapsto \sum_{x\in\{0,1\}^5} \frac{1}{\sqrt{2^5}}|x\rangle|0^3\rangle \\
&\mapsto (|00000\rangle + |00001\rangle + |00010\rangle)\,|000\rangle \\
&\quad + (|00011\rangle + |00100\rangle + |00101\rangle)\,|001\rangle \\
&\quad + (|00110\rangle + |00111\rangle + |01000\rangle + |01001\rangle)\,|010\rangle \\
&\quad + (|01010\rangle + |01011\rangle + |01100\rangle + |01101\rangle + |01110\rangle + |01111\rangle + |10000\rangle)\,|011\rangle \\
&\quad + (|10001\rangle + |10010\rangle + |10011\rangle + |10100\rangle + |10101\rangle + |10110\rangle)\,|100\rangle \\
&\quad + (|10111\rangle + |11000\rangle + |11001\rangle + |11010\rangle)\,|101\rangle \\
&\quad + (|11011\rangle + |11100\rangle + |11101\rangle)\,|110\rangle \\
&\quad + (|11110\rangle + |11111\rangle)\,|111\rangle.
\end{aligned}$$

Then, we can observe that if we only preserve the first three qubits in the first register, we will get the same distribution when we only measure the qubits in the second register. The reduced qubits state is

$$\begin{aligned}
&(|000\rangle + |001\rangle + |010\rangle)\,|000\rangle \\
&\quad + (|011\rangle + |100\rangle + |101\rangle)\,|001\rangle \\
&\quad + (|110\rangle + |111\rangle + |000\rangle + |001\rangle)\,|010\rangle \\
&\quad + (|010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle + |000\rangle)\,|011\rangle \\
&\quad + (|001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle)\,|100\rangle \\
&\quad + (|111\rangle + |000\rangle + |001\rangle + |010\rangle)\,|101\rangle \\
&\quad + (|011\rangle + |100\rangle + |101\rangle)\,|110\rangle \\
&\quad + (|110\rangle + |111\rangle)\,|111\rangle.
\end{aligned}$$

This means there is a space to reduce the required qubits.

**Question 3.7.** *How to reduce the required qubits? Find such methods as many as possible.*

*Sol.* To reduce qubits, we define the function $\widehat{F}_b : \{0,1\}^m \mapsto \{0,1\}^n$ as

$$\widehat{F}_b(y) = \begin{cases} 0, & y = 0 \\ \left\lceil 2^n F\left(\frac{y-2^{-1}}{2^d} - M\right)\right\rceil, & y \neq 0. \end{cases}$$

In the above illustration example, with the definition of $\widehat{F}_b^{-1}$ and $\widehat{F}_b$, we may have

$$\begin{aligned}
&\widehat{F}_b(000) = 00000, \quad \widehat{F}_b(001) = 00011, \quad \widehat{F}_b(010) = 00110, \quad \widehat{F}_b(011) = 01010, \\
&\widehat{F}_b(100) = 10001, \quad \widehat{F}_b(101) = 10111, \quad \widehat{F}_b(110) = 11011, \quad \widehat{F}_b(111) = 11110.
\end{aligned}$$

Suppose there is a quantum circuit to implement the transformation: $|y\rangle_m|0\rangle_n \mapsto |y\rangle|\widehat{F}_b(y)\rangle$, then we calim the following process can reduce the required qubits to represent the distribution corresponding to $F$.

$$\begin{aligned}
|0\rangle_n|0\rangle_m|0\rangle_n &\mapsto \frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}|x\rangle_n|0\rangle_m|0\rangle_n \mapsto \frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}|x\rangle_n|\widehat{F}_b^{-1}(x)\rangle_m|0\rangle_n \\
&\overset{\mathcal{U}}{\mapsto} \frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}|x\rangle_n|\widehat{F}_b^{-1}(x)\rangle_m\left|\widehat{F}_b\left(\widehat{F}_b^{-1}(x)\right)\right\rangle_n \\
&\mapsto \frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}\left|x-\widehat{F}_b\left(\widehat{F}_b^{-1}(x)\right)\right\rangle_n|\widehat{F}_b^{-1}(x)\rangle_m\left|\widehat{F}_b\left(\widehat{F}_b^{-1}(x)\right)\right\rangle_n \\
&\overset{\mathcal{U}^{-1}}{\mapsto} \frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}\left|x-\widehat{F}_b\left(\widehat{F}_b^{-1}(x)\right)\right\rangle_n|\widehat{F}_b^{-1}(x)\rangle_m|0\rangle_n .
\end{aligned}$$

Apply this process to the illustration example, we will finally get the state:

$$\begin{aligned}
&(|00000\rangle+|00001\rangle+|00010\rangle)|000\rangle|00000\rangle \\
&+(|00000\rangle+|00001\rangle+|00010\rangle)|001\rangle|00000\rangle \\
&+(|00000\rangle+|00001\rangle+|00010\rangle+|00011\rangle)|010\rangle|00000\rangle \\
&+(|00000\rangle+|00001\rangle+|00010\rangle+|00011\rangle+|00100\rangle+|00101\rangle+|00110\rangle)|011\rangle|00000\rangle \\
&+(|00000\rangle+|00001\rangle+|00010\rangle+|00011\rangle+|00100\rangle+|00101\rangle)|100\rangle|00000\rangle \\
&+(|00000\rangle+|00001\rangle+|00010\rangle+|00011\rangle)|101\rangle|00000\rangle \\
&+(|00000\rangle+|00001\rangle+|00010\rangle)|110\rangle|00000\rangle \\
&+(|00000\rangle+|00001\rangle)|111\rangle|00000\rangle \\
=|00\rangle\big[&(|000\rangle+|001\rangle+|010\rangle)|000\rangle \\
&+(|000\rangle+|001\rangle+|010\rangle)|001\rangle \\
&+(|000\rangle+|001\rangle+|010\rangle+|011\rangle)|010\rangle \\
&+(|000\rangle+|001\rangle+|010\rangle+|011\rangle+|100\rangle+|101\rangle+|110\rangle)|011\rangle \\
&+(|000\rangle+|001\rangle+|010\rangle+|011\rangle+|100\rangle+|101\rangle)|100\rangle \\
&+(|000\rangle+|001\rangle+|010\rangle+|011\rangle)|101\rangle \\
&+(|000\rangle+|001\rangle+|010\rangle)|110\rangle \\
&+(|000\rangle+|001\rangle)|111\rangle\big]|00000\rangle.
\end{aligned}$$

Thus, the first two and the last five qubits are released. □

**Question 3.8.** *How many qubits can we reduce?*

*Sol.* Let $f\equiv F'$ be the density function of the distribution $F$, and suppose $f_{max}=\max_{x\in\mathbb{R}}f(x)$. Then, we have

$$N_y=\frac{F\left(\frac{y+2^{-1}}{2^d}-M\right)-F\left(\frac{y-2^{-1}}{2^d}-M\right)}{\frac{1}{2^n}}\le\frac{f_{max}\left(\frac{y+2^{-1}}{2^d}-M-\frac{y-2^{-1}}{2^d}+M\right)}{\frac{1}{2^n}}=2^{n-d}f_{max},$$

where $N_y$ is defined the same as in (3.3). Then, the number of qubits can be reduced is at least

$$n - \log_2(2^{n-d} f_{max}) = d - \log_2 f_{max}.$$

In the case of standard normal distribution, we have $f_{max} = \frac{1}{\sqrt{2\pi}}$ and thus the number of qubits can be reduced is at least $d + \log_2(\sqrt{2\pi}) \geq d + 1$. □

## References


[1] I. F. Araujo, D. K. Park, F. Petruccione, and A. J. da Silva. A divide-and-conquer algorithm for quantum state preparation. *Scientific reports*, 11(1):6329, 2021.

[2] S. Chakrabarti, R. Krishnakumar, G. Mazzola, N. Stamatopoulos, S. Woerner, and W. J. Zeng. A threshold for quantum advantage in derivative pricing. *Quantum*, 5:463, 2021.

[3] L. Grover and T. Rudolph. Creating superpositions that correspond to efficiently integrable probability distributions. *arXiv preprint quant-ph/0208112*, 2002.

[4] T. Häner, M. Roetteler, and K. M. Svore. Optimizing quantum circuits for arithmetic. *arXiv preprint arXiv:1805.12445*, 2018.

[5] S. Herbert. No quantum speedup with grover-rudolph state preparation for quantum monte carlo integration. *Physical Review E*, 103(6):063302, 2021.

[6] A. Holmes and A. Y. Matsuura. Efficient quantum circuits for accurate state preparation of smooth, differentiable functions. pages 169–179, 2020.

[7] J. Iaconis, S. Johri, and E. Y. Zhu. Quantum state preparation of normal distributions using matrix product states. *arXiv preprint arXiv:2303.01562*, 2023.

[8] S. A. Justin S.Dyer. Approximations to inverse error functions.

[9] A. Manzano, A. Gómez, and C. C. Vázquez. D5. 7: Update of review of state-of-the-art for pricing and computation of var. 2023.

[10] G. Marin-Sanchez, J. Gonzalez-Conde, and M. Sanz. Quantum algorithms for approximate function loading. *Physical Review Research*, 5(3):033114, 2023.

[11] S. McArdle, A. Gilyén, and M. Berta. Quantum state preparation without coherent arithmetic. *arXiv preprint arXiv:2210.14892*, 2022.

[12] A. Montanaro. Quantum speedup of monte carlo methods. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2181):20150301, 2015.

[13] J. K. Patel and C. B. Read. *Handbook of the normal distribution*, volume 150. CRC Press, 1996.

[14] A. G. Rattew, Y. Sun, P. Minssen, and M. Pistoia. The efficient preparation of normal distributions in quantum registers. *Quantum*, 5:609, 2021.

[15] P. Rebentrost, B. Gupt, and T. R. Bromley. Quantum computational finance: Monte carlo pricing of financial derivatives. *Physical Review A*, 98(2):022321, 2018.

[16] A. C. Vazquez and S. Woerner. Efficient state preparation for quantum amplitude estimation. *Physical Review Applied*, 15(3):034027, 2021.