

# Quantum Monte Carlo algorithm for solving Black-Scholes PDEs for high-dimensional option pricing in finance and its complexity analysis

Jianjun Chen<sup>1</sup>, Yongming Li<sup>2</sup>, Ariel Neufeld<sup>3</sup>

<sup>1</sup> Division of Physics and Applied Physics, School of Physical and Mathematical Sciences,  
Nanyang Technological University, Singapore, e-mail: chen1554@e.ntu.edu.sg.

<sup>2</sup> Department of Mathematics, Texas A&M University,  
Texas, USA, e-mail: liyo0008@tamu.edu

<sup>3</sup> Division of Mathematical Sciences, School of Physical and Mathematical Sciences,  
Nanyang Technological University, Singapore, e-mail: ariel.neufeld@ntu.edu.sg

May 29, 2025

## Abstract

In this paper we provide a quantum Monte Carlo algorithm to solve high-dimensional Black-Scholes PDEs with correlation for high-dimensional option pricing. The payoff function of the option is of general form and is only required to be continuous and piece-wise affine (CPWA), which covers most of the relevant payoff functions used in finance. We provide a rigorous error analysis and complexity analysis of our algorithm. In particular, we prove that the computational complexity of our algorithm is bounded polynomially in the space dimension  $d$  of the PDE and the reciprocal of the prescribed accuracy  $\varepsilon$ . Moreover, we show that for payoff functions which are bounded, our algorithm indeed has a speed-up compared to classical Monte Carlo methods. Furthermore, we provide numerical simulations in one and two dimensions using our developed package within the `Qiskit` framework tailored to price CPWA options with respect to the Black-Scholes model, as well as discuss the potential extension of the numerical simulations to arbitrary space dimension.

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Setting and Main result</b>	<b>3</b>
2.1	Black-Scholes PDE for option pricing . . . . .	3
2.1.1	Geometric Brownian motion process for the price evolution of multiple assets . . . . .	3
2.1.2	Continuous piece-wise affine (CPWA) payoff functions . . . . .	4
2.2	Brief Introduction to Quantum Computing . . . . .	5
2.2.1	Dirac Bra-Ket notation and tensor products . . . . .	5
2.2.2	Qubits, quantum gates, and quantum circuits . . . . .	6
2.2.3	Quantum measurements . . . . .	9
2.3	Quantum amplitude estimation algorithms . . . . .	9
2.4	Algorithm 1 and main result . . . . .	11
2.4.1	Outline of Algorithm 1 . . . . .	11
2.4.2	Main Theorem . . . . .	13
<b>3</b>	<b>Numerical Simulations</b>	<b>14</b>
3.1	Vanilla Call Option . . . . .	15
3.2	Basket Call Option . . . . .	15
3.3	Spread Call Option . . . . .	16
3.4	Call-on-max Option . . . . .	17
3.5	Call-on-min Option . . . . .	18
3.6	Best-of-call Option . . . . .	18
3.7	Discussion on the numerical simulations and their possible extension to higher dimensions . . . . .	18
<b>4</b>	<b>Quantum Circuits</b>	<b>20</b>
4.1	Representing signed dyadic rationals using the two's complement method . . . . .	21
4.2	Quantum circuits for elementary arithmetic operations . . . . .	23
4.3	Distribution loading . . . . .	28
4.4	Loading CPWA payoff functions . . . . .	29
<b>5</b>	<b>Error analysis</b>	<b>44</b>
5.1	Step 1: Truncation error bounds . . . . .	45
5.2	Step 2: Quadrature error bounds . . . . .	46
5.3	Step 3: Approximation error bounds for payoff function . . . . .	48
5.4	Step 4: Distribution loading error bounds . . . . .	49
5.5	Step 5: Rotation error bounds . . . . .	50
5.6	Step 6: Quantum amplitude estimation error bounds . . . . .	52
<b>6</b>	<b>Proof of Theorem 1</b>	<b>53</b>
<b>7</b>	<b>Conclusion</b>	<b>55</b>

# 1 Introduction

An option in finance is a contract between a seller and a buyer, which provides a future payoff to the buyer of the option at the maturity date in dependence of the underlying financial securities involved in the contract, such as, for example, stocks or indexes. Since the underlying securities evolve randomly over time and hence their future values at maturity cannot be known at any previous time neither by the buyer nor the seller, the exact payoff the buyer of the option receives at maturity cannot be known either. Hence a key problem in financial theory is to evaluate the value of such options, i.e. to define and determine a fair price between buyer and seller of the option under consideration.

In 1973, Black, Scholes, and Merton introduced in [10] and [49] the so-called Black-Scholes-Merton model, also known simply as the Black-Scholes model, which is a pricing model allowing to evaluate the value of options which only depend on the underlying single security. They have shown in [10] and [49] that under the Black-Scholes model, the fair price of an option can be characterized as a solution of a particular partial differential equation (PDE). More precisely, at each time  $t \leq T$  the value of an option with corresponding payoff function  $h : \mathbb{R}_+ \rightarrow \mathbb{R}$ , which provides a payoff  $h(S_T)$  to the buyer in dependence of the value of the underlying single security  $S_T$  at maturity time  $T$ , coincides with the solution  $u(t, x)$  of the so-called Black-Scholes PDE with terminal condition  $h$ , given that the value  $S_t$  of the underlying security at time  $t$  satisfies  $S_t = x$ .

Later, the Black-Scholes model has been extended to price options involving multiple securities and to incorporate their correlations into consideration. Analogously to the original model, the price of an option can be characterized by a PDE, whose space dimension  $d$  corresponds to the amount of securities involved in the financial contract. For example, an option depending on the index of the S&P 500 could be priced in the (multidimensional) Black-Scholes model by the solution of a 500-dimensional PDE.

Unfortunately, in the multidimensional setting of the Black-Scholes model, there are no explicit expressions for the solution of the corresponding PDE, and hence numerical methods are necessary to approximately solve these high-dimensional PDEs. It is crucial both from a theoretical, but especially also from a practical point of view to provide a rigorous error analysis and complexity analysis of these numerical methods. Indeed, while the precise error analysis allows, in the context of option pricing, to precisely determine the true absolute error of the output of the numerical algorithm to the theoretical price of the option under consideration, a rigorous complexity analysis allows to determine how well an algorithm scales in dependence of the space-dimension of the underlying PDE. Ideally, one would like to build a numerical algorithm whose precise error and complexity analysis can be determined and whose number of computational operations only grows *polynomially* in the dimension  $d$  and the reciprocal of the precision  $\varepsilon$ . Due to the Feynman-Kac formula for the Black-Scholes PDE, allowing to write the solution of the Black-Scholes PDE in form of an expectation of the payoff function  $h : \mathbb{R}_+^d \rightarrow \mathbb{R}$  with respect to the multivariate log-normal distribution, Monte Carlo based algorithms have demonstrated both theoretically and practically to be efficient for pricing high-dimensional options under the Black-Scholes model. In particular, the computational complexity of Monte Carlo methods typically does not grow exponentially in the dimension  $d$  and the reciprocal of the precision  $\varepsilon$ . We refer to, e.g., [2, 30] for numerical methods which involve finite difference or finite element, to, e.g., [11] for Monte Carlo based methods, as well as to, e.g., [8, 9, 27, 34, 37] for deep learning based methods to approximately solve the (multidimensional) Black-Scholes PDE.

In recent years, there has been a rapid development of numerical methods dealing with problems in quantitative finance using quantum computers. The motivation comes from the fact that qubits, compared to classical bits, are allowed quantum mechanically to be in a state of superposition, from which one anticipates that quantum computers should be able to achieve much higher computational power than classical (super-) computers. We also refer to [32] for a universal approximation theorem for quantum neural networks. The applications of quantum algorithms in finance include portfolio optimization [64], the computation of risk measures such as *Value at Risk* (VAR) [74], volatility modeling [7], and option pricing, particularly in the Black-Scholes model [15, 23, 28, 44, 61, 63, 64, 69]. We also refer to the monograph [42] and surveys [26, 41, 55] for (further) applications of quantum computing in finance. Furthermore, [5, 6, 18, 21, 45, 51] proposed quantum algorithms to approximately solve different PDEs than the Black-Scholes PDE used in finance for option pricing.

While [28] proposes a hybrid quantum-classical algorithm to approximately solve the one-dimensional Black-Scholes PDE exploiting its relation to the Schrödinger equation in imaginary time and [44] proposes a variational quantum approach, most literature uses quantum Monte Carlo methods to approximately solve the Black-Scholes PDE in order to price financial options. More precisely, these works rely on the *Quantum Amplitude Estimation algorithm* (QAE) [12] which estimates the expected value of a random parameter (see Section 2.3 for a detailed discussion) based on an extension of *Grover's search algorithm* [36]. Several variations of the Quantum Amplitude Estimation algorithm have been proposed recently, see e.g. [1, 29, 31, 33, 47, 52, 57, 60, 70, 73, 75]. Quantum Monte Carlo methods can ideally achieve a quadratic speed-up [38],[50] compared to classical (i.e. non-quantum) Monte Carlo methods. However, the quadratic speedup can only be achieved if there is a so-called *oracle* quantum circuit which can correctly upload the corresponding distribution in rotated form, without any approximation errors (caused, e.g., from discretization and rotation), such that it is applicable to a quantum amplitude estimation algorithm. This assumption however in most cases cannot be justified in practice, as highlighted, e.g., in [15, 76].

In this paper, we propose a quantum Monte Carlo algorithm to solve high-dimensional Black-Scholes PDEs with correlation and general payoff function which is continuous and piece-wise affine (CPWA), enabling to price most

relevant payoff functions used in finance (see also Section 2.1.2). Our algorithm follows the idea of the quantum Monte Carlo algorithm proposed in [15, 63, 69] which first uploads the multivariate log-normal distribution and the payoff function in rotated form and then applies a QAE algorithm to approximately solve the Black-Scholes PDE to price options.

Our main contribution lies in a rigorous error analysis as well as complexity analysis of our algorithm. To that end, we first introduce quantum circuits that can perform arithmetic operations on two complement's numbers representing signed dyadic rational numbers, together with its complexity analysis. This allows us to provide a rigorous error and complexity analysis when uploading first a truncated and discretized approximation of the multivariate log-normal distribution and then uploading an approximation of the CPWA payoff function in rotated form, where the approximation consists of truncation as well as the rounding of the coefficients of the CPWA payoff function. This together with a rigorous error and complexity analysis when applying the modified iterative quantum amplitude estimation algorithm [29] allows us to control the output error of our algorithm to be bounded by the pre-specified accuracy level  $\varepsilon \in (0, 1)$ , while bounding its computational complexity; we refer to Theorem 1 for the precise statement of our main result. In particular, we prove that the computational complexity of our algorithm only grows polynomially in the space dimension  $d$  of the Black-Scholes PDE and in the (reciprocal of the) accuracy level  $\varepsilon$ . This is in line with classical (i.e. non-quantum) Monte Carlo methods, where the number of Monte Carlo samples required in order to approximate the solution of the Black-Scholes PDE only grows polynomially in the dimension  $d$  and  $\varepsilon^{-1}$ , see, e.g., [39]. Moreover, we show that for payoff functions which are bounded, our algorithm indeed has a speed-up compared to classical Monte Carlo methods. To the best of our knowledge, this is the first work in the literature which provides a rigorous mathematical error and complexity analysis for a quantum Monte Carlo algorithm which approximately solves high-dimensional PDEs. We refer to Remark 2.22 for a detailed discussion of the complexity analysis.

Furthermore, we provide numerical simulations in one and two dimensions for six different payoff functions. To that end, we developed a package we named `qfinance` within the `Qiskit` framework tailored to price CPWA options with respect to the Black-Scholes model. Moreover, we discuss the potential extension of the numerical simulations to arbitrary space dimension.

The rest of this paper is organized as follows. In Section 2, we introduce the main setting of this paper, present our algorithm, and state our main theorem, as well as provide a detailed discussion of our complexity analysis. In Section 3 we present our numerical simulations in one and two space dimensions as well as discuss their potential extension to higher dimensions. In Section 4, we introduce and analyze all relevant quantum circuits we need in our quantum Monte Carlo algorithm. In Section 5, we provide a detailed error analysis of the steps of our algorithm outlined in Section 2.4.1. Finally, in Section 6, we provide the proof of Theorem 1.

**Notation.** We denote the set of real numbers and positive real numbers by  $\mathbb{R}$  and  $\mathbb{R}_+ := (0, \infty)$ , respectively. The set of natural numbers is denoted by  $\mathbb{N} := \{1, 2, \dots\}$ , and we use  $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$ . The set of complex numbers is denoted by  $\mathbb{C}$ , and we define  $i := \sqrt{-1}$ . Moreover, we denote by  $I_2$  and  $I_2^{\otimes n}$  the corresponding identity matrices in  $\mathbb{C}^{2 \times 2}$  and  $\mathbb{C}^{2^n \times 2^n}$ , respectively, for every  $n \in \mathbb{N}$ . Furthermore, for each  $n \in \mathbb{N}$  we denote by  $\mathcal{U}(2^n)$  the set of unitary matrices in  $\mathbb{C}^{2^n \times 2^n}$ , i.e. matrices  $U \in \mathbb{C}^{2^n \times 2^n}$  satisfying  $UU^\dagger = U^\dagger U = I_2^{\otimes n}$ , where  $U^\dagger$  denotes the *conjugate transpose* of  $U$ .

## 2 Setting and Main result

### 2.1 Black-Scholes PDE for option pricing

Let  $r \in (0, \infty)$  be the risk-free interest rate, let  $T \in (0, \infty)$  be a finite time horizon determining the maturity, and let  $d \in \mathbb{N}$  be the number of assets. We consider the multiple-asset Black-Scholes PDE

$$\frac{\partial u}{\partial t} + \frac{1}{2} \sum_{i,j=1}^d C_{ij} x_i x_j \frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_{i=1}^d r x_i \frac{\partial u}{\partial x_i} - ru = 0, \quad \text{in } [0, T) \times \mathbb{R}_+^d \quad (1)$$

subjected to a terminal condition  $u(T, \cdot) = h(\cdot)$ . Here,  $h : \mathbb{R}_+^d \rightarrow \mathbb{R}$  represents the payoff function and  $u(t, \mathbf{x})$  represents the option price at time  $t$  with spot price  $\mathbf{x}$ . The covariance matrix  $\mathbf{C} = (C_{i,j})_{i,j=1}^d \in \mathbb{R}^{d \times d}$  is assumed to be symmetric positive definite with a Cholesky factorization  $\mathbf{C} = \boldsymbol{\sigma} \boldsymbol{\sigma}^\top$ , where  $\boldsymbol{\sigma} \in \mathbb{R}^{d \times d}$  is the log-volatility coefficient matrix, so that there is a unique risk-neutral measure (see, e.g., [14]). Note that the PDE (1) has a unique solution<sup>1</sup> whenever  $h : \mathbb{R}_+^d \rightarrow \mathbb{R}$  is continuous and at most polynomially growing, see, e.g., [34, Proposition 2.22, Corollary 4.5].

#### 2.1.1 Geometric Brownian motion process for the price evolution of multiple assets

In the multidimensional Black-Scholes model, the prices of the  $d$  stocks under consideration are modeled by a multidimensional geometric Brownian motion (GBM) having constant growth rate and volatility, see, e.g., [14]. We briefly describe the dynamics of the geometric Brownian motion process for multiple assets.

<sup>1</sup>The solution  $u(t, \mathbf{x})$  of the PDE (1) is meant in the viscosity sense, see, e.g., [19],[34].

Let  $(\Omega, \mathcal{F}, \mathbb{P})$  be a probability space and let  $\mathbf{W} = (W^1, \dots, W^d) : [0, T] \times \Omega \rightarrow \mathbb{R}^d$  be a standard  $d$ -dimensional Brownian motion. For a log-volatility coefficient matrix  $\boldsymbol{\sigma} \in \mathbb{R}^{d \times d}$  assumed to be invertible, let  $\boldsymbol{\sigma}_1, \dots, \boldsymbol{\sigma}_d \in \mathbb{R}^d$  denote the row vectors of matrix  $\boldsymbol{\sigma}$ , and let  $\sigma_i := \|\boldsymbol{\sigma}_i\|_{\ell^2(\mathbb{R}^d)}$ . Let  $\mathbf{S} = (S^1, \dots, S^d) : [0, T] \times \Omega \rightarrow \mathbb{R}_+^d$  be the stock price process governed by the following stochastic differential equation

$$dS_t^i = S_t^i \left( r dt + \sum_{j=1}^d \sigma_{ij} dW_t^j \right), \quad \text{for } i = 1, \dots, d, \quad (2)$$

with some initial spot price  $\mathbf{S}_0 \in \mathbb{R}_+^d$ . Here  $\mathbf{S}_t = (S_t^1, \dots, S_t^d)$  represents the values of each stock  $i = 1, \dots, d$  at time  $0 \leq t \leq T$ . Let  $\mathbf{R} = (R^1, \dots, R^d) : [0, T] \times \Omega \rightarrow \mathbb{R}^d$  be the log-return process defined component-wise by  $R_t^i = \ln(S_t^i/S_0^i)$  for  $i = 1, \dots, d$ . It follows from Itô's formula for all  $t \in [0, T]$  that

$$dR_t^i = (r - \frac{1}{2}\sigma_i^2)dt + \sum_{j=1}^d \sigma_{ij} dW_t^j, \quad \text{for } i = 1, \dots, d, \quad (3)$$

with initial condition  $R_0^i = 0$  for  $i = 1, \dots, d$ . Let  $\hat{\boldsymbol{\mu}} = (\hat{\mu}_1, \dots, \hat{\mu}_d) \in \mathbb{R}^d$  be a vector satisfying  $\hat{\mu}_i = (r - \frac{1}{2}\sigma_i^2)$  for  $i = 1, \dots, d$ . From equation (3), it holds that  $\mathbf{R}_T$  is a multivariate normal distribution with mean  $T\hat{\boldsymbol{\mu}}$  and covariance  $T\mathbf{C}$ . Hence, by taking the inverse of the log transform, we observe that the law of the stock price process  $\mathbf{S}_T = \mathbf{S}_0 \exp(\mathbf{R}_T)$  is a multivariate log-normal distribution with log-mean  $T\hat{\boldsymbol{\mu}}$  and log-covariance  $T\mathbf{C}$ . In general, for a given fixed initial condition  $(t, \mathbf{x}) \in [0, T] \times \mathbb{R}_+^d$ , there is a well-known formula for the probability transition density function of  $\mathbf{S}_T$ , subjected to the condition that  $\mathbf{S}_t = \mathbf{x}$ .

**Lemma 2.1 (Density formula)** *Let  $d \in \mathbb{N}$ , let  $\mathbf{x} = (x_1, \dots, x_d) \in \mathbb{R}_+^d$ , and let  $t \in [0, T]$ . Let  $\boldsymbol{\mu} = (\mu_1, \dots, \mu_d) \in \mathbb{R}^d$  be given by  $\mu_i = \ln(x_i) + (r - \frac{1}{2}\sigma_i^2)(T - t)$  for  $i = 1, \dots, d$  and let  $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$  be given by  $\boldsymbol{\Sigma} = (T - t)\mathbf{C}$ . Then, the stock price process  $\mathbf{S}_T$  introduced by (2) conditional on  $\mathbf{S}_t = \mathbf{x}$  follows a multivariate log-normal distribution with log-mean  $\boldsymbol{\mu}$  and log-covariance  $\boldsymbol{\Sigma}$ , and the joint transition probability density function is given by*

$$p(\mathbf{y}, T; \mathbf{x}, t) := \frac{\exp(-\frac{1}{2}(\log(\mathbf{y}) - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\log(\mathbf{y}) - \boldsymbol{\mu}))}{(2\pi)^{d/2}(\det \boldsymbol{\Sigma})^{1/2} \prod_{i=1}^d y_i}, \quad (4)$$

where for  $\mathbf{y} = (y_1, \dots, y_d) \in \mathbb{R}_+^d$ ,  $\log(\mathbf{y}) \in \mathbb{R}^d$  is given by

$$(\log(\mathbf{y}))_i = \ln(y_i), \quad i = 1, \dots, d. \quad (5)$$

*Proof.* See, e.g., Campolieti and Makarov [14, page 485-486].  $\square$

Throughout the paper we impose the following assumption on the covariance matrix  $\mathbf{C} \equiv \mathbf{C}_d$  in dependence of the dimension  $d$ .

**Assumption 2.2 (Covariance matrix)** *There is a constant  $C_1 \in [1, \infty)$  not depending on the dimension  $d \in \mathbb{N}$  such that the covariance matrix  $\mathbf{C} \equiv \mathbf{C}_d = ((\mathbf{C}_d)_{i,j})_{i,j=1}^d \in \mathbb{R}^{d \times d}$  defined as in (1) satisfies for every  $i, j = 1, \dots, d$  that*

$$|(\mathbf{C}_d)_{i,j}| \leq C_1. \quad (6)$$

### 2.1.2 Continuous piece-wise affine (CPWA) payoff functions

For any  $d \in \mathbb{N}$ , we consider a payoff function  $h : \mathbb{R}_+^d \rightarrow \mathbb{R}$ , which takes the stock prices  $\mathbf{S}_T \in \mathbb{R}_+^d$  at terminal time  $T$  as input. The option price  $u(t, \mathbf{x}) \in \mathbb{R}$  at time  $t \in [0, T]$  given that the spot price satisfies  $\mathbf{S}_t = \mathbf{x} \in \mathbb{R}_+^d$  is characterized by the following Feynman-Kac formula (see, e.g., [14, Equation (13.33)])

$$u(t, \mathbf{x}) = e^{-r(T-t)} \mathbb{E}[h(\mathbf{S}_T) \mid \mathbf{S}_t = \mathbf{x}] = e^{-r(T-t)} \int_{\mathbb{R}_+^d} h(\mathbf{y}) p(\mathbf{y}, T; \mathbf{x}, t) d\mathbf{y}, \quad (7)$$

where  $p(\cdot, T; \mathbf{x}, t)$  is the transition density formula given in Lemma 2.1. In this paper, we consider payoff functions restricted to the class of continuous piece-wise affine functions<sup>2</sup>. This type of function represents most of the payoff functions seen in financial mathematics literature [53]; see also the examples below.

**Definition 2.3 (CPWA payoff)** *Let  $d \in \mathbb{N}$ . A function  $h : \mathbb{R}_+^d \rightarrow \mathbb{R}$  is a continuous piece-wise affine (CPWA) function if it can be represented as*

$$h(\mathbf{x}) = \sum_{k=1}^K \xi_k \max\{\mathbf{a}_{k,l} \cdot \mathbf{x} + b_{k,l} : l = 1, \dots, I_k\}, \quad (8)$$

where  $K, I_k \in \mathbb{N}$  and  $\xi_k \in \{-1, 1\}$  for  $k = 1, \dots, K$ , and where  $\mathbf{a}_{k,l} \in \mathbb{R}^d$ ,  $b_{k,l} \in \mathbb{R}$  for  $k = 1, \dots, K$ ,  $l = 1, \dots, I_k$ .

<sup>2</sup>In particular, any CPWA function is linearly growing, see, e.g., Lemma 5.2. Hence the PDE (1) has a unique solution.

Throughout the paper we impose the following assumptions on the CPWA payoff function  $h : \mathbb{R}_+^d \rightarrow \mathbb{R}$  in dependence of the dimension  $d$ .

**Assumption 2.4 (CPWA)** *There is a constant  $C_2 \in [1, \infty)$  not depending on the dimension  $d \in \mathbb{N}$  such that the CPWA function  $h : \mathbb{R}_+^d \rightarrow \mathbb{R}$  defined as in (8) satisfies both*

$$\max \{ \|\mathbf{a}_{k,l}\|_\infty, |b_{k,l}| : k = 1, \dots, K, l = 1, \dots, I_k \} \leq C_2 \quad (9)$$

and

$$K \cdot \max\{I_1, \dots, I_K\} \leq C_2 d. \quad (10)$$

**Example 2.5** *The list below contains examples showcasing that many popular payoff functions  $h : \mathbb{R}_+^d \rightarrow \mathbb{R}$  used in finance are CPWA, see also [53, Appendix EC.2]. In the following, we denote  $\mathbf{e}_i$  the  $i$ -th unit vector in  $\mathbb{R}^d$ .*

1. *Call option on the  $i$ -th asset with strike  $\kappa$ : setting  $K = 1$ ,  $\xi_1 = 1$ ,  $I_1 = 2$ ,  $\mathbf{a}_{1,1} = \mathbf{e}_i$ ,  $\mathbf{a}_{1,2} = \mathbf{0}$ ,  $b_{1,1} = -\kappa$ ,  $b_{1,2} = 0$ , we have*

$$h(\mathbf{x}) = \max\{x_i - \kappa, 0\}. \quad (11)$$

2. *Basket call option with weights  $\mathbf{w}$  and strike  $\kappa$ : setting  $K = 1$ ,  $\xi_1 = 1$ ,  $I_1 = 2$ ,  $\mathbf{a}_{1,1} = \mathbf{w}$ ,  $\mathbf{a}_{1,2} = \mathbf{0}$ ,  $b_{1,1} = -\kappa$ ,  $b_{1,2} = 0$ , we have*

$$h(\mathbf{x}) = \max\{\mathbf{w} \cdot \mathbf{x} - \kappa, 0\}. \quad (12)$$

3. *Spread call option: using setting 2., but by replacing  $\mathbf{a}_{1,1}$  with  $\mathbf{a}_{1,1} = \sum_{i \in \mathcal{I}} \mathbf{e}_i - \sum_{j \in \mathcal{I}'} \mathbf{e}_j$  for  $\mathcal{I}, \mathcal{I}' \subset \{1, \dots, d\}$  and  $\mathcal{I} \cap \mathcal{I}' = \emptyset$ , we have*

$$h(\mathbf{x}) = \max \left\{ \sum_{i \in \mathcal{I}} x_i - \sum_{j \in \mathcal{I}'} x_j - \kappa, 0 \right\}. \quad (13)$$

4. *Call-on-max option with strike  $\kappa$ : setting  $K = 1$ ,  $\xi_1 = 1$ ,  $I_1 = d + 1$ ,  $\mathbf{a}_{1,j} = \mathbf{e}_j$ ,  $b_{1,j} = -\kappa$  for all  $j = 1, \dots, d$ ,  $\mathbf{a}_{1,d+1} = \mathbf{0}$ ,  $b_{1,d+1} = 0$ , we have*

$$h(\mathbf{x}) = \max\{x_1 - \kappa, \dots, x_d - \kappa, 0\}. \quad (14)$$

5. *Call-on-min option with strike  $\kappa$ : setting  $K = 2$ ,  $\xi_1 = 1$ ,  $\xi_2 = -1$ ,  $I_1 = d$ ,  $I_2 = d + 1$ ,  $\mathbf{a}_{1,j} = \mathbf{a}_{2,j} = -\mathbf{e}_j$ ,  $b_{1,j} = b_{2,j} = \kappa$  for all  $j = 1, \dots, d$ ,  $\mathbf{a}_{1,d+1} = \mathbf{0}$ ,  $b_{1,d+1} = 0$ , we have*

$$h(\mathbf{x}) = \max\{\kappa - x_1, \dots, \kappa - x_d, 0\} - \max\{\kappa - x_1, \dots, \kappa - x_d\}. \quad (15)$$

6. *Best-of-call option with strikes  $\kappa_1, \dots, \kappa_d$ : setting  $K = 1$ ,  $\xi_1 = 1$ ,  $I_1 = d + 1$ ,  $\mathbf{a}_{1,j} = \mathbf{e}_j$ ,  $b_{1,j} = -\kappa_j$  for all  $j = 1, \dots, d$ ,  $\mathbf{a}_{1,d+1} = \mathbf{0}$ ,  $b_{1,d+1} = 0$ , we have*

$$h(\mathbf{x}) = \max\{x_1 - \kappa_1, \dots, x_d - \kappa_d, 0\}. \quad (16)$$

We note that all of the above examples satisfy Assumption 2.4 provided that the coefficients  $(\mathbf{a}_{k,l}, b_{k,l})$  are bounded by some constant  $C_2 \in [1, \infty)$  uniformly in the dimension  $d$ , c.f. (9).

## 2.2 Brief Introduction to Quantum Computing

In this section, we briefly introduce the notions used in quantum computing. A classic reference for this subject is the textbook by Nielsen and Chuang [54].

### 2.2.1 Dirac Bra-Ket notation and tensor products

In this section, we recall the Dirac *bra-ket* notation from quantum mechanics. Let  $\mathcal{H}$  be a finite dimensional complex Hilbert space. A vector  $v \in \mathcal{H}$ , also referred as a *state*, is denoted by the *ket* notation  $|v\rangle$ . The inner product of two vectors  $v, w \in \mathcal{H}$  is denoted by the *bra-ket* notation  $\langle v|w\rangle := \langle v, w\rangle \in \mathbb{C}$ . Elements  $u \in \mathcal{H}^*$  of the dual space  $\mathcal{H}^*$  are denoted by the *bra* notation  $\langle u|$ . The action of the dual vector  $u \in \mathcal{H}^*$  on a vector  $v \in \mathcal{H}$  is also denoted by the *bra-ket* notation  $\langle u|v\rangle$ . The action of a linear operator  $A : \mathcal{H} \rightarrow \mathcal{H}$  on a vector  $|v\rangle$  is denoted by  $A|v\rangle$ . The operator  $A$  acts on dual vectors  $\langle u| \in \mathcal{H}^*$  by the rule  $(\langle u|A)|v\rangle := \langle u|(A|v\rangle) := \langle u, Av\rangle$  for all  $v \in \mathcal{H}$  which is also denoted by  $\langle u|A|v\rangle$ . A special case is the *expectation value* of an operator  $A$  on a normalized state, i.e. a state  $|\psi\rangle$  satisfying  $\langle \psi|\psi\rangle = 1$ , which is denoted by  $\langle A\rangle := \langle \psi|A|\psi\rangle$ . The linear operator given by the *outer product* of two vectors  $v, u \in \mathcal{H}$  is denoted by  $|v\rangle\langle u| : \mathcal{H} \rightarrow \mathcal{H}$ , whose action on a vector  $|x\rangle \in \mathcal{H}$  is defined by  $(|v\rangle\langle u|)|x\rangle := \langle u|x\rangle |v\rangle$ .

For the Hilbert space  $\mathcal{H} = \mathbb{C}^2$ , we consider the  $n$ -fold tensor product Hilbert space  $\mathcal{H}^{\otimes n} := \mathcal{H} \otimes \dots \otimes \mathcal{H} \simeq \mathbb{C}^{2^n}$ . We denote a state  $\psi \in \mathcal{H}^{\otimes n}$  by  $|\psi\rangle_n$ , where the subscript  $n$  emphasizes the  $(\log_2)$ -dimension of the tensor product Hilbert space  $\mathcal{H}^{\otimes n}$ . We use the orthonormal basis  $\mathcal{B}_n = \{|i\rangle_n : i = (i_1, i_2, \dots, i_n) \in \{0, 1\}^n\} \subset \mathbb{C}^{2^n}$ , where  $|i\rangle_n := |i_1\rangle \otimes |i_2\rangle \otimes \dots \otimes |i_n\rangle := |i_1\rangle |i_2\rangle \dots |i_n\rangle$ ,  $i \in \{0, 1\}^n$ . The basis  $\mathcal{B}_n$  is referred as the *computational basis* in the literature.

We illustrate some examples of tensor product of vectors and operators for  $n = 2$  using the standard matrix-vector notation. The standard orthonormal basis  $\{|0\rangle, |1\rangle\} \subset \mathcal{H} = \mathbb{C}^2$  is given by

$$|0\rangle := \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle := \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \quad (17)$$

The standard orthonormal basis for  $\mathcal{H} \otimes \mathcal{H}$  is given by the tensor product basis,

$$\mathcal{B}_2 = \{ |i\rangle_2 = |i_1\rangle \otimes |i_2\rangle : i_1, i_2 \in \{0, 1\} \} = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \right\}. \quad (18)$$

Let  $|v\rangle = (v_1, v_2)^\top, |u\rangle = (u_1, u_2)^\top \in \mathcal{H}$ . The tensor product of these two vectors is given by

$$|v\rangle \otimes |u\rangle = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \otimes \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} v_1 u_1 \\ v_1 u_2 \\ v_2 u_1 \\ v_2 u_2 \end{bmatrix}. \quad (19)$$

The Kronecker product of two square matrices  $A, B \in \mathbb{C}^{2 \times 2}$  is given by

$$A \otimes B = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \otimes \begin{bmatrix} b_1 & b_2 \\ b_3 & b_4 \end{bmatrix} = \begin{bmatrix} a_1 B & a_2 B \\ a_3 B & a_4 B \end{bmatrix} = \begin{bmatrix} a_1 b_1 & a_1 b_2 & a_2 b_1 & a_2 b_2 \\ a_1 b_3 & a_1 b_4 & a_2 b_3 & a_2 b_4 \\ a_3 b_1 & a_3 b_2 & a_4 b_1 & a_4 b_2 \\ a_3 b_3 & a_3 b_4 & a_4 b_3 & a_4 b_4 \end{bmatrix}. \quad (20)$$

Furthermore, one can deduce from (20) and (19) that the following (multi)-linearity relation holds

$$(A \otimes B)(|v\rangle \otimes |u\rangle) = A|v\rangle \otimes B|u\rangle. \quad (21)$$

Similar rules for tensor and Kronecker products in  $\mathcal{H}^{\otimes n}$ ,  $n \geq 3$  can be deduced inductively. Note that the tensor product of vectors and operators *may not be commutative*.

### 2.2.2 Qubits, quantum gates, and quantum circuits

A classical computing bit,  $x \in \{0, 1\}$ , represents a basic unit of computing information. The *qubit* (quantum bit) is the generalization of the classical bit, and it represents a unit of quantum information. Quantum mechanics allow the qubit to be in a state of *superposition*. For example, a single qubit may represent a normalized vector  $|\psi\rangle$  in superposition of the state  $|0\rangle$  and  $|1\rangle$  simultaneously, and we may express  $|\psi\rangle$  as

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle, \quad \text{with } |\alpha|^2 + |\beta|^2 = 1, \text{ and } \alpha, \beta \in \mathbb{C}. \quad (22)$$

For  $n \geq 2$ , an arbitrary  $n$ -qubit state  $|\psi\rangle_n$  is represented by a normalized vector in  $\mathbb{C}^{2^n}$  which can be described by a  $\mathbb{C}$ -linear combination in the computational basis  $\mathcal{B}_n$ , i.e.

$$|\psi\rangle_n = \sum_{i \in \{0,1\}^n} \alpha_i |i\rangle_n, \quad \text{with } \sum_{i \in \{0,1\}^n} |\alpha_i|^2 = 1. \quad (23)$$

The coefficients  $\alpha_i \in \mathbb{C}$  in (23) are referred as *probability amplitudes* (or simply *amplitudes*) due to the fact that for each  $|i\rangle_n \in \mathcal{B}_n$  we have that the square amplitude  $|\alpha_i|^2 = |\langle i|\psi\rangle|^2$  is the probability of observing that the state  $|\psi\rangle_n$  collapses during a projective measurement to the state  $|i\rangle_n$ , according to *Born's rule*; see, e.g., [48, Chapter 2.5].

Both classical bits and qubits are manipulated using circuit *gates*. The classical bits on a computer are processed by a series of *logical/boolean gates*, e.g. NOT, OR, and AND gates. The input bits are connected by wires and gates to the output bits. A *circuit diagram* is often accompanied to show how these gates are placed. In contrast, the qubits on a quantum computer are manipulated by a sequence of elementary *quantum gates*; these quantum gates act on either one or two qubits at a time. The state evolution of the qubits, according to the axioms of quantum mechanics, is unitary. Thus, quantum gates are unitary operators, represented by *unitary matrices* in  $\mathcal{U}(2^n) \subset \mathbb{C}^{2^n \times 2^n}$ , where  $n$  is the number of qubits acted on by the quantum gate. A *quantum circuit* is a finite sequence of composition of quantum gates and wires (wires represent the identity operator). The fact that the product of unitary matrices is again a unitary matrix implies that the quantum circuit performs a unitary operation in the Hilbert space  $\mathcal{H}^{\otimes n}$ . Similar to classical computer boolean circuits, quantum circuits are presented by a circuit model diagram, which specifies where and how the quantum gates and wires are placed. For example, Figure 1 shows the quantum circuit for the Toffoli gate (see Example 2.18), which corresponds to the classical boolean gate NOT with two control bits.

Next, we provide some examples of quantum circuit gates commonly used in the literature which allows us to describe an *algebraic definition* for a quantum circuit; see Definition 2.14.

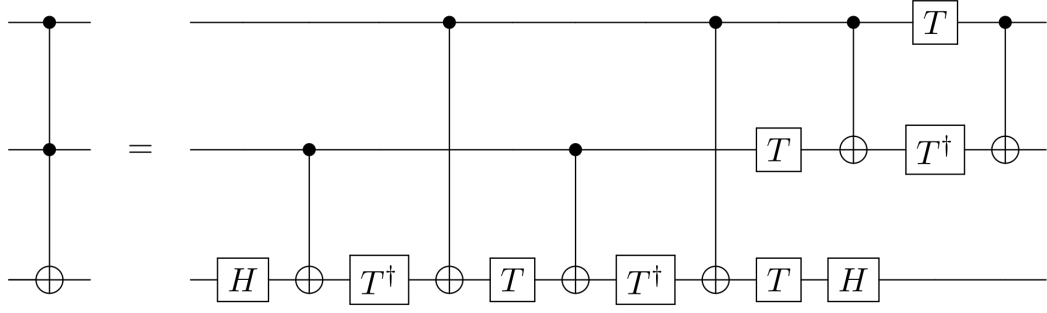


Figure 1: Implementation of the Toffoli gate as a quantum circuit using single-qubit gates and CNOT gates.<sup>3</sup>

**Example 2.6 (Pauli transformation gates)** The one-qubit gates corresponding to the Pauli transformations are

$$I := \sigma_I := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad X := \sigma_x := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y := \sigma_y := \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad Z := \sigma_z := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \quad (24)$$

These gates satisfy the relations  $I = -i\sigma_x\sigma_y\sigma_z$ ,  $I = \sigma_x^2 = \sigma_y^2 = \sigma_z^2$ , where  $i := \sqrt{-1}$ , and they span the group of  $2 \times 2$ -unitary matrices.

**Example 2.7 (Hadamard gate)** The Hadamard gates is given by

$$H := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \quad (25)$$

It maps the computation basis states to a uniform superposition, i.e.  $H|0\rangle := |+\rangle := \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$  and  $H|1\rangle := |-\rangle := \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ .

**Example 2.8 (Phase shift gates)** For  $\phi \in [0, 2\pi]$ , we define the family of phase shift gates

$$P(\phi) := \begin{bmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{bmatrix}. \quad (26)$$

Special cases include the Pauli Z gate, S gate, and T gate:

$$Z := P(\pi) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad S := P(\pi/2) = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, \quad T := P(\pi/4) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}. \quad (27)$$

**Example 2.9 (Rotation gates)** For  $\theta \in [0, 4\pi]$ , we introduce the three rotation gates:

$$R_x(\theta) := \exp(-i\theta X/2) = \begin{bmatrix} \cos(\frac{\theta}{2}) & -i\sin(\frac{\theta}{2}) \\ -i\sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{bmatrix}, \quad (28)$$

$$R_y(\theta) := \exp(-i\theta Y/2) = \begin{bmatrix} \cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2}) \\ \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{bmatrix}, \quad (29)$$

$$R_z(\theta) := \exp(-i\theta Z/2) = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}. \quad (30)$$

**Example 2.10 (Controlled-NOT gate)** The controlled-NOT gate has the following matrix representation

$$\text{CNOT} := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (31)$$

The CNOT acts on two qubits: the first qubit is the control qubit and the second qubit is the target qubit, which is flipped if the control qubit is in state  $|1\rangle$ .

**Example 2.11 (Swap gate)** The swap gate has the following matrix representation

$$\text{SWAP} := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (32)$$

<sup>3</sup>This figure can be found on [https://commons.wikimedia.org/wiki/File:Qcircuit\\_ToffolifromCNOT.svg](https://commons.wikimedia.org/wiki/File:Qcircuit_ToffolifromCNOT.svg)

**Definition 2.12 (Elementary quantum gate set)** We call any element of the set of 1-qubit and 2-qubits quantum gates  $\mathbb{G} \subset \mathbb{C}^{2 \times 2} \cup \mathbb{C}^{2^2 \times 2^2}$  defined by

$$\mathbb{G} := \{X, Y, Z, H, S, T, R_x(\theta), R_y(\theta), R_z(\theta), P(\phi) : \theta \in (0, 4\pi), \phi \in (0, 2\pi)\} \cup \{\text{CNOT}, \text{SWAP}\} \quad (33)$$

as elementary gates; (see Examples 2.6–2.11).

**Remark 2.13 (Universality of  $\mathbb{G}$ )** We note that the set of quantum gates  $\mathbb{G}$  in (33) is universal in the sense of the Solovay-Kitaev theorem [54, Appendix 3]. Moreover, the set  $\mathbb{G}$  consists of quantum gates used in practical quantum computing softwares, such as IBM's Qiskit [59] and Google's Cirq [22].

**Definition 2.14 (Quantum circuit)** Let  $n, M, L \in \mathbb{N}$ . A quantum circuit  $\mathcal{Q}$  acting on  $n$  qubits is a  $2^n \times 2^n$  unitary matrix of the form<sup>4</sup>

$$\mathcal{Q} = \prod_{l=1}^L (G_{l,1} \otimes G_{l,2} \otimes \cdots \otimes G_{l,n_l}) \in \mathcal{U}(2^n), \quad (34)$$

where  $(G_{l,1}, G_{l,2}, \dots, G_{l,n_l})_{l=1}^L \subset \mathbb{G} \cup \{I_2\}$ , and  $\prod_{m=1}^{n_l} \dim(G_{l,m}) = 2^n$  for all  $l = 1, \dots, L$ . We define the following quantum circuit complexities<sup>5</sup>:

- $M$  = number of elementary quantum gates used to construct quantum circuit  $\mathcal{Q}$ , i.e.

$$M := \sum_{l=1}^L \sum_{j=1}^{n_l} \mathbb{1}_{\mathbb{G}}(G_{l,j}), \quad (35)$$

- $n$  = number of qubits used in quantum circuit  $\mathcal{Q}$ , and
- $L$  = depth of quantum circuit  $\mathcal{Q}$ .

**Remark 2.15 (Depth)** For any quantum circuit  $\mathcal{Q}$ , we may bound its depth complexity by the number of gates in the quantum circuit. In this paper, we focus only on the number of elementary gates and qubits used in constructing quantum circuits.

**Remark 2.16 (Inverse  $\mathcal{Q}^\dagger$ )** We note that the set of elementary quantum gates defined in (33) is closed under matrix inversion. Moreover, for any unitary matrix  $\mathcal{Q}$ , the matrix inverse  $\mathcal{Q}^{-1}$  is also the conjugate transpose  $\mathcal{Q}^\dagger$ . Thus, for any quantum circuit  $\mathcal{Q}$  satisfying Definition 2.14, we can construct the quantum circuit  $\mathcal{Q}^\dagger$  representing its inverse by

$$\mathcal{Q}^\dagger = \prod_{l=0}^{L-1} (G_{L-l,1}^\dagger \otimes G_{L-l,2}^\dagger \otimes \cdots \otimes G_{L-l,n_{L-l}}^\dagger). \quad (36)$$

**Remark 2.17 (Ancilla qubits)** In most quantum circuits, auxiliary qubits are used as additional memory to perform necessary quantum computations but may not be used for the output. One calls these qubits ancilla qubits or simply ancillas and denotes them by  $|\text{anc}\rangle_\star$ , where the subscript  $\star$  usually indicates in the literature that the amount of ancillas used are not specified precisely. The complexity of a quantum circuit is usually described by the number of elementary quantum gates used, and the number of qubits and ancilla qubits used. For simplicity, we count both qubits and ancilla qubits together as the number of qubits used in a circuit.

**Example 2.18 (Toffoli gate as a quantum circuit)** The classical AND gate, which takes a pair of bits  $x, y \in \{0, 1\}$  as input, returns the bit  $\text{AND}(x, y) := xy$  as output. This classical gate is not a reversible operation. Hence the AND gate cannot be represented by a unitary matrix. However, we can implement the quantum version of this classical logic gate by using 3 qubits. The operation  $|a\rangle|b\rangle|c\rangle \mapsto |a\rangle|b\rangle|c + ab \bmod 2\rangle$  is reversible. This operation is implemented as a quantum gate called the Toffoli gate (also referred as the CCNOT gate). The Toffoli gate has the matrix representation

$$\text{CCNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \quad (37)$$

For  $a, b, c \in \{0, 1\}$ , we have

$$\text{CCNOT} : |a\rangle|b\rangle|c\rangle \mapsto |a\rangle|b\rangle|c + ab \bmod 2\rangle. \quad (38)$$

<sup>4</sup>Note that for any  $n \in \mathbb{N}$  and any  $\mathcal{Q}_1, \dots, \mathcal{Q}_L \in \mathcal{U}(2^n)$ , their product is defined by  $\prod_{l=1}^L \mathcal{Q}_l := \mathcal{Q}_L \mathcal{Q}_{L-1} \cdots \mathcal{Q}_2 \mathcal{Q}_1 \in \mathcal{U}(2^n)$ .

<sup>5</sup>The indicator function  $\mathbb{1}_S$  for a non-empty subset  $S \subset H$  is the unique function satisfying  $\mathbb{1}_S : H \rightarrow \{0, 1\}$  such that  $\mathbb{1}_S(x) = 1$  if  $x \in S$  and  $\mathbb{1}_S(x) = 0$  if  $x \notin S$ .



Here one refers  $|a\rangle$  as the control qubit,  $|b\rangle$  the target qubit, and  $|c\rangle$  the output qubit.

The Toffoli gate is constructed as a quantum circuit on three qubits using the Hadamard gates,  $T$  gates, and CNOT gates; see also Figure 1 for the corresponding circuit diagram. The quantum circuit for the Toffoli gate uses 15 elementary gates in total.

### 2.2.3 Quantum measurements

We use the following definition to describe the quantum measurement of any arbitrary normalized state characterized by  $n$ -qubits. For further details, we refer to, e.g., [48, Chapter 2.5].

**Definition 2.19 (Quantum measurement in the computational basis)** Let  $n \in \mathbb{N}$  and let  $\mathcal{B}_n := \{|i\rangle_n := |i_1\rangle \otimes \cdots \otimes |i_n\rangle \mid i = (i_1, \dots, i_n) \in \{0, 1\}^n\}$  be the computational basis of  $\mathbb{C}^{2^n}$ . For each  $|i\rangle_n \in \mathcal{B}_n$  and for any normalized state  $|\psi\rangle_n$ , we define a projector  $M_i := |i\rangle_n \langle i|_n : \mathbb{C}^{2^n} \rightarrow \mathbb{C}^{2^n}$  which is defined by  $(|i\rangle_n \langle i|_n) |\psi\rangle_n := \langle i | \psi \rangle_n |i\rangle_n$ . A measurement of a normalized state  $|\psi\rangle_n$  in the computational basis induces a discrete probability space  $(\Omega, \mathcal{F}, \mathbb{P}_\psi)$ , where  $\Omega = \{0, 1\}^n$  is the sample space,  $\mathcal{F} = 2^\Omega$  is the  $\sigma$ -algebra, and  $\mathbb{P}_\psi : \mathcal{F} \rightarrow [0, 1]$  is the discrete probability measure defined by

$$\mathbb{P}_\psi(\{i\}) := \langle \psi | M_i^\dagger M_i | \psi \rangle_n = |\langle \psi | i \rangle_n|^2, \quad i \in \Omega. \quad (39)$$

### 2.3 Quantum amplitude estimation algorithms

In this section, we briefly review quantum algorithms for solving the quantum amplitude estimation (QAE) problem. Given an unitary operator  $\mathcal{A}$  acting on  $n + 1$  qubits, defined by

$$\mathcal{A} |0\rangle_n |0\rangle = \sqrt{1-a} |\psi_0\rangle_n |0\rangle + \sqrt{a} |\psi_1\rangle_n |1\rangle, \quad (40)$$

where the so-called bad state is  $|\psi_0\rangle_n |0\rangle$  and the good state is  $|\psi_1\rangle_n |1\rangle$ , Brassard et al. introduced in [12] the amplitude estimation problem where the goal is to estimate the unknown amplitude  $a \in [0, 1]$ , which is the probability of measuring the good state  $|\psi_1\rangle_n |1\rangle$  according to Born's rule. Let  $a = \sin^2(\theta_a)$  for some  $\theta_a \in [0, \frac{\pi}{2}]$  so that we can rewrite (40) as

$$\mathcal{A} |0\rangle_n |0\rangle = \cos(\theta_a) |\psi_0\rangle_n |0\rangle + \sin(\theta_a) |\psi_1\rangle_n |1\rangle. \quad (41)$$

To achieve a quantum speed-up, they introduced in [12] the amplitude amplification operator (also known as the Grover operator)

$$\mathcal{Q} := \mathcal{A} \mathcal{S}_0 \mathcal{A}^\dagger \mathcal{S}_{\psi_0}, \quad (42)$$

where  $\mathcal{S}_0 := I_2^{\otimes n+1} - 2|0\rangle_{n+1} \langle 0|_{n+1}$ , and  $\mathcal{S}_{\psi_0} := I_2^{\otimes n} \otimes Z$ . Note that for every  $k \in \mathbb{N}$ , it holds that

$$\mathcal{Q}^k \mathcal{A} |0\rangle_n |0\rangle = \cos((2k+1)\theta_a) |\psi_0\rangle_n |0\rangle + \sin((2k+1)\theta_a) |\psi_1\rangle_n |1\rangle, \quad (43)$$

(c.f. [12, Section 2]). We observe that measuring (43) boosts the probability of obtaining the good state to  $\sin^2((2k+1)\theta_a)$  which is larger than  $\sin^2(\theta_a)$  when measuring (41) directly, provided  $\theta_a$  is sufficiently small so that  $(2k+1)\theta_a \leq \frac{\pi}{2}$ . Using quantum Fourier transform and a number of multi-controlled operators for  $\mathcal{Q}^k$ , Brassard et al. designed the QAE algorithm [12, Algorithm(Est\_Amp)] to estimate  $a$  with high probability using only  $O(\varepsilon^{-1})$  queries of  $\mathcal{A}$  (see [12, Theorem 12]). It is noted that Brassard et al.'s algorithm enables a quadratic speed-up for many approximation problems which are solved classically by Monte Carlo simulations under the assumption that the corresponding distribution can be uploaded in rotated form (41). However, due to difficulties in implementing large number of controlled unitary operators as well as the quantum Fourier transform (QFT) operator on quantum computers, several variants of the QAE algorithm without using QFT have been proposed recently; see e.g., [1, 31, 33, 47, 52, 57, 60, 70, 73, 75]. In this paper, we use the modified iterative quantum amplitude estimation algorithm (Modified IQAE) [29, Algorithm 1] introduced recently by Fukuzawa et al. [29], which is a modification of the IQAE algorithm presented by Suzuki et al. [70]. In brief, the Modified IQAE algorithm consists of several rounds where for each round  $i$ , the algorithm maintains a confidence interval  $[\theta_l^{(i)}, \theta_u^{(i)}]$  so that  $\theta_a$  lies inside this interval with a certain probability. The confidence interval is narrowed in each subsequent round until the terminating condition  $\theta_u - \theta_l < 2\varepsilon$  for prespecified  $\varepsilon \in (0, 1)$  is satisfied. The return output of the Modified IQAE algorithm is the confidence interval  $[a_l, a_u]$  for  $a$ , where  $a_l := \sin^2(\theta_l)$  and  $a_u := \sin^2(\theta_u)$ . The following statement is a direct rephrase of the main results in [29].

**Proposition 2.20** Let  $\alpha, \varepsilon \in (0, 1)$  and let  $\mathcal{A}$  be an  $(n+1)$ -qubit quantum circuit satisfying

$$\mathcal{A} |0\rangle_n |0\rangle = \sqrt{1-a} |\psi_0\rangle_n |0\rangle + \sqrt{a} |\psi_1\rangle_n |1\rangle \quad (44)$$

where  $n \in \mathbb{N}$ ,  $|\psi_0\rangle_n, |\psi_1\rangle_n$  are normalized states,  $a \in [0, 1]$ , and where  $\mathcal{A}$  can be constructed with  $N_{\mathcal{A}} \in \mathbb{N}$  number of elementary gates. Then, the following holds.

1. The Modified IQAE Algorithm [29, Algorithm 1] outputs a confidence interval  $[a_l, a_u]$  that satisfies

$$a \notin [a_l, a_u], \quad \text{with probability at most } \alpha, \quad (45)$$

where  $0 \leq a_u - a_l < 2\varepsilon$ . In particular, the estimator  $\hat{a} := \frac{a_u + a_l}{2}$  of  $a$  satisfies

$$|a - \hat{a}| < \varepsilon, \quad \text{with probability at least } 1 - \alpha, \quad (46)$$

2. the Modified IQAE Algorithm uses at most

$$\frac{62}{\varepsilon} \ln \left( \frac{21}{\alpha} \right) \quad (47)$$

applications of  $\mathcal{A}$ , and

3. the Modified IQAE Algorithm uses  $n + 1$  qubits and requires at most

$$\frac{\pi}{4\varepsilon} (8n^2 + 23 + N_{\mathcal{A}}) \quad (48)$$

number of elementary gates.

*Proof.* Item 1. and Item 2. are proven in [29, Theorem 3.1] and [29, Lemma 3.7], respectively. For Item 3., we note that [29, Algorithm 1 Modified IQAE] uses quantum circuits  $\mathcal{Q}^k \mathcal{A}$  which is defined by (43), where  $k \in \mathbb{N}$ . Let us construct the operator  $\mathcal{Q}$  (c.f. (42)), as outlined in [62, Section 3]. We note that the operator  $\mathcal{S}_{\psi_0}$  be constructed using one  $Z$  gate (c.f. Example 2.6). By direct computation (or see [62, Figure 5]), the operator  $\mathcal{S}_0$  satisfy the identity

$$\mathcal{S}_0 = X^{\otimes(n+1)} (I_2^{\otimes n} \otimes H) C^n(X) (I_2^{\otimes n} \otimes H) X^{\otimes(n+1)}, \quad (49)$$

where  $C^n(X)$  is the generalized version of Toffoli gate, which uses the first  $n$ -qubits for control. The multi-control gate  $C^n(X)$  is constructed as a quantum circuit in [66, Theorem 2] using  $2n^2 - 6n + 5$  controlled  $X$ -rotation gates  $CR_x(\theta)$ , where each  $CR_x(\theta)$  gate can be constructed with 4 elementary gates by the following definition

$$CR_x(\theta) = (I_2 \otimes R_x(\frac{\theta}{2})) \text{CNOT}(I_2 \otimes R_x(\frac{\theta}{2})) \text{CNOT}, \quad (50)$$

see also the proof of Lemma 4.15. Thus, by (49), the operator  $\mathcal{S}_0$  can be constructed using

$$(n+1) + 1 + 4(2n^2 - 6n + 5) + 1 + (n+1) \leq 8n^2 + 22 \quad (51)$$

elementary gates. Finally, since the set of elementary gates is closed under inversion, the number of elementary gates used to construct  $\mathcal{A}^\dagger$  is the same as the number of elementary gates used for  $\mathcal{A}$ . Hence, the number of elementary gates used to construct  $\mathcal{Q}$  is at most

$$N_{\mathcal{A}} + 1 + N_{\mathcal{A}} + (8n^2 + 22) = 8n^2 + 23 + 2N_{\mathcal{A}}. \quad (52)$$

Next, by [29, Lemma 3.1], the integer  $k$  satisfies the bound  $2k + 1 \leq \frac{\pi}{4\varepsilon}$ . Thus, the number of elementary gates used to construct the operator  $\mathcal{Q}^k \mathcal{A}$  is at most

$$\begin{aligned} & k(8n^2 + 23 + 2N_{\mathcal{A}}) + N_{\mathcal{A}} \\ &= k(8n^2 + 23) + (2k + 1)N_{\mathcal{A}} \\ &\leq (2k + 1)(8n^2 + 23 + N_{\mathcal{A}}) \\ &\leq \frac{\pi}{4\varepsilon} (8n^2 + 23 + N_{\mathcal{A}}). \end{aligned} \quad (53)$$

□

**Remark 2.21** Let us remark the following on the Modified IQAE algorithm [29].

1. The total number of rounds  $t \in \mathbb{N}$  in [29, Algorithm 1 Modified IQAE] is bounded by  $\log_3(\frac{\pi}{4\varepsilon})$ , see [29, Section 3.1]. For each round  $i = 1, \dots, t$ , the quantum circuit  $\mathcal{Q}^{k_i} \mathcal{A}$  is prepared on a quantum computer, where each  $k_i \in \mathbb{N}$  are found recursively by using the subroutine [29, Algorithm 2, FindNextK]. Note that each of these quantum circuits require the same number of qubits as with the quantum circuit  $\mathcal{A}$ . Moreover, each  $k_i$  satisfy  $2k_i + 1 \leq \frac{\pi}{4\varepsilon}$ , [29, Lemma 3.1].
2. Since the number  $k_t$  is the maximum among the  $k_i$ 's, we infer that [29, Algorithm 1, Modified IQAE] requires the number of elementary gates used to construct the quantum circuit  $\mathcal{Q}^{k_t} \mathcal{A}$  on a quantum computer in order to run the Modified IQAE algorithm, which can be bounded by (48).

3. The query complexity (i.e. number of applications) of  $\mathcal{A}$  in Proposition 2.20 is defined to be the number of times the operator  $\mathcal{Q}$  is applied in the algorithm, which is

$$\sum_{i=1}^t k_i N_i, \quad (54)$$

where  $N_i$  is the number of measurements made on  $\mathcal{Q}^{k_i} \mathcal{A} |0\rangle_n |0\rangle$  in round  $i$ . Hence, the query complexity of  $\mathcal{A}$  can be interpreted as the computational running time for the Modified IQAE algorithm. It was shown in [29, Lemma 3.7] that this number is bounded by (47).

4. We emphasize that other versions of the QAE (or IQAE) algorithm offer essentially the same query complexity (i.e.  $O(\frac{1}{\varepsilon} \ln(\frac{1}{\alpha}))$  up to logarithmic factors of  $\varepsilon^{-1}$ ). In this paper, we chose the Modified IQAE for the quantum amplitude estimation subroutine in Algorithm 1 since the bounds on the query complexities in [29] were explicit. We refer the reader to Section 3.2 in [46] for a detailed comparison for the query complexities of the different QAE algorithms that are available in the literature.

## 2.4 Algorithm 1 and main result

In this section, we first present our quantum Monte Carlo algorithm named Algorithm 1 to solve Black-Scholes PDEs (1) with corresponding CPWA payoff function (8). Moreover, we then outline Algorithm 1 and present our main result in Theorem 1, namely a convergence and complexity analysis of our algorithm.

---

**Algorithm 1:** Quantum algorithm for solving Black-Scholes PDEs with CPWA payoff functions

---

**Input:**  $\varepsilon \in (0, 1)$ ,  $\alpha \in (0, 1)$ ,  $d \in \mathbb{N}$ ,  $r, T \in (0, \infty)$ ,  $(t, \mathbf{x}) \in [0, T) \times \mathbb{R}_+^d$ , covariance matrix  $\mathbf{C}_d \in \mathbb{R}^{d \times d}$ , and CPWA function

$$\mathbb{R}_+^d \ni \mathbf{x} \mapsto h(\mathbf{x}) = \sum_{k=1}^K \xi_k \max\{\mathbf{a}_{k,l} \cdot \mathbf{x} + b_{k,l} : l = 1, \dots, I_k\} \in \mathbb{R}$$

**Output:**  $\tilde{U}_{t,\mathbf{x}} \in \mathbb{R}$

- 1 Set  $C_1, C_2, C_3 \in [1, \infty)$  to be the constants given by Assumption 2.2, Assumption 2.4, and Assumption 4.16, respectively.

- 2 Set

$$n_1 := \lceil n_{1,d,\varepsilon} \rceil, \quad n_2 := 1 + \lceil \log_2(C_2) \rceil, \quad m_1 := \lceil m_{1,d,\varepsilon} \rceil, \quad m_2 := \lceil m_{2,d,\varepsilon} \rceil,$$

where  $n_{1,d,\varepsilon}, m_{1,d,\varepsilon}, m_{2,d,\varepsilon}$  are defined in (315)-(317) in Proposition 5.12.

- 3 Set

$$N = (202), \quad \gamma = (321), \quad \mathfrak{s} = (322), \quad \text{and} \quad \tilde{\mathbf{a}}_{n_2, m_2, k, l}, \tilde{b}_{n_2, m_2, k, l} = (267) \text{ for } k = 1, \dots, K, l = 1, \dots, I_k.$$

- 4 Construct probability distribution quantum circuit  $\mathcal{P} \equiv \mathcal{P}_{d,\varepsilon}$  using Assumption 4.16 (with  $n \leftarrow n_1$ ,  $m \leftarrow m_1$ ,  $\varepsilon \leftarrow \frac{\varepsilon}{6C_2^2 d^2 2^{n_1+1}}$  in the notation of Assumption 4.16).

- 5 Construct CPWA payoff with rotation quantum circuit  $\mathcal{R}_h$  given by Proposition 4.24 (with  $s \leftarrow \mathfrak{s}$ ,  $\mathbf{a}_{k,l,j} \leftarrow E_{n_2, m_2}(\tilde{\mathbf{a}}_{n_2, m_2, k, l, j})$ ,  $b_{k,l} \leftarrow E_{n_2, m_2}(\tilde{b}_{n_2, m_2, k, l})$  for  $k = 1, \dots, K$ ,  $l = 1, \dots, I_k$ ,  $j = 1, \dots, d$  in the notation of Proposition 4.24).

- 6 Construct the quantum circuit  $\mathcal{A} = \mathcal{R}_h(\mathcal{P} \otimes I_2^{\otimes (N-d(n_1+m_1))})$  using the quantum circuits  $\mathcal{R}_h$  and  $\mathcal{P}$ .

- 7 Set  $\hat{a} = \frac{a_u + a_l}{2}$  using the output  $[a_l, a_u]$  from the modified iterative quantum amplitude estimation algorithm [29, Algorithm 1 Modified IQAE] (with  $\varepsilon \leftarrow \frac{\varepsilon}{12}$ ,  $\alpha \leftarrow \alpha$ ,  $N_{\text{shots}} \leftarrow 1$ , and  $\mathcal{A} \leftarrow \mathcal{A}$  in the notation of [29, Algorithm 1]).

- 8 Return  $\tilde{U}_{t,\mathbf{x}} := \mathfrak{s}^{-1} \gamma e^{-r(T-t)} (2\hat{a} - 1)$ .
- 

### 2.4.1 Outline of Algorithm 1

The steps of Algorithm 1 can be briefly described into four parts as follows:

1. upload the transition probability function  $p(\cdot, T; \mathbf{x}, t)$  given in (4),
2. upload the CPWA payoff function  $h : \mathbb{R}_+^d \rightarrow \mathbb{R}$  given in (8),
3. apply the Modified IQAE algorithm [29, Algorithm 1] to obtain an estimated amplitude  $\hat{a} \in [0, 1]$ ,
4. rescale the estimated amplitude  $\hat{a}$  to output  $\tilde{U}_{t,\mathbf{x}}$  which approximates  $u(t, \mathbf{x})$  defined in (7).

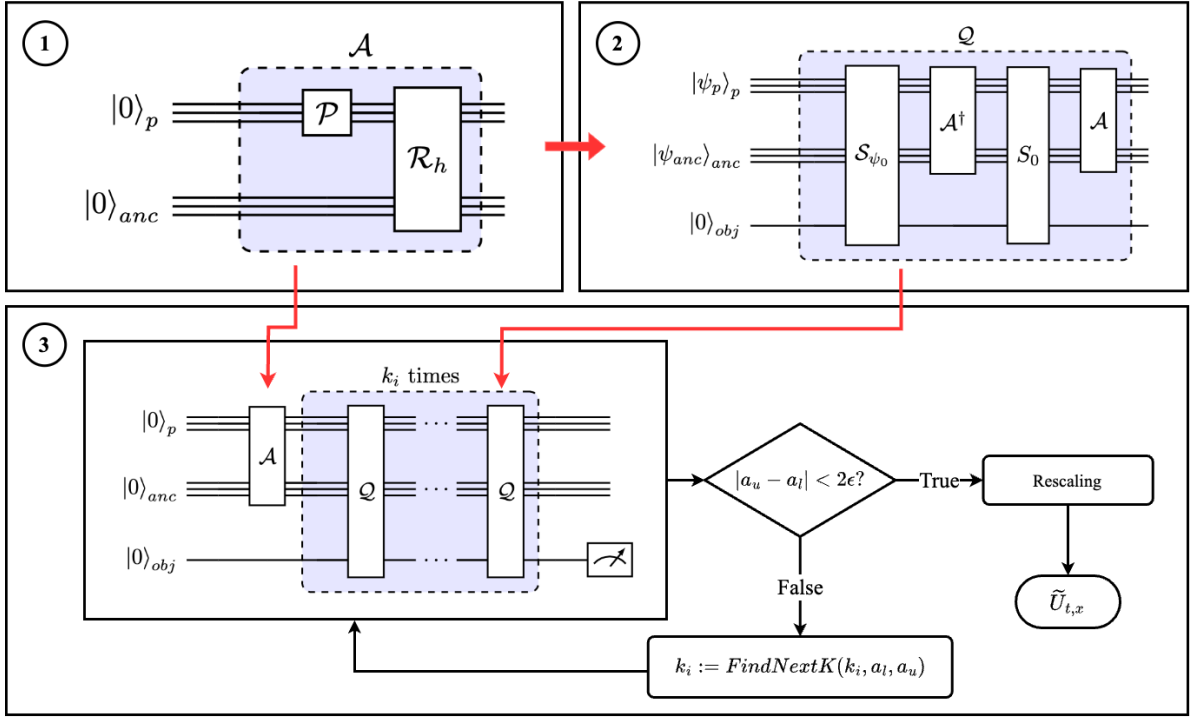


Figure 2: Flowchart of Algorithm 1. (Top left) (1) Construction of the operator  $\mathcal{A}$  using probability distribution operator  $\mathcal{P}$  and CPWA payoff operator  $\mathcal{R}_h$ . (Top right) (2) Construction of the Grover operator using the operator  $\mathcal{A}$ , oracle operator  $S_{\psi_0}$  and phase flip operator  $S_0$ . (Bottom) (3) Illustration of the Modified Iterative Quantum Amplitude Estimation algorithm to produce the final estimate  $\tilde{U}_{t,x}$ .

More precisely, in order to apply the Modified IQAE algorithm in step 3, we first need to upload both the probability distribution function and the payoff function as quantum circuits. For the first step, we need to truncate and discretize the distribution function on a grid. The corresponding parameters  $n_1, m_1 \in \mathbb{N}$  encodes the grid  $([-2^{n_1-1}, 2^{n_1-1} - 2^{-m_1}] \cap 2^{-m_1}\mathbb{Z})^d$ , which is the support of the discretized distribution. Then, the truncated, discretized distribution can be uploaded approximately on the quantum computer using the quantum circuit  $\mathcal{P}$ , introduced in Section 4.3. The corresponding parameter  $\gamma$  is needed to renormalize the amplitude coefficients in  $\mathcal{P}|0\rangle_{d(n_1+m_1)}$  to  $[0, 1]$ . We note that step 1 accounts for the truncation error, quadrature error, and distribution loading error in Section 5, see Proposition 5.4, Proposition 5.5, and Proposition 5.10, respectively.

In step 2, we discretize the coefficients  $(\mathbf{a}_{k,l}, b_{k,l})$  of the CPWA payoff function (8) onto a grid. The corresponding parameter  $n_2 \in \mathbb{N}$  encodes the bounds on the coefficients while the parameter  $m_2 \in \mathbb{N}$  encodes the rounding off accuracy level for the coefficients. We construct the quantum circuit  $\mathcal{R}_h$  that encodes the approximated CPWA payoff in a phase amplitude using controlled  $Y$ -rotation gates. The corresponding parameter  $s \equiv s$  indicates the scaling parameter for the rotation circuit  $\mathcal{R}_h$ , see Proposition 4.24. We account for the errors for approximating the payoff function  $h : \mathbb{R}_+^d \rightarrow \mathbb{R}$  and the rotation errors in Proposition 5.8 and Proposition 5.11. The quantum circuit  $\mathcal{A}$  is then defined as the combination (i.e. composition) of the two quantum circuits  $\mathcal{P}$  and  $\mathcal{R}_h$  which enables us to apply quantum amplitude estimation (QAE) algorithms.

In step 3, we apply the Modified IQAE algorithm<sup>6</sup> [29] with  $\mathcal{A}$  to get an output  $\hat{a} \in [0, 1]$ . Then in the last step, since the estimated amplitude  $\hat{a}$  given by the Modified IQAE algorithm is between 0 and 1, we need to rescale this number to approximate the option price  $u(t, \mathbf{x})$ . We account for the QAE error in Proposition 5.12.

<sup>6</sup>We emphasize that the other QAE algorithms can also be used since the query complexities are essentially similar to [29], c.f. Remark 2.21 Item 4.

## 2.4.2 Main Theorem

**Theorem 1** Let  $\varepsilon \in (0, 1)$ ,  $\alpha \in (0, 1)$ ,  $d \in \mathbb{N}$ ,  $r, T \in (0, \infty)$ ,  $(t, \mathbf{x}) \in [0, T] \times \mathbb{R}_+^d$ , and covariance matrix  $\mathbf{C}_d \in \mathbb{R}^{d \times d}$  be the input of Algorithm 1. Let  $u(t, \mathbf{x}) \in \mathbb{R}$  be the option price given by (7) with CPWA payoff  $h : \mathbb{R}_+^d \rightarrow \mathbb{R}$  given by (8), let Assumption 2.2, Assumption 2.4, and Assumption 4.16 hold with respective constants  $C_1, C_2, C_3 \in [1, \infty)$ , and let  $\mathfrak{c}, \mathfrak{C}_1, \mathfrak{C}_2, \mathfrak{C}_3 \in [2, \infty)$  be constants defined by

$$\mathfrak{c} := 2C_2^2 e^{4C_1^2 T^2} e^{2rT} \max_{i=1, \dots, d} \{1, x_i^2\}, \quad (55)$$

$$\mathfrak{C}_1 := 648C_2 \log_2(\mathfrak{c}), \quad (56)$$

$$\mathfrak{C}_2 := (1.6 \times 10^8) C_2^4 C_3 (27 \log_2(\mathfrak{c}))^{\max\{3, 2C_3\}}, \quad (57)$$

$$\mathfrak{C}_3 := (6.1 \times 10^5) C_2 \mathfrak{c}^{\frac{3}{2}}. \quad (58)$$

Then, Algorithm 1 outputs  $\tilde{U}_{t, \mathbf{x}} \in \mathbb{R}$  which satisfies

$$|u(t, \mathbf{x}) - \tilde{U}_{t, \mathbf{x}}| \leq \varepsilon, \quad \text{with probability at least } 1 - \alpha, \quad (59)$$

where the number of qubits used in Algorithm 1 is at most

$$\mathfrak{C}_1 d^2 (1 + \log_2(d\varepsilon^{-1})), \quad (60)$$

the number of elementary gates used in Algorithm 1 is at most

$$\mathfrak{C}_2 d^{\max\{10.75, 4.75 + C_3\}} \varepsilon^{-3} (1 + \log_2(d\varepsilon^{-1}))^{\max\{3, 2C_3\}}, \quad (61)$$

and the number of applications<sup>7</sup> of quantum circuit  $\mathcal{A}$  in Algorithm 1 is at most

$$\mathfrak{C}_3 d^{4.75} \varepsilon^{-3} \ln\left(\frac{21}{\alpha}\right). \quad (62)$$

**Remark 2.22** Let us remark the following on the complexity of Algorithm 1.

1. The bounds (60) and (61) on the number of qubits and the number of elementary gates in Algorithm 1 specify the requirements on the quantum computer needed to run Algorithm 1. The bound (62) on the number of applications of quantum circuit  $\mathcal{A}$  can be interpreted as the computational running time for Algorithm 1.
2. The  $O(\varepsilon^{-3})$  running time complexity in (62) can be attributed as follows.

- (i) The truncation of the integral (7) from  $\mathbb{R}_+^d$  to the cube  $[0, M]^d$  requires  $M \sim 2^{n_1} \sim O(\varepsilon^{-1})$ , see Proposition 5.4.
- (ii) Since the payoff function  $h$  grows linearly,  $\|h\|_{L^\infty((0, M)^d)}$  grows of order  $O(\varepsilon^{-1})$ . Hence we require the scaling parameter  $\mathfrak{s}$  to satisfy  $\mathfrak{s} \sim O\left(\left(\frac{\varepsilon}{\|h\|_{L^\infty((0, M)^d)}^3}\right)^{1/2}\right) \sim O(\varepsilon^2)$ , see Proposition 5.11.
- (iii) We use the Modified IQAE to output  $\hat{a}$  with accuracy  $\varepsilon \mathfrak{s} \sim O(\varepsilon^3)$  to obtain the estimate (59), see Proposition 5.12. This implies the query complexity bound (62).

In the case where the payoff function  $h : \mathbb{R}_+^d \rightarrow \mathbb{R}$  is bounded uniformly in  $\mathbf{x} \in \mathbb{R}_+^d$ , then the scaling parameter  $\mathfrak{s}$  requires only  $O(\varepsilon^{\frac{1}{2}})$ , see (ii) above. This implies that the number of applications of  $\mathcal{A}$  can be reduced to  $O(\varepsilon^{-\frac{3}{2}})$ , which is a speed-up compared to classical Monte Carlo methods and recovers the complexity observed in [69]. We highlight that the fact that an unbounded payoff function can lead to a higher complexity has been already outlined in [15, Equation (36)]. Moreover, we highlight that one cannot expect to obtain  $O(\varepsilon^{-1})$ , which would have meant to have a quadratic speed-up over classical Monte Carlo methods, since one cannot expect to have an oracle which can perfectly upload the distribution and payoff function in rotated form (see, e.g., (44) or [15, Equation (16)]), as already pointed out, e.g., in [38].

3. We note from (59)–(62) that the number of qubits and elementary gates used in Algorithm 1 as well as the number of applications of the quantum circuit  $\mathcal{A}$  grow only polynomially<sup>8</sup> in  $d$  and  $\varepsilon^{-1}$ . This is in line with classical (i.e. non-quantum) Monte Carlo methods, where the number of Monte Carlo samples required in order to approximate the solution of the Black-Scholes PDE  $u(t, \mathbf{x})$  only grows polynomially in the dimension  $d$  and  $\varepsilon^{-1}$ , see, e.g., [39].

Indeed, let  $N \in \mathbb{N}$  be the number of classical Monte Carlo samples, let  $\mathbf{Y}^{(n)}, n = 1, \dots, N$  be  $N$  independent samples of the stock price process  $\mathbf{S}_T$  given that  $\mathbf{S}_t = \mathbf{x}$  (c.f. Lemma 2.1), and let  $u_{\text{MC}}^N(t, \mathbf{x}) :=$

<sup>7</sup>c.f. Remark 2.21 Item 3. for the precise meaning of number of applications.

<sup>8</sup>under the additional assumption that  $\max_{i=1, \dots, d} \{x_i\}$  is uniformly bounded in  $d \in \mathbb{N}$ . This assumption is naturally fulfilled in practice, as  $x_i^2$  corresponds to the (squared) spot price of the  $i$ -th asset.

$\frac{e^{-r(T-t)}}{N} \sum_{n=1}^N h(\mathbf{Y}^{(n)})$  be the classical Monte Carlo approximation of  $u(t, \mathbf{x})$ . By using Markov inequality, the linear growth property of the payoff function  $h$  under Assumption 2.4 (see Lemma 5.2), and second moment estimates for the multi-variate lognormal distribution under Assumption 2.2 (see (237)), we see for any prescribed accuracy  $\varepsilon \in (0, 1)$  that

$$\begin{aligned} \mathbb{P}(|u(t, \mathbf{x}) - u_{\text{MC}}^N(t, \mathbf{x})| \geq \varepsilon) &= \mathbb{P}\left(\left|\mathbb{E}[h(\mathbf{S}_T) | \mathbf{S}_t = \mathbf{x}] - \frac{1}{N} \sum_{n=1}^N h(\mathbf{Y}^{(n)})\right| \geq \varepsilon e^{r(T-t)}\right) \\ &\leq \frac{\mathbb{E}\left[\left|\mathbb{E}[h(\mathbf{S}_T) | \mathbf{S}_t = \mathbf{x}] - \frac{1}{N} \sum_{n=1}^N h(\mathbf{Y}^{(n)})\right|^2\right]}{\varepsilon^2 e^{2r(T-t)}} \\ &\leq \frac{2C_2^4 d^3 \left(1 + de^{2C_1^2 T^2} e^{2rT} \max_{i=1, \dots, d} \{1, x_i^2\}\right)}{N \varepsilon^2 e^{2r(T-t)}}. \end{aligned}$$

Therefore, for any confidence level  $\alpha \in (0, 1)$ , by choosing

$$N \geq \frac{2C_2^4 d^3 \left(1 + de^{2C_1^2 T^2} e^{2rT} \max_{i=1, \dots, d} \{1, x_i^2\}\right)}{\alpha \varepsilon^2 e^{2r(T-t)}},$$

we can ensure that

$$\mathbb{P}(|u(t, \mathbf{x}) - u_{\text{MC}}^N(t, \mathbf{x})| \geq \varepsilon) \leq \alpha$$

as required. Since each sample of  $\mathbf{Y}^{(n)}$  requires  $d$  copies of standard Gaussian random variables, we conclude that the total computational cost for a classical Monte Carlo method to approximately solve the multi-variate Black-Scholes PDE (1) is about  $Nd = \mathcal{O}(d^5 \varepsilon^{-2})$ .

4. The explicit constants (56)–(58) are not likely to be sharp since we did not optimize every inequality when bounding the number of elementary gates used to construct the quantum circuits in Section 4.4.

### 3 Numerical Simulations

In this section, we discuss the implementation of the proposed quantum algorithm, and illustrate its numerical performance on six concrete European options introduced in Example 2.5 in dimensions one and two. We have developed a package we named **qfinance** using the **Qiskit** framework to implement our proposed algorithm. The **OptionPricing** class within this package enables the user to input the stock parameters (i.e. mean  $\mu$ , volatility  $\sigma$ , correlation  $\rho$ , interest rate  $r$ , current time  $t$ , spot price  $\mathbf{S}_t$ , and maturity  $T$ ), select from any of the six classes of CPWA payoff functions  $h$  presented in Example 2.5, together with its corresponding parameters (i.e. weights and strikes), as well as specify the error tolerance  $\varepsilon$  in order to obtain an approximated value for the solution  $u(t, \mathbf{x})$  of the PDE (1) with payoff function  $h$  which corresponds to the fair price of the option  $h$  at time  $t$  given spot price  $\mathbf{S}_t = \mathbf{x}$ .

In our presented numerical simulations, we implement Algorithm 1 to approximate the expected option payoff  $u(t, \mathbf{x})$  at time  $t = 0$ , see (7) for the analytic expression. In Section 3.1 for the single vanilla call option, we use an initial spot price of  $x = S_0 = 2.0$ , volatility of  $\sigma = 0.4$ , an annual risk-free interest rate of  $r = 0.04$ , and a time to maturity of  $T = 40$  days. In Sections 3.2, 3.4–3.6, we used two assets with each asset following the same parameters as above, and where the assets are correlated to each other with a correlation coefficient of  $\rho = 0.2$ . These asset parameters are processed to produce an expected future price of  $\mu_{S_T} := \mathbb{E}[S_T] \approx 2.00879$  with a standard deviation  $\sigma_{S_T} \approx 0.267168$ . In Section 3.3, the initial spot price is lowered to  $S_0 = 0.5$  to reduce the approximation error, and this results in a expected future price of  $\mu_{S_T} := \mathbb{E}[S_T] \approx 0.502197$  with a standard deviation  $\sigma_{S_T} \approx 0.083697$ .

All numerical experiments presented in this section were implemented using IBM's quantum computing toolkit **Qiskit** [59]. We utilized the **LogNormalDistribution** class from **Qiskit** to load the multivariate log-normal distribution on the quantum computer. We discretize each asset price  $S_T$  using three qubits, hence into eight distinct values on equally spaced points in the interval  $[\max\{\mu_{S_T} - 3\sigma_{S_T}, 0\}, \mu_{S_T} + 3\sigma_{S_T}]$ . Namely,  $|000\rangle$  is mapped to  $\max\{\mu_{S_T} - 3\sigma_{S_T}, 0\}$  and  $|111\rangle$  is mapped to  $\mu_{S_T} + 3\sigma_{S_T}$  in the natural binary order.

The integration of the distribution loading quantum circuit and the payoff quantum circuit forms the quantum circuit  $\mathcal{A}$ , which is used as the input for the Quantum Amplitude Estimation (QAE) algorithm. We employ the Modified Iterative Quantum Amplitude Estimation (Modified IQAE) in [29, Algorithm 1], with the parameters  $\alpha = 0.005$  and  $\epsilon = 0.001$ ; see Proposition 2.20. For the numerical results, *Estimated (mid)* corresponds to  $\tilde{U}_{t, \mathbf{x}}$  (see line 8 of Algorithm 1 for the precise rescaling of  $\hat{a}$ ) whereas *Estimated (high)* and *Estimated (low)* correspond to the same scaling as line 8 of Algorithm 1 but with respect to  $a_u$  and  $a_\ell$ , respectively. All numerical experiments in this section were implemented in **Python** using **Qiskit** on an Ubuntu 22.04 machine with a AMD Ryzen 9 5950X @ 3.875GHz CPU with 64GB of RAM and Nvidia GeForce RTX3090 GPU. The source codes are available at <https://github.com/jianjun-dot/quantum-finance>.

### 3.1 Vanilla Call Option

We first consider a single-variable call option with strike  $\kappa$  with the following payoff function

$$h(x) = \max\{x - \kappa, 0\}. \quad (63)$$

We then test the algorithm efficacy across a range of strike prices. The results, shown in Figure 3 demonstrate that the estimated expected payoffs align closely with the reference values, which are computed based on the probabilities in `LogNormalDistribution`.

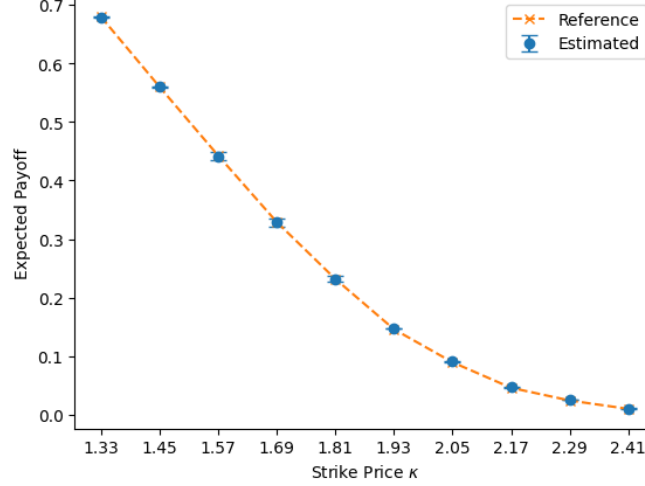


Figure 3: Comparison of the expected payoff estimates from the algorithm for the vanilla call option across a range of strike prices with the reference expected payoff. The tested strike prices are labeled on the horizontal axis.

Strike Price $\kappa$	Reference	Estimated (mid)	Estimated (high)	Estimated (low)
1.33	0.679331	0.679036	0.680195	0.677876
1.45	0.559664	0.559867	0.561016	0.558718
1.57	0.442470	0.441612	0.448517	0.434707
1.69	0.329094	0.328816	0.335622	0.322009
1.81	0.231919	0.232101	0.237502	0.226700
1.93	0.146172	0.147515	0.148352	0.146678
2.05	0.089769	0.090915	0.091357	0.090473
2.17	0.046210	0.047473	0.048274	0.046673
2.29	0.024531	0.025329	0.025692	0.024967
2.41	0.010191	0.010727	0.011324	0.010129

Table 1: Numerical results for vanilla call options

### 3.2 Basket Call Option

In our example, we consider the basket call with two assets and strike  $\kappa$  where the weight of each asset is set to one, i.e.,

$$h(x_1, x_2) = \max\{x_1 + x_2 - \kappa, 0\}. \quad (64)$$

Compared to the vanilla call option, the basket call option is more complex and requires the additional use of an adder circuit to compute  $x_1 + x_2$ . The results of the algorithm across a range of strike prices are shown in Figure 4.

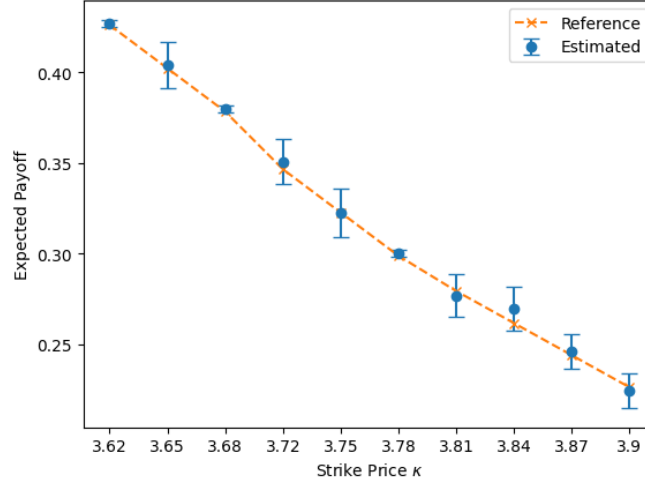


Figure 4: Expected payoff estimates from our proposed algorithm for the basket call option across a range of strike prices, compared with the reference expected payoff. The strike prices tested are labeled on the horizontal axis.

Strike Price $\kappa$	Reference	Estimated (mid)	Estimated (high)	Estimated (low)
3.62	0.426180	0.427173	0.428978	0.425367
3.65	0.402294	0.404174	0.416717	0.391631
3.68	0.378408	0.379875	0.381912	0.377837
3.72	0.346561	0.350753	0.363412	0.338095
3.75	0.322675	0.322466	0.335617	0.309314
3.78	0.298789	0.300295	0.302123	0.298467
3.81	0.279313	0.276816	0.288819	0.264814
3.84	0.261602	0.269334	0.281401	0.257267
3.87	0.243890	0.246027	0.255811	0.236242
3.90	0.226178	0.224180	0.233560	0.214800

Table 2: Numerical results for basket call options

### 3.3 Spread Call Option

We consider a spread call option involving two assets, given by

$$h(x_1, x_2) = \max\{x_1 - x_2 - \kappa, 0\}, \quad (65)$$

where  $\kappa$  is the strike price. Unlike the basket call option, which uses an addition operation, the spread call option requires a subtraction circuit to compute  $x_1 - x_2$ . This subtraction operation adds complexity to the quantum circuit due to the introduction of negative numbers as a result of the computation. Given that the input to the payoff function circuit falls within the domain  $\{0, 1, \dots, 2^n\}$ , where  $n$  is the number of qubits used to represent the number, additional processing of the subtraction results (which includes negative integers) to non-negative integers is required to maintain computational consistency.

We assess the performance of our algorithm across a range of strike prices, with the results shown in Figure 5.

Strike Price $\kappa$	Reference	Estimated (mid)	Estimated (high)	Estimated (low)
0.010	0.026466	0.026805	0.027640	0.025970
0.015	0.025174	0.025178	0.026008	0.024348
0.02	0.023883	0.024037	0.024871	0.023204
0.025	0.022591	0.023138	0.023978	0.022299
0.03	0.021300	0.021790	0.022623	0.020958
0.035	0.020008	0.019895	0.020714	0.019076
0.04	0.018717	0.019175	0.019988	0.018362
0.045	0.017425	0.018021	0.018831	0.017211
0.05	0.016134	0.016448	0.017254	0.015641
0.055	0.014842	0.014569	0.015341	0.013797

Table 3: Numerical results for spread call options



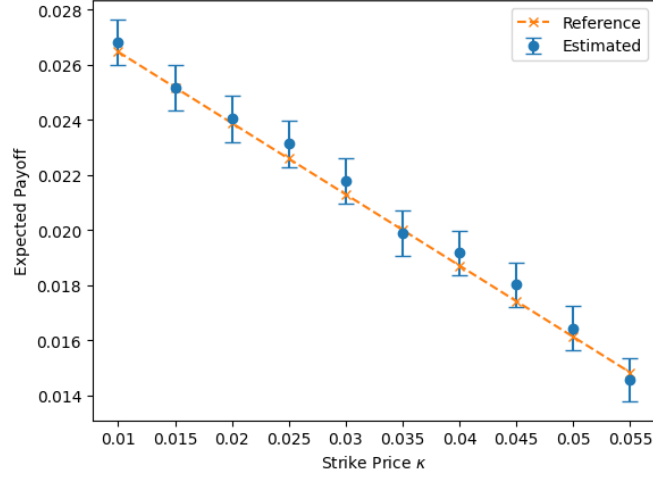


Figure 5: Expected payoff estimates from the algorithm for the spread call option across a range of strike prices, compared with the reference expected payoff. The strike prices tested is labeled on the horizontal axis.

### 3.4 Call-on-max Option

We explore a call-on-max option involving two assets and strike  $\kappa$ , defined by the payoff function

$$h(x_1, x_2) = \max\{x_1 - \kappa, x_2 - \kappa, 0\}. \quad (66)$$

This option requires the use of a comparison circuit  $\mathcal{Q}_{(\text{comp})}$  (later introduced in Lemma 4.13) to select of the higher value of  $\{x_1 - \kappa, x_2 - \kappa\}$  to compute the payoff. The algorithm is evaluated across a range of strike prices, with the results presented in Figure 6.

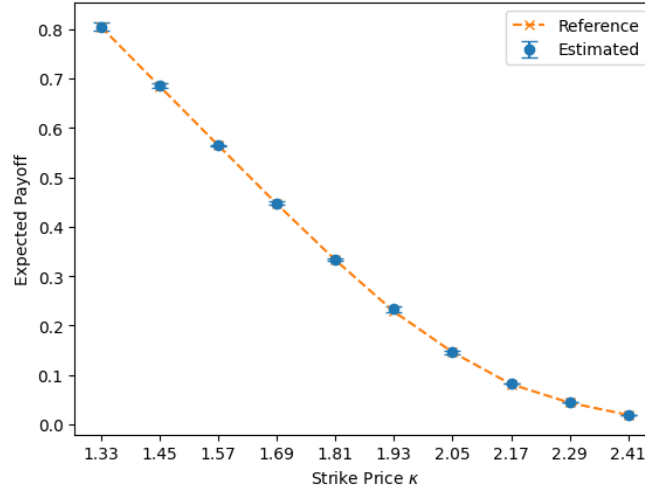


Figure 6: Expected payoff estimates from the algorithm for the call-on-max option across a range of strike prices, compared with the reference expected payoff. The strike prices tested is labeled on the horizontal axis.

Strike Price $\kappa$	Reference	Estimated (mid)	Estimated (high)	Estimated (low)
1.33	0.804465	0.805098	0.813612	0.796584
1.45	0.684487	0.685483	0.690424	0.680542
1.57	0.564676	0.564527	0.565501	0.563553
1.69	0.446152	0.447838	0.452069	0.443607
1.81	0.332589	0.332920	0.334962	0.330878
1.93	0.228361	0.233305	0.238528	0.228082
2.05	0.146165	0.145905	0.150115	0.141696
2.17	0.081361	0.082417	0.083002	0.081833
2.29	0.043842	0.044748	0.045053	0.044444
2.41	0.019170	0.019729	0.020016	0.019441

Table 4: Numerical results for call-on-max options

### 3.5 Call-on-min Option

We consider the two-asset call-on-min option with the following payoff function

$$h(x_1, x_2) = \max\{\min\{x_1, x_2\} - \kappa, 0\}. \quad (67)$$

This option is similar to that of the call-on-max option, resulting in a similar circuit design. The main change to the circuit is in selecting the lower value, rather than the higher value, of the comparison for the computation of the payoff. The performance of our proposed algorithm is evaluated across a range of strike prices, with the results shown in Figure 7.

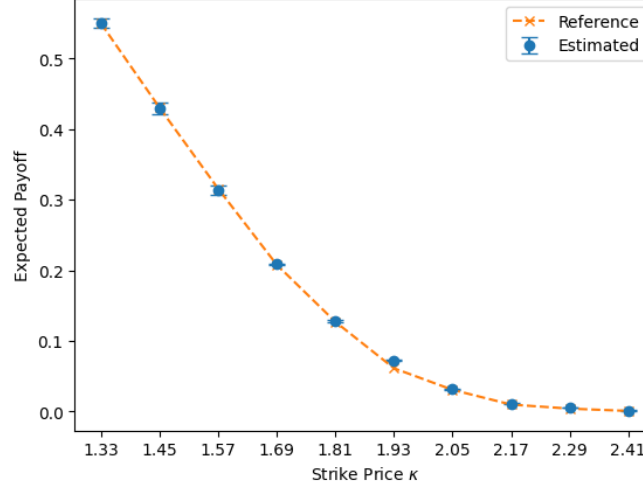


Figure 7: Expected payoff estimates from the algorithm for the call-on-min option across a range of strike prices, compared with the reference expected payoff. The strike prices tested is labeled on the horizontal axis.

Strike Price $\kappa$	Reference	Estimated (mid)	Estimated (high)	Estimated (low)
1.33	0.549242	0.549779	0.556598	0.542961
1.45	0.429959	0.429376	0.437922	0.420830
1.57	0.315390	0.313495	0.319649	0.307340
1.69	0.207789	0.208521	0.209452	0.207590
1.81	0.127052	0.128095	0.129009	0.127182
1.93	0.061264	0.072553	0.073112	0.071993
2.05	0.030753	0.031998	0.032863	0.031132
2.17	0.009879	0.011256	0.012077	0.010435
2.29	0.004124	0.005356	0.005689	0.005022
2.41	0.000863	0.001378	0.001902	0.000854

Table 5: Numerical results for call-on-min options

### 3.6 Best-of-call Option

The best-of-call option is the most complex option among the examples discussed, as it involves multiple assets with multiple strike prices. Its circuit requires a combination of the subtraction subroutine, the comparator subroutine and multiple payoff function circuits. In our case, we consider a two assets, two strike prices best-of-call option, with the following payoff function

$$h(x_1, x_2) = \max\{x_1 - \kappa_1, x_2 - \kappa_2, 0\}. \quad (68)$$

We tested our proposed algorithm across a range of strike prices, with the results shown in Figure 8.

### 3.7 Discussion on the numerical simulations and their possible extension to higher dimensions

We see from the numerical results that the algorithm meets our performance expectations. Compared to the results in [69, 74], our work extends the application to cover more complex call options, focusing on multi-asset call options such as spread call, call-on-max, call-on-min and best-of-call options. New circuits are constructed to handle these options, incorporating components such as subtraction subroutines and comparison subroutines.

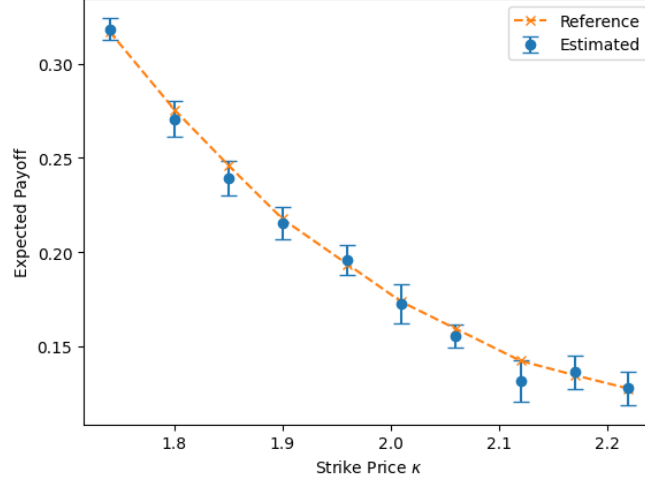


Figure 8: Expected payoff estimates from the algorithm for the best-of-call option across a range of strike prices, compared with the reference theoretical expected payoff. The strike price of the first asset is fixed, while the strike price of the second assets is varied.

Second strike price fixed at $\kappa_2 = 2.01$				
Strike Price $\kappa_1$	Reference	Estimated (mid)	Estimated (high)	Estimated (low)
1.74	0.317163	0.318514	0.324408	0.312619
1.80	0.275635	0.270604	0.280077	0.261131
1.85	0.246069	0.239256	0.248482	0.230031
1.90	0.217595	0.215355	0.223858	0.206853
1.96	0.193578	0.195811	0.203952	0.187671
2.01	0.173564	0.172656	0.183125	0.162187
2.06	0.159330	0.155583	0.161787	0.149379
2.12	0.142249	0.131429	0.142335	0.120523
2.17	0.134635	0.136207	0.145155	0.127260
2.22	0.127487	0.127613	0.136500	0.118725

Table 6: Numerical results for best-of-call options

Currently, our implementation is limited to options involving two variables. Nevertheless, the framework can be easily extended to multiple variables through the integration of multiple  $\mathcal{Q}_{\text{comp}}$  subroutines for variable comparisons. To discuss the involving steps more in detail, let us consider, e.g., the extension of the Best-of-call option from two variables to three variables. The payoff function of the three variable case is defined as:

$$h(x_1, x_2, x_3) = \max\{x_1 - \kappa_1, x_2 - \kappa_2, x_3 - \kappa_3, 0\} \quad (69)$$

which can be decomposed into multiple nested two-variable comparisons:

$$h(x_1, x_2, x_3) = \max\{\max\{\max\{x_1 - \kappa_1, x_2 - \kappa_2\}, x_3 - \kappa_3\}, 0\}. \quad (70)$$

This approach systematically extends the circuit from the two-variable case. For each variable, an ancilla register is added to load the corresponding strike price. Three different subtraction subroutines are applied on the corresponding variable and its ancilla, resulting in three different registers with that stores  $|x_1 - \kappa_1\rangle$ ,  $|x_2 - \kappa_2\rangle$ ,  $|x_3 - \kappa_3\rangle$ . To determine the maximum of three registers, we can use two comparison operators. The first  $\mathcal{Q}_{\text{comp}}$  operation compares registers  $|x_1 - \kappa_1\rangle$  and  $|x_2 - \kappa_2\rangle$ , storing the comparison results in the ancilla register  $|c_1\rangle$ . By controlling on the results of the ancilla register  $|c_1\rangle$ , a second  $\mathcal{Q}_{\text{comp}}(\max\{x_1 - \kappa_1, x_2 - \kappa_2\}, x_3 - \kappa_3)$  operation then compares the result  $\max\{x_1 - \kappa_1, x_2 - \kappa_2\}$  from the first comparison with  $x_3 - \kappa_3$ , and stores the result in another ancilla register  $|c_2\rangle$ . Using the outcomes of the two comparison operators  $\mathcal{Q}_{\text{comp}}(x_1 - \kappa_1, x_2 - \kappa_2)$  and  $\mathcal{Q}_{\text{comp}}(\max\{x_1 - \kappa_1, x_2 - \kappa_2\}, x_3 - \kappa_3)$ , the largest value can be identified as outlined Table 7. Accordingly, by controlling the circuit based on the comparison results, the appropriate register can be selected to compute the expected payoff. By iteratively applying this nested  $\mathcal{Q}_{\text{comp}}$  strategy, the algorithm can be generalized to calculate the payoff for any  $d$  variables.

For general CPWA payoff functions, we expect a quadratic increase (ignoring logarithmic factors) with respect to the dimension  $d$  in the number of qubits resource requirement due to the following two reasons – the number of comparison operations required is  $O(d)$  under Assumption 2.4 and the computation for each affine sum (i.e.  $\mathbf{x} \mapsto \mathbf{a}_{k,l} \cdot \mathbf{x} + b_{k,l}$  in (8)) requires  $O(d)$  arithmetic operations and storage. We refer to equation (218) for the amount  $N$  of qubits needed to upload the (discretized version of the) CPWA payoff function, which under

Assumption 2.4 satisfies  $N = O(d^2)$ . This and line 6 of Algorithm 1 hence imply that the amount of qubits needed for Algorithm 1 scales quadratically in the dimension  $d$  of the PDE.

We highlight that the presented steps involving nested  $\mathcal{Q}_{\text{comp}}$  can be similarly applied to other options presented in Example 2.5.

However, current quantum computing hardware are still too nascent to handle complex computations involving multiple variables. Thus, given the computational constraints, only two variables systems are implemented and tested in the package. As compute power increases, we can use the method described to extend the circuit to accommodate more variables.

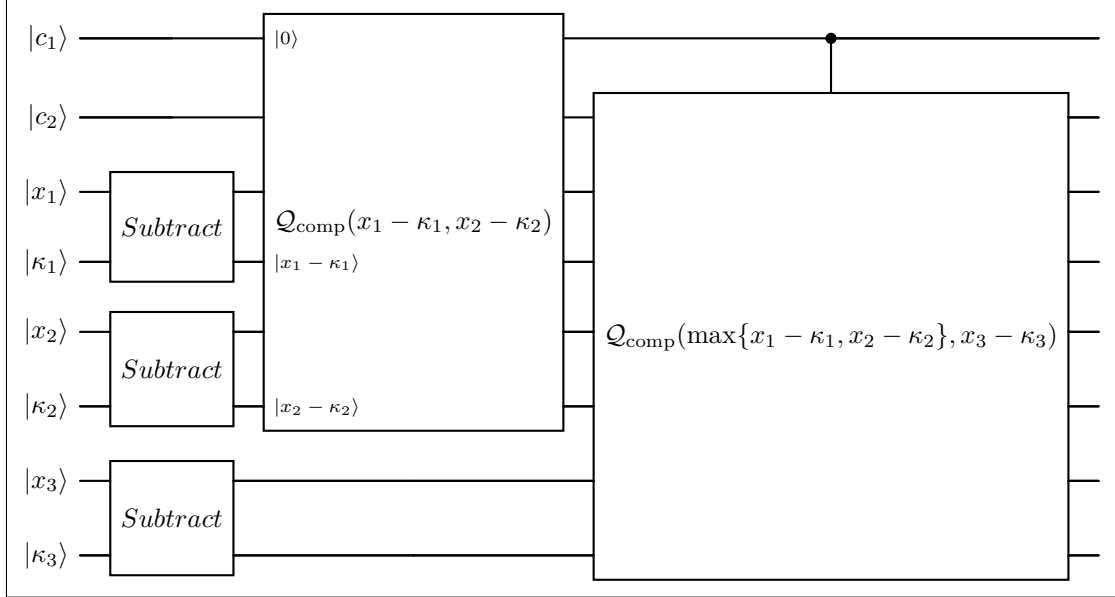


Figure 9: Description of the circuit to compare three variables. The first  $\mathcal{Q}_{\text{comp}}(x_1 - \kappa_1, x_2 - \kappa_2)$  compares the subtraction result stored in the register  $|\kappa_1\rangle$  and  $|\kappa_2\rangle$ , and the result of the comparison is store in register  $|c_1\rangle$ . The second comparison  $\mathcal{Q}_{\text{comp}}(\max\{x_1 - \kappa_1, x_2 - \kappa_2\}, x_3 - \kappa_3)$  is then controlled on the results from the first comparison, thereby selecting the correct register for the second comparison. The result of the second comparison is then stored in  $|c_2\rangle$ .

Measurement results for $ c_1\rangle_3$	Measurement results for $ c_2\rangle_3$	Maximal value
100	100	$x_3 - \kappa_3$
100	010	$x_2 - \kappa_2$
100	001	$x_2 - \kappa_2$
010	100	$x_3 - \kappa_3$
010	010	$x_1 - \kappa_1$
010	001	$x_1 - \kappa_1$
001	100	$x_3 - \kappa_3$
001	010	$x_1 - \kappa_1$
001	001	$x_1 - \kappa_1$

Table 7: The measurement results for the corresponding  $\mathcal{Q}_{\text{comp}}$  operations, and the identified maximal value among the three variables. The ancilla register  $|c_1\rangle$  holds the result for the first comparison  $\mathcal{Q}_{\text{comp}}(x_1 - \kappa_1, x_2 - \kappa_2)$  between  $x_1 - \kappa_1$  and  $x_2 - \kappa_2$ , whereas  $|c_2\rangle$  holds the result of the second comparison  $\mathcal{Q}_{\text{comp}}(\max\{x_1 - \kappa_1, x_2 - \kappa_2\}, x_3 - \kappa_3)$ . For cases where the two variables are equal, the first variable is selected as the maximal value. The ancilla results can locate the maximal value for computing the payoff in the subsequent parts of the circuit.

## 4 Quantum Circuits

In Section 4.1, we introduce the so-called *two's complement method* for representing signed dyadic rational numbers on a bounded interval using binary strings of finite length. These binary strings correspond to the grid points on the truncated interval. The binary strings are represented by the qubits on the quantum computer, in the form of linear combinations of the computational basis states  $|i\rangle_n = |i_1\rangle |i_2\rangle \dots |i_n\rangle$ , where  $i = (i_1, \dots, i_n) \in \{0, 1\}^n$ . In Section 4.2, we describe how quantum circuits are constructed to perform arithmetic operations on two complement's numbers. In Section 4.3, we assume that the discretized multivariate log-normal distribution can be loaded on the quantum computer with complexities comparable to the estimates in [15]. In Section 4.4, the approximate option payoff function is loaded, by using quantum circuits that perform arithmetic operations on numbers represented by the two's complement method.

#### 4.1 Representing signed dyadic rationals using the two's complement method

The two's complement method is a way of representing signed integers on a computer using binary strings, see e.g. Chapter 2.2 [13] for an introduction to the subject. We first describe the representation of signed integers using the two's complement method. For a given  $n \in \mathbb{N}$  and for an integer  $x \in [-2^{n-1}, 2^{n-1} - 1] \cap \mathbb{Z}$ , we encode  $x$  in the two's complement method by a  $n$ -bit string denoted by  $(x_{n-1}, x_{n-2}, \dots, x_0) \in \{0, 1\}^n$ . The value of the integer  $x$  is converted from the bit string  $(x_{n-1}, \dots, x_0)$  by the following formula

$$x = -x_{n-1}2^{n-1} + \sum_{k=0}^{n-2} x_k 2^k. \quad (71)$$

The most significant bit (MSB) is the bit  $x_{n-1} \in \{0, 1\}$  which determines the sign of  $x$ . There are classical computer algorithms for performing arithmetic operations (such as addition and multiplication) in the two's complement representation, such as the *carry adder algorithm* and *Booth's multiplication algorithm* [13]. Numbers with a fractional part can also be represented using the two's complement method. This can be done by introducing the *radix point* (commonly referred as the decimal point in decimal expansion) to separate the integer part and fractional part. The additional bits are mapped to the dyadics  $2^{-m}, m \in \mathbb{N}$  to represent the fractional part of a number.

**Definition 4.1 (Two's complement representation)** Let  $n \in \mathbb{N}$ ,  $m \in \mathbb{N}_0$ . For  $m \geq 1$ , we define the following set of  $(n, m)$ -bit strings by

$$\mathbb{F}_{n,m} := \{0, 1\}^n \times \{0, 1\}^m := \{((x_{n-1}, x_{n-2}, \dots, x_0), (x_{-1}, \dots, x_{-m})) \in \{0, 1\}^n \times \{0, 1\}^m\}, \quad (72)$$

and the set of dyadic rational numbers on a closed interval by

$$\mathbb{K}_{n,m} := [-2^{n-1}, 2^{n-1} - 2^{-m}] \cap 2^{-m}\mathbb{Z} := \{-2^{n-1}, -2^{n-1} + 2^{-m}, -2^{n-1} + 2 \cdot 2^{-m}, \dots, 2^{n-1} - 2^{-m}\}. \quad (73)$$

Further, we denote

$$\mathbb{K}_{n,m,+} := \mathbb{K}_{n,m} \cap [0, \infty), \quad \text{and} \quad \mathbb{K}_{n,m,-} := \mathbb{K}_{n,m} \cap (-\infty, 0), \quad (74)$$

and we denote

$$\begin{aligned} \mathbb{F}_{n,m,+} &:= \{((0, x_{n-2}, \dots, x_0), (x_{-1}, \dots, x_{-m})) : x_{n-2}, \dots, x_{-m} \in \{0, 1\}\}, \quad \text{and} \\ \mathbb{F}_{n,m,-} &:= \{((1, x_{n-2}, \dots, x_0), (x_{-1}, \dots, x_{-m})) : x_{n-2}, \dots, x_{-m} \in \{0, 1\}\}. \end{aligned} \quad (75)$$

If  $m = 0$ , we then use the usual signed integers  $\mathbb{K}_{n,0} := [-2^{n-1}, 2^{n-1} - 1] \cap \mathbb{Z}$  and the set of  $n$ -bit strings  $\mathbb{F}_{n,0} := \{(x_{n-1}, \dots, x_0) \in \{0, 1\}^n\}$ , and define  $\mathbb{K}_{n,0,\pm}$  and  $\mathbb{F}_{n,0,\pm}$  analogously.

**Definition 4.2 (Encoder and decoder maps)** Let  $n \in \mathbb{N}$ ,  $m \in \mathbb{N}_0$ . We define the encoder function which maps the rational numbers to bit strings by

$$\begin{aligned} E_{n,m} : \mathbb{K}_{n,m} &\longrightarrow \mathbb{F}_{n,m} \\ y &\mapsto ((x_{n-1}, x_{n-2}, \dots, x_0), (x_{-1}, \dots, x_{-m})) \end{aligned} \quad (76)$$

where we define  $E_{n,m}(y) = ((x_{n-1}, x_{n-2}, \dots, x_0), (x_{-1}, \dots, x_{-m}))$  recursively by

$$\begin{aligned} x_{n-1} &= \begin{cases} 1, & \text{if } y < 0, \\ 0, & \text{if } y \geq 0, \end{cases} \\ x_{n-2} &= \begin{cases} 1, & \text{if } -x_{n-1}2^{n-1} + 2^{n-2} \leq y, \\ 0, & \text{if } -x_{n-1}2^{n-1} + 2^{n-2} > y, \end{cases} \end{aligned} \quad (77)$$

and for  $k = n-3, n-4, \dots, -m$ ,

$$x_k = \begin{cases} 1, & \text{if } -x_{n-1}2^{n-1} + \sum_{j=k+1}^{n-2} x_j 2^j + 2^k \leq y, \\ 0, & \text{if } -x_{n-1}2^{n-1} + \sum_{j=k+1}^{n-2} x_j 2^j + 2^k > y. \end{cases} \quad (78)$$

We define the decoder function which maps the bit strings to the rational numbers by

$$\begin{aligned} D_{n,m} : \mathbb{F}_{n,m} &\longrightarrow \mathbb{K}_{n,m} \\ ((x_{n-1}, x_{n-2}, \dots, x_0), (x_{-1}, \dots, x_{-m})) &\mapsto -x_{n-1}2^{n-1} + \sum_{k=-m}^{n-2} x_k 2^k. \end{aligned} \quad (79)$$

The sets  $\mathbb{F}_{n,m}$  and  $\mathbb{K}_{n,m}$  are equivalent in the following sense.

**Proposition 4.3 (Bijection between  $(n, m)$ -bit strings and dyadics)** *Let  $n \in \mathbb{N}$ ,  $m \in \mathbb{N}_0$ . The sets  $\mathbb{F}_{n,m}$  and  $\mathbb{K}_{n,m}$  (c.f. Definition 4.1) have the same finite cardinality, and the encoder and decoder functions  $E_{n,m} : \mathbb{K}_{n,m} \rightarrow \mathbb{F}_{n,m}$  and  $D_{n,m} : \mathbb{F}_{n,m} \rightarrow \mathbb{K}_{n,m}$  (c.f. Definition 4.2) are bijective and inverses of the other.*

*Proof.* For the first part of the statement, by observing that

$$\mathbb{K}_{n,m} = \{j \cdot 2^{-m} : j = 0, 1, \dots, 2^{n+m-1} - 1\} \cup \{-j \cdot 2^{-m} : j = 1, \dots, 2^{n+m-1}\}, \quad (80)$$

it follows that

$$\#\mathbb{K}_{n,m} = 2 \cdot 2^{n+m-1} = 2^n \cdot 2^m = \#(\{0, 1\}^n \times \{0, 1\}^m) = \#\mathbb{F}_{n,m},$$

where we denote by  $\#A$  the cardinality of a set  $A$ . This shows that the two sets  $\mathbb{F}_{n,m}$  and  $\mathbb{K}_{n,m}$  have the same cardinality. Injectivity is clear from Definition 4.2, and bijectivity follows from injectivity since both sets have same finite cardinality.  $\square$

In the later sections, we will use the notation  $\mathbb{F}_{n,m}^k := \underbrace{\mathbb{F}_{n,m} \times \dots \times \mathbb{F}_{n,m}}_{k\text{-times}}$  for any  $k \in \mathbb{N}$ . The arithmetic algo-

rithms in two's complement (TC) representation for signed rational numbers can be extended from the arithmetic algorithms on the two's complement signed integers. The modifying process is done by shifting the fractional bits to the integer bits, applying the integer arithmetic algorithms, then shifting the integer bits back to fractional bits. The proofs in the two following lemmas provide the extension procedure.

**Lemma 4.4 (Addition in two's complement)** *Let  $n_1, n_2 \in \mathbb{N}$ , and let  $n := \max\{n_1, n_2\}$ . Let  $\boxplus : \mathbb{F}_{n_1,0} \times \mathbb{F}_{n_2,0} \rightarrow \mathbb{F}_{n+1,0}$  be the addition algorithm for integers represented in the two's complement method. Then, for any  $m_1, m_2 \in \mathbb{N}_0$  with  $m := \max\{m_1, m_2\}$ , there is a natural extension of the addition algorithm to the rational numbers represented in the two's complement method where  $\boxplus : \mathbb{F}_{n_1,m_1} \times \mathbb{F}_{n_2,m_2} \rightarrow \mathbb{F}_{n+1,m}$ , such that for any  $x \in \mathbb{F}_{n_1,m_1}$ ,  $y \in \mathbb{F}_{n_2,m_2}$ , there is a unique element  $x \boxplus y \in \mathbb{F}_{n+1,m}$  that satisfies*

$$x \boxplus y = E_{n+1,m}(D_{n_1,m_1}(x) + D_{n_2,m_2}(y)). \quad (81)$$

*Proof.* First, consider the case where both  $m_1$  and  $m_2$  are positive. Let  $x = ((x_{n_1-1}, \dots, x_0), (x_{-1}, \dots, x_{-m_1})) \in \mathbb{F}_{n_1,m_1}$  and  $y = ((y_{n_2-1}, \dots, y_0), (y_{-1}, \dots, y_{-m_2})) \in \mathbb{F}_{n_2,m_2}$  be given. For every  $p, q, r \in \mathbb{N}$  with  $r \geq q$ , define a left-shift operator  $\tau_r : \mathbb{F}_{p,q} \rightarrow \mathbb{F}_{p+r,0}$  defined by

$$\tau_r : ((z_{p-1}, \dots, z_0), (z_{-1}, \dots, z_{-q})) \mapsto (z_{p-1}, \dots, z_0, z_{-1}, \dots, z_{-q}, \underbrace{0, \dots, 0}_{(r-q)\text{-times}}). \quad (82)$$

Then, it holds that

$$\tau_m(x) = (x_{n_1-1}, \dots, x_0, x_{-1}, \dots, x_{-m_1}, \dots, x_{-m}) \in \mathbb{F}_{n_1+m,0}, \quad (83)$$

$$\tau_m(y) = (y_{n_2-1}, \dots, y_0, y_{-1}, \dots, y_{-m_2}, \dots, y_{-m}) \in \mathbb{F}_{n_2+m,0}, \quad (84)$$

where  $x_{-k} = 0$  for  $k = m_1 + 1, \dots, m$  if  $m_1 < m$  and  $y_{-l} = 0$  for  $l = m_2 + 1, \dots, m$  if  $m_2 < m$ . Hence, we may apply the integer addition algorithm and get an output  $\tau_m(x) \boxplus \tau_m(y) \in \mathbb{F}_{n+1+m,0}$ . Then, for every  $p, r \in \mathbb{N}$  with  $r \leq p$ , we define a right-shift operator  $\tau_{-r} : \mathbb{F}_{p,0} \rightarrow \mathbb{F}_{p-r,r}$  defined by

$$\tau_{-r} : (z_{p-1}, \dots, z_0) \mapsto ((z_{p-1}, \dots, z_r), (z_{r-1}, \dots, z_0)). \quad (85)$$

Let  $\tau_m(x) \boxplus \tau_m(y) = z = (z_{n+m}, z_{n+m-1}, \dots, z_0)$ , for some bit string  $z \in \mathbb{F}_{n+m+1,0}$ . Then, we have

$$\tau_{-m}(\tau_m(x) \boxplus \tau_m(y)) = ((z_{n+m}, z_{n+m-1}, \dots, z_m), (z_{m-1}, \dots, z_0)) \in \mathbb{F}_{n+1,m}. \quad (86)$$

Furthermore, it holds that  $D_{n_1+m,0}(\tau_m(x)) = 2^m D_{n_1,m_1}(x)$  and  $D_{n_2+m,0}(\tau_m(y)) = 2^m D_{n_2,m_2}(y)$ . This implies that

$$x \boxplus y := \tau_{-m}(\tau_m(x) \boxplus \tau_m(y)) = \tau_{-m}E_{n+m+1,0}(2^m(D_{n_1,m_1}(x) + D_{n_2,m_2}(y))) = E_{n+1,m}(D_{n_1,m_1}(x) + D_{n_2,m_2}(y)). \quad (87)$$

Hence we have shown (81). The cases where  $m_1$  and/or  $m_2$  equals to zero follow analogously.  $\square$

**Lemma 4.5 (Multiplication in two's complement)** *Let  $n_1, n_2 \in \mathbb{N}$ , and let  $n := n_1 + n_2$ . Let  $\boxtimes : \mathbb{F}_{n_1,0} \times \mathbb{F}_{n_2,0} \rightarrow \mathbb{F}_{n,0}$  be the multiplication algorithm for integers represented in the two's complement method. Then, for any  $m_1, m_2 \in \mathbb{N}_0$  with  $m := m_1 + m_2$ , there is a natural extension of the multiplication algorithm to the rational numbers represented in the two's complement method where  $\boxtimes : \mathbb{F}_{n_1,m_1} \times \mathbb{F}_{n_2,m_2} \rightarrow \mathbb{F}_{n,m}$  such that for any  $x \in \mathbb{F}_{n_1,m_1}$ ,  $y \in \mathbb{F}_{n_2,m_2}$ , there is a unique element  $x \boxtimes y \in \mathbb{F}_{n,m}$ , where*

$$x \boxtimes y = E_{n,m}(D_{n_1,m_1}(x) \cdot D_{n_2,m_2}(y)). \quad (88)$$

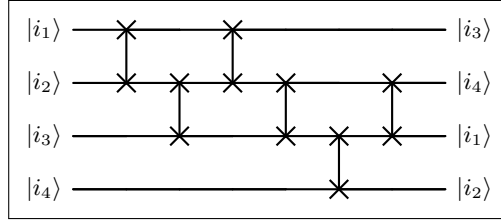


Figure 10: An example for the quantum circuit for permutation in Lemma 4.8, with  $\pi = (13)(24)$ .

*Proof.* First, consider the case where both  $m_1$  and  $m_2$  are positive. Let  $x = ((x_{n_1}, \dots, x_0), (x_{-1}, \dots, x_{-m_1})) \in \mathbb{F}_{n_1, m_1}$  and  $y = ((y_{n_2}, \dots, y_0), (y_{-1}, \dots, y_{-m_2})) \in \mathbb{F}_{n_2, m_2}$  be given. Then, with the left-shift operator  $\tau_m$  defined in (82) in the proof of the previous lemma, it holds that  $\tau_m(x) \in \mathbb{F}_{n_1+m, 0}$  and  $\tau_m(y) \in \mathbb{F}_{n_2+m, 0}$ . Hence, applying the multiplication algorithm on integers we have  $\tau_m(x) \boxtimes \tau_m(y) \in \mathbb{F}_{n+m, 0}$ . This implies that  $\tau_{-m}(\tau_m(x) \boxtimes \tau_m(y)) \in \mathbb{F}_{n, m}$ . We verify that

$$x \boxtimes y := \tau_{-m}(\tau_m(x) \boxtimes \tau_m(y)) = \tau_{-m}(E_{n+m, 0}(2^m D_{n_1, m_1}(x) \cdot D_{n_2, m_2}(y))) = E_{n, m}(D_{n_1, m_1}(x) \cdot D_{n_2, m_2}(y)). \quad (89)$$

Hence we have shown (88). The cases where  $m_1$  and/or  $m_2$  equals to zero follow analogously.  $\square$

## 4.2 Quantum circuits for elementary arithmetic operations

We now describe quantum circuits for arithmetic and elementary operations (such as addition, multiplication, comparison, absolute value), on qubit registers representing numbers in two's complement method. There are many quantum circuits for performing arithmetic operations with its quantum circuit complexities available in the literature, see, e.g., [4, 20, 24, 65, 67, 72]. We first introduce in Lemma 4.8 an important quantum circuit to perform permutations, which will be necessary for arithmetic computations in the later parts. We have included quantum circuit diagrams in this section to visualize the construction of the quantum circuits for their better understanding. For Lemma 4.9, Lemma 4.11, and Lemma 4.13, we refer the reader to [67] for its quantum circuit diagrams.

**Definition 4.6 (Cycle)** ([25, Section 1.3, pg 29]) Let  $n, m \in \mathbb{N}$  satisfy  $2 \leq m \leq n$ , and let  $\{a_1, \dots, a_m\} \subset \{1, 2, \dots, n\}$  be distinct numbers. A cycle  $C := (a_1 a_2 \dots a_m)$  is a permutation  $\sigma : \{1, 2, \dots, n\} \rightarrow \{1, 2, \dots, n\}$  such that

$$\sigma(j) = \begin{cases} a_{i+1}, & \text{if } j = a_i \text{ for } 1 \leq i \leq m-1, \\ a_1, & \text{if } j = a_m, \\ j, & \text{if } j \notin \{a_1, \dots, a_m\}. \end{cases} \quad (90)$$

**Proposition 4.7 (Cycle decomposition theorem)** ([25, Section 4.1, pg 115]) Let  $n \in \mathbb{N}$  and  $\pi : \{1, 2, \dots, n\} \rightarrow \{1, 2, \dots, n\}$  be a permutation. Then  $\pi$  can be written as a composition of disjoint cycles<sup>9</sup>

$$\pi = C_1 C_2 \dots C_k = (a_1 a_2 \dots a_{m_1})(a_{m_1+1} a_{m_1+2} \dots a_{m_2}) \dots (a_{m_{k-1}+1} a_{m_{k-1}+2} \dots a_{m_k}), \quad (91)$$

where  $k$  is the number of cycles, and  $\{a_j : j = 1, \dots, m_k\} \subset \{1, \dots, n\}$  are distinct integers. Moreover, the cycle decomposition above is unique up to a rearrangement of the cycles and up to a cyclic permutation of the integers within each cycle.

**Lemma 4.8 (Quantum circuit for permutation)** 1. Let  $n \in \mathbb{N}$ , and let  $\pi : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$  be a permutation. Then, there is a quantum circuit  $\mathcal{T}_\pi \in U(2^n)$  on  $n$  qubits such that for every  $|i\rangle_n = |i_1\rangle \dots |i_n\rangle \in \{0, 1\}^n$ , it holds that

$$\mathcal{T}_\pi |i\rangle_n = \mathcal{T}_\pi |i_1\rangle \dots |i_n\rangle = |i_{\pi(1)}\rangle \dots |i_{\pi(n)}\rangle, \quad (92)$$

and that  $\mathcal{T}_\pi$  uses at most  $2n^2$  swap gates (c.f. Example 2.11).

See Figure 10 for an example.

2. Let  $\mathcal{Q} \in U(2^n)$  be a given quantum circuit such that for every  $|i\rangle_n \in \{0, 1\}^n$ ,

$$\mathcal{Q}|i\rangle_n = \sum_{j=(j_1, \dots, j_n) \in \{0, 1\}^n} \alpha_{i,j} |j_1\rangle \dots |j_n\rangle, \quad (93)$$

where  $\alpha_{i,j} \in \mathbb{C}$ . Then,  $\mathcal{T}_\pi \mathcal{Q}$  is also a quantum circuit such that for every  $|i\rangle_n \in \{0, 1\}^n$ ,

$$\mathcal{T}_\pi \mathcal{Q}|i\rangle_n = \sum_{j \in \{0, 1\}^n} \alpha_{i,j} |j_{\pi(1)}\rangle \dots |j_{\pi(n)}\rangle. \quad (94)$$

<sup>9</sup>We adopt the convention that cycles of length 1 will not be written.

*Proof.* Firstly, we show, for any  $j, k \in \{1, 2, \dots, n\}$ ,  $j < k$ , and for any  $n$ -qubit  $|i\rangle_n = |i_1\rangle \cdots |i_n\rangle \in \{0, 1\}^n$ , that there exists a quantum circuit  $\mathcal{T}_{j \leftrightarrow k}$  consisting of  $2(k - j) - 1$  swap gates (c.f. Example 2.11) where it holds that

$$\begin{aligned} \mathcal{T}_{j \leftrightarrow k} : |i\rangle_n &= |i_1\rangle \cdots |i_{j-1}\rangle |i_j\rangle |i_{j+1}\rangle \cdots |i_{k-1}\rangle |i_k\rangle |i_{k+1}\rangle \cdots |i_n\rangle \\ &\mapsto |i_1\rangle \cdots |i_{j-1}\rangle |i_k\rangle |i_{j+1}\rangle \cdots |i_{k-1}\rangle |i_j\rangle |i_{k+1}\rangle \cdots |i_n\rangle. \end{aligned} \quad (95)$$

Denote by  $\mathcal{S} := \text{SWAP} \in U(2^2)$  the swap gate (c.f. Example 2.11) which satisfy for all  $|i_1\rangle |i_2\rangle \in \{0, 1\}^2$  that

$$\mathcal{S} |i_1\rangle |i_2\rangle = |i_2\rangle |i_1\rangle. \quad (96)$$

If  $k - j = 1$ , (i.e.  $k = j + 1$ ) then we simply set

$$\mathcal{T}_{j \leftrightarrow j+1} = I_2^{\otimes j-1} \otimes \mathcal{S} \otimes I_2^{\otimes n-j-1}. \quad (97)$$

If  $k - j \geq 2$ , then proceeding inductively set

$$\mathcal{T}_{j \leftrightarrow k} = \prod_{l=1}^{k-j-1} (I_2^{\otimes k-2-l} \otimes \mathcal{S} \otimes I_2^{\otimes n-k+l}) \prod_{l=0}^{k-j-1} (I_2^{\otimes j-1+l} \otimes \mathcal{S} \otimes I_2^{\otimes n-j-1-l}), \quad (98)$$

(c.f. Definition 2.14), where we use the usual convention that  $I_2^{\otimes 0} = 1 \in \mathbb{C}$  and  $A \otimes 1 = A = 1 \otimes A$  for any  $A \in U(2^m)$ ,  $m \in \mathbb{N}$ . By direct verification, we note that  $\mathcal{T}_{j \leftrightarrow k}$  satisfy (95) for all  $|i\rangle_n \in \{0, 1\}^n$  and that only  $2(k - j) - 1$  swap gates were required in its construction.

Secondly, by the cycle decomposition theorem (Proposition 4.7), the given permutation  $\pi$  can be written as a composition of disjoint cycles (c.f. Definition 4.6, Proposition 4.7) as

$$\pi = C_1 C_2 \cdots C_k = (a_1 a_2 \cdots a_{m_1}) (a_{m_1+1} a_{m_1+2} \cdots a_{m_2}) \cdots (a_{m_{k-1}+1} a_{m_{k-1}+2} \cdots a_{m_k}), \quad (99)$$

where  $k$  is the number of cycles, and  $a_1, \dots, a_{m_k} \in \{1, \dots, n\}$  are distinct numbers. Note by convention that each of these cycles has length  $m_l \geq 2$ . For each of these cycles  $C_l = (a_{m_{l-1}+1} \cdots a_{m_l})$ ,  $l = 1, \dots, k$ , with  $m_0 := 0$  we construct the quantum circuits  $\mathcal{T}_{C_l}$ ,  $l = 1, \dots, k$ , via

$$\mathcal{T}_{C_l} = \prod_{i=m_{l-1}}^{m_l-1} \mathcal{T}_{a_i \leftrightarrow a_{i+1}} = \mathcal{T}_{a_{m_{l-1}} \leftrightarrow a_{m_l}} \cdots \mathcal{T}_{a_{m_{l-1}+1} \leftrightarrow a_{m_{l-1}+2}} \mathcal{T}_{a_{m_{l-1}} \leftrightarrow a_{m_{l-1}+1}}, \quad (100)$$

where the quantum circuits  $\mathcal{T}_{a_i \leftrightarrow a_{i+1}}$  are constructed based on the first step of the proof (c.f. (95), (97), (98)). Finally, we construct the quantum circuit  $\mathcal{T}_\pi$  via

$$\mathcal{T}_\pi = \prod_{l=1}^k \mathcal{T}_{C_l} = \mathcal{T}_{C_k} \cdots \mathcal{T}_{C_2} \mathcal{T}_{C_1}. \quad (101)$$

Thus, (95), (99), (100), and (101) imply that the quantum circuit  $\mathcal{T}_\pi$  satisfies (92). Moreover, we note that the total number of quantum circuits of the form  $\mathcal{T}_{j \leftrightarrow k}$  (c.f. (95)) is  $m_k \leq n$ , where each of these quantum circuits requires  $2|a_i - a_{i+1}| - 1 \leq (2n - 1)$  swap gates. Hence, the total number of swap gates used to construct  $\mathcal{T}_\pi$  is at most  $n \cdot (2n - 1) \leq 2n^2$ . Thus, we have proved the first statement of the lemma. The second statement of the lemma follows directly from the fact that  $\mathcal{T}_\pi$  is a linear operator.  $\square$

**Lemma 4.9 (Quantum circuit for addition)** ([67, Section 3.1, QNMAAdd]) *Let  $n_1, n_2 \in \mathbb{N}$ , with  $n_1 \geq n_2$ . Then, there is a quantum circuit  $\mathcal{Q}_{(+)}$  on  $(n_1 + n_2 + 1)$  qubits such that for any  $a \in \mathbb{F}_{n_1, 0}, b \in \mathbb{F}_{n_2, 0}$ ,*

$$\mathcal{Q}_{(+)} : |0\rangle |a\rangle_{n_1} |b\rangle_{n_2} \mapsto |a \boxplus b\rangle_{n_1+1} |b\rangle_{n_2}. \quad (102)$$

*The quantum circuit  $\mathcal{Q}_{(+)}$  requires  $n_1^2 + 3n_1 + 18 + \frac{1}{2}(n_2(2n_1 - n_2 + 3))$  elementary gates.*

**Corollary 4.10 (Quantum circuit for addition with fractional part)** *Let  $n_1, n_2, m_1, m_2 \in \mathbb{N}$ , with  $n_1 + m_1 \geq n_2 + m_2$ . Let  $n = n_1 + n_2$ , and  $m = m_1 + m_2$ . Then, there is a quantum circuit  $\tilde{\mathcal{Q}}_{(+)}$  on  $(n + m + 1)$  qubits such that for any  $a \in \mathbb{F}_{n_1, m_1}, b \in \mathbb{F}_{n_2, m_2}$ ,*

$$\tilde{\mathcal{Q}}_{(+)} : |a\rangle_{n_1+m_1} |b\rangle_{n_2+m_2} |0\rangle \mapsto |b\rangle_{n_2+m_2} |a \boxplus b\rangle_{n_1+m_1+1}. \quad (103)$$

*The quantum circuit  $\tilde{\mathcal{Q}}_{(+)}$  requires at most  $29(n + m + 1)^2$  elementary gates. See Figure 11 for the circuit diagram.*

*Proof.* By Lemma 4.8, there is a quantum circuit  $\mathcal{T}_{\pi_1}$  with at most  $2(n_1 + m_1 + n_2 + m_2 + 1)^2 = 2(n + m + 1)^2$  swap gates satisfying

$$\mathcal{T}_{\pi_1} : |a\rangle_{n_1+m_1} |b\rangle_{n_2+m_2} |0\rangle \mapsto |0\rangle |a\rangle_{n_1+m_1} |b\rangle_{n_2+m_2}. \quad (104)$$



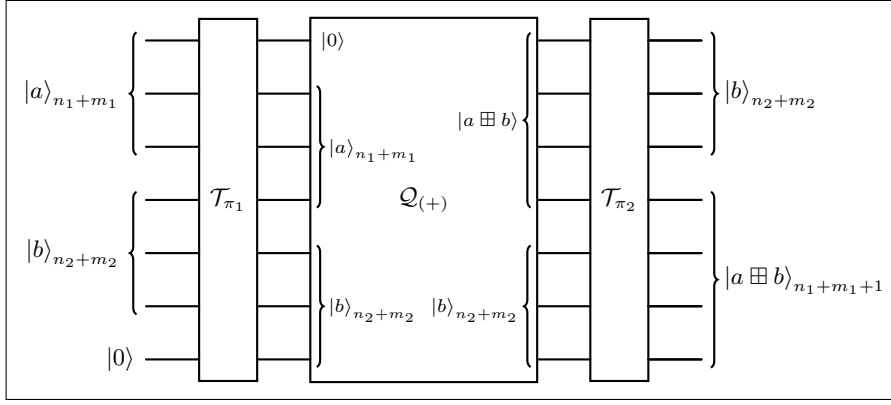


Figure 11: Circuit diagram for  $\tilde{Q}_{(+)}$  in Corollary 4.10.

Note that by Lemma 4.4, we can extend the addition operation  $\boxplus : \mathbb{F}_{n_1+m_1,0} \times \mathbb{F}_{n_2+m_2,0} \rightarrow \mathbb{F}_{n+m+1,0}$  to  $\boxplus : \mathbb{F}_{n_1,m_1} \times \mathbb{F}_{n_2,m_2} \rightarrow \mathbb{F}_{\tilde{n}+1,\tilde{m}}$  where  $\tilde{n} = \max\{n_1, n_2\}$  and  $\tilde{m} = \max\{m_1, m_2\}$ . This, the hypothesis that  $n_1 + m_1 \geq n_2 + m_2$ , and Lemma 4.9 (with  $n_1 \leftarrow n_1 + m_1$ ,  $n_2 \leftarrow n_2 + m_2$  in the notation of Lemma 4.9) imply that there exists a quantum circuit  $Q_{(+)}$  such that for any  $a \in \mathbb{F}_{n_1,m_1}$ ,  $b \in \mathbb{F}_{n_2,m_2}$  that

$$Q_{(+)} : |0\rangle |a\rangle_{n_1+m_1} |b\rangle_{n_2+m_2} \mapsto |a \boxplus b\rangle_{n_1+m_1+1} |b\rangle_{n_2+m_2}, \quad (105)$$

and that the number of elementary gates required to construct  $Q_{(+)}$  is

$$(n_1 + m_1)^2 + 3(n_1 + m_1) + 18 + \frac{1}{2}(n_2 + m_2)(2(n_1 + m_1) - (n_2 + m_2) + 3). \quad (106)$$

Moreover, by Lemma 4.8, there is a quantum circuit  $T_{\pi_2}$  such that

$$|a \boxplus b\rangle_{n_1+m_1+1} |b\rangle_{n_2+m_2} \mapsto |b\rangle_{n_2+m_2} |a \boxplus b\rangle_{n_1+m_1+1}, \quad (107)$$

which uses at most  $2(n + m + 1)^2$  swap gates.

Define the quantum circuit  $\tilde{Q}_{(+)} := T_{\pi_2} Q_{(+)} T_{\pi_1}$ . Observe that (104), (105), and (107) shows that  $\tilde{Q}_{(+)}$  satisfies (103), and that the total number of elementary gates required to construct  $\tilde{Q}_{(+)}$  is at most

$$\begin{aligned} & 2(n + m + 1)^2 + (n_1 + m_1)^2 + 3(n_1 + m_1) + 18 + \frac{1}{2}(n_2 + m_2)(2(n_1 + m_1) - (n_2 + m_2) + 3) + 2(n + m + 1)^2 \\ & \leq 2(n + m + 1)^2 + (n_1 + m_1)^2 + 3(n_1 + m_1) + 18 + (n + m + 1)^2 + \frac{3}{2}(n + m + 1)^2 + 2(n + m + 1)^2 \\ & \leq (2 + 1 + 3 + 18 + 1 + 2 + 2)(n + m + 1)^2 \\ & = 29(n + m + 1)^2. \end{aligned} \quad (108)$$

□

**Lemma 4.11 (Quantum circuit for multiplication)** ([67, Section 3.5, QNMMul]) Let  $n_1, n_2 \in \mathbb{N}$ , with  $n_1 \geq n_2$ . Then, there is a quantum circuit  $Q_{(\times)}$  on  $(2n_1 + 3n_2 + 3)$  qubits such that for any  $a \in \mathbb{F}_{n_1,0}$ ,  $b \in \mathbb{F}_{n_2,0}$ ,

$$Q_{(\times)} : |0\rangle |0\rangle_{n_1+n_2} |a\rangle_{n_1} |b\rangle_{n_2} |0\rangle_{n_2} |0\rangle_2 \mapsto |\text{anc}\rangle |a \boxtimes b\rangle_{n_1+n_2} |a\rangle_{n_1} |b\rangle_{n_2} |\text{anc}\rangle_{n_2+2}. \quad (109)$$

The quantum circuit  $Q_{(\times)}$  requires  $(\frac{1}{2}(5n_1^2 + n_1) + 4n_2^2 + 4n_1n_2 + 6n_2 + 7)$  elementary gates.

**Corollary 4.12 (Quantum circuit for multiplication with fractional part)** Let  $n_1, n_2, m_1, m_2 \in \mathbb{N}$ , with  $n_1 + m_1 \geq n_2 + m_2$ . Let  $n := n_1 + n_2$  and  $m := m_1 + m_2$ . Then, there is a quantum circuit  $\tilde{Q}_{(\times)}$  on  $(2n + 2m + n_2 + m_2 + 3)$  qubits such that for any  $a \in \mathbb{F}_{n_1,m_1}$ ,  $b \in \mathbb{F}_{n_2,m_2}$ ,

$$\tilde{Q}_{(\times)} : |a\rangle_{n_1+m_1} |b\rangle_{n_2+m_2} |0\rangle_{n+m} |0\rangle_{n_2+m_2+3} \mapsto |a\rangle_{n_1+m_1} |b\rangle_{n_2+m_2} |a \boxtimes b\rangle_{n+m} |\text{anc}\rangle_{n_2+m_2+3}. \quad (110)$$

The quantum circuit  $\tilde{Q}_{(\times)}$  requires at most  $61(n + m + 1)^2$  elementary gates. See Figure 12 for the circuit diagram.

*Proof.* By Lemma 4.8, there is a quantum circuit  $T_{\pi}$  with at most  $2(2n + 2m + n_2 + m_2 + 3)^2$  swap gates satisfying for any  $a \in \mathbb{F}_{n_1,m_1}$ ,  $b \in \mathbb{F}_{n_2,m_2}$  that

$$T_{\pi} : |a\rangle_{n_1+m_1} |b\rangle_{n_2+m_2} |0\rangle_{n+m} |0\rangle_{n_2+m_2+3} \mapsto |0\rangle |0\rangle_{n+m} |a\rangle_{n_1+m_1} |b\rangle_{n_2+m_2} |0\rangle_{n_2+m_2} |0\rangle_2. \quad (111)$$

Note that by Lemma 4.5, we can extend the multiplication operation  $\boxtimes : \mathbb{F}_{n_1+m_1,0} \times \mathbb{F}_{n_2+m_2,0} \rightarrow \mathbb{F}_{n+m,0}$  to  $\boxtimes : \mathbb{F}_{n_1,m_1} \times \mathbb{F}_{n_2,m_2} \rightarrow \mathbb{F}_{n,m}$ . This, the condition that  $n_1 + m_1 \geq n_2 + m_2$ , and Lemma 4.11 (with  $n_1 \leftarrow$

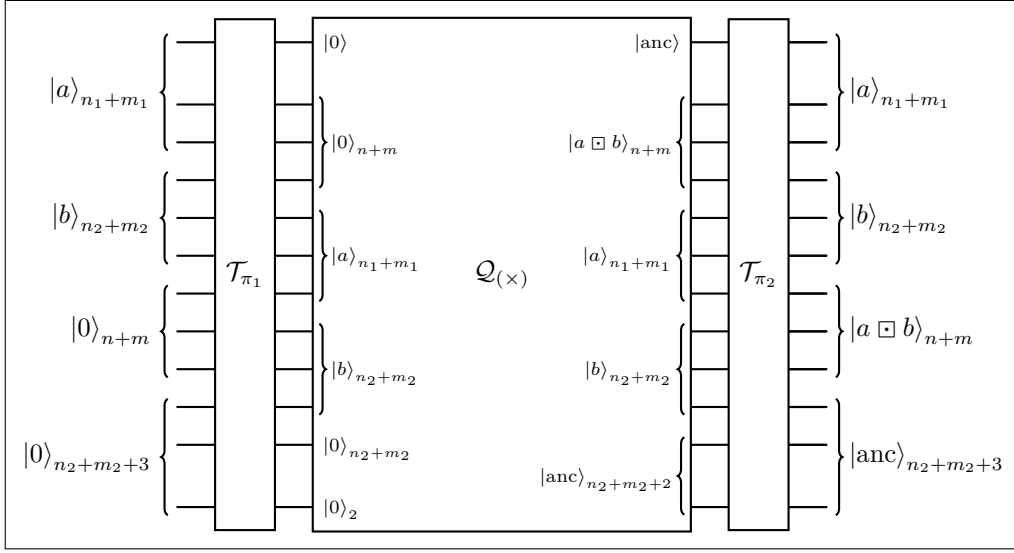


Figure 12: Circuit diagram for  $\tilde{\mathcal{Q}}_{(\times)}$  in Corollary 4.12.

$n_1 + m_1, n_2 \leftarrow n_2 + m_2$  in the notation of Lemma 4.11) imply that there exists a quantum circuit  $\mathcal{Q}_{(\times)}$  such that for any  $a \in \mathbb{F}_{n_1, m_1}$ ,  $b \in \mathbb{F}_{n_2, m_2}$

$$\mathcal{Q}_{(\times)} : |0\rangle |0\rangle_{n+m} |a\rangle_{n_1+m_1} |b\rangle_{n_2+m_2} |0\rangle_{n_2+m_2} |0\rangle_2 \mapsto |\text{anc}\rangle |a \boxminus b\rangle_{n+m} |a\rangle_{n_1+m_1} |b\rangle_{n_2+m_2} |\text{anc}\rangle_{n_2+m_2+2}, \quad (112)$$

and that the number of elementary gates required to construct  $\mathcal{Q}_{(\times)}$  is at most

$$\left(\frac{1}{2}(5(n_1 + m_1)^2 + (n_1 + m_1)) + 4(n_2 + m_2)^2 + 4(n_1 + m_1)(n_2 + m_2) + 6(n_2 + m_2) + 7\right). \quad (113)$$

By another application of Lemma 4.8, there is a quantum circuit  $\mathcal{T}_{\pi'}$  with at most  $2(2n + 2m + n_2 + m_2 + 3)^2$  swap gates satisfying

$$\mathcal{T}_{\pi'} : |\text{anc}\rangle |a \boxminus b\rangle_{n+m} |a\rangle_{n_1+m_1} |b\rangle_{n_2+m_2} |\text{anc}\rangle_{n_2+m_2+2} \mapsto |a \boxminus b\rangle_{n+m} |a\rangle_{n_1+m_1} |b\rangle_{n_2+m_2} |\text{anc}\rangle_{n_2+m_2+3}. \quad (114)$$

We define the quantum circuit  $\tilde{\mathcal{Q}}_{(\times)} = \mathcal{T}_{\pi'} \mathcal{Q}_{(\times)} \mathcal{T}_{\pi}$ . Hence, (111), (112), and (114) imply that the quantum circuit  $\tilde{\mathcal{Q}}_{(\times)}$  satisfy (110) for all  $a \in \mathbb{F}_{n_1, m_1}$ ,  $b \in \mathbb{F}_{n_2, m_2}$ . Moreover, the number of elementary gates required to construct  $\tilde{\mathcal{Q}}_{(\times)}$  is at most

$$\begin{aligned} & 2(2n + 2m + n_2 + m_2 + 3)^2 + \frac{1}{2}(5(n_1 + m_1)^2 + (n_1 + m_1)) + 4(n_2 + m_2)^2 + 4(n_1 + m_1)(n_2 + m_2) \\ & + 6(n_2 + m_2) + 7 + 2(2n + 2m + n_2 + m_2 + 3)^2 \\ & \leq 2 \cdot 3^2(n + m + 1)^2 + \frac{5}{2}(n + m + 1)^2 + \frac{1}{2}(n + m + 1) + 4(n + m + 1)^2 + 4(n + m + 1)^2 \\ & + 6(n + m + 1) + 7(n + m + 1) + 2 \cdot 3^2(n + m + 1)^2 \\ & \leq (18 + 3 + 1 + 4 + 4 + 6 + 7 + 18)(n + m + 1)^2 \\ & = 61(n + m + 1)^2. \end{aligned} \quad (115)$$

□

**Lemma 4.13 (Quantum circuit for integer comparison)** ([67, Section 3.4, QComp]) *Let  $n_1, n_2 \in \mathbb{N}$ , with  $n_1 \geq n_2$ . Let  $n = n_1 + n_2$ . Then, there is a quantum circuit  $\mathcal{Q}_{(\text{comp})}$  on  $(n_1 + n_2 + 4)$  qubits such that for any  $a \in \mathbb{F}_{n_1, 0}$ ,  $b \in \mathbb{F}_{n_2, 0}$ ,*

$$\mathcal{Q}_{(\text{comp})} : |0\rangle |a\rangle_{n_1} |b\rangle_{n_2} |0\rangle |0\rangle |0\rangle \mapsto |a \boxminus b\rangle_{n_1+1} |b\rangle_{n_2} |c_1\rangle |c_2\rangle |c_3\rangle, \quad (116)$$

where<sup>10</sup>

$$|c_1\rangle |c_2\rangle |c_3\rangle = \begin{cases} |1\rangle |0\rangle |0\rangle, & \text{if } D_{n_1, 0}(a) > D_{n_2, 0}(b), \\ |0\rangle |1\rangle |0\rangle, & \text{if } D_{n_1, 0}(a) < D_{n_2, 0}(b), \\ |0\rangle |0\rangle |1\rangle, & \text{if } D_{n_1, 0}(a) = D_{n_2, 0}(b). \end{cases} \quad (117)$$

The quantum circuit  $\mathcal{Q}_{(\text{comp})}$  uses  $(n_1^2 + 3n_1 + 41 + n_2(2n_1 - n_2 + 3)/2)$  elementary gates.

<sup>10</sup>The notation  $a \boxminus b$  here refers to subtraction for a pair of two complement numbers, i.e., for any  $a, b \in \mathbb{F}_{n, m}$ ,  $a \boxminus b \in \mathbb{F}_{n+1, m}$  is defined by  $E_{n+1, m}(D_{n, m}(a) - D_{n, m}(b))$ .

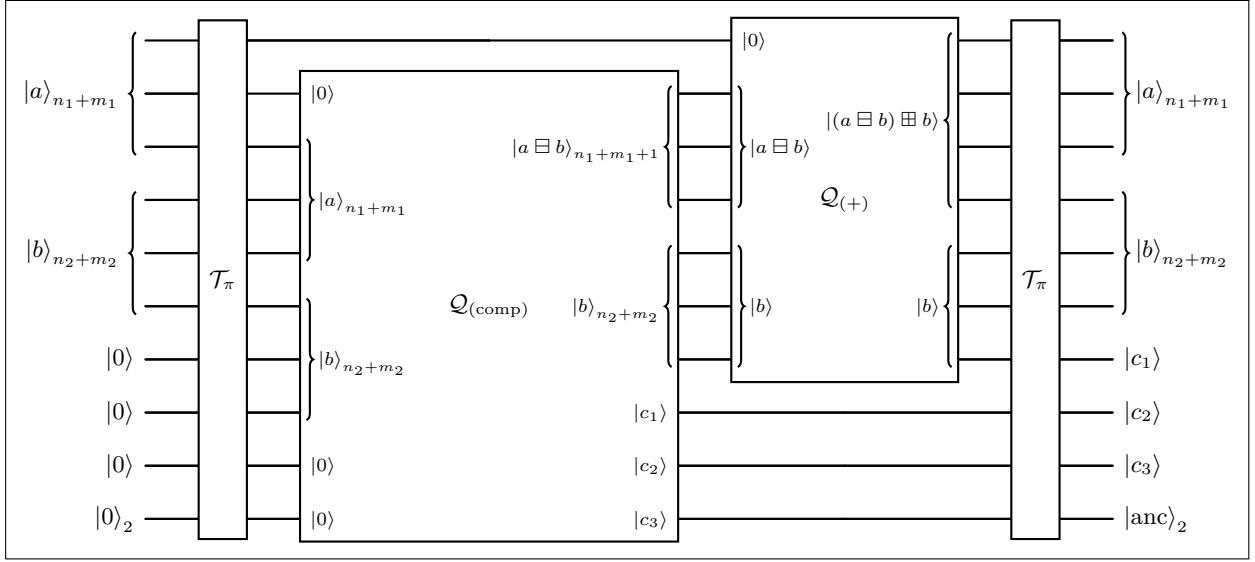


Figure 13: Circuit diagram for  $\tilde{\mathcal{Q}}_{(\text{comp})}$  in Corollary 4.14.

**Corollary 4.14 (Quantum circuit for fractional comparison)** Let  $n_1, n_2, m_1, m_2 \in \mathbb{N}$ , with  $n_1 + m_1 \geq n_2 + m_2$ . Let  $n = n_1 + n_2$ , and  $m = m_1 + m_2$ . Then, there is a quantum circuit  $\tilde{\mathcal{Q}}_{(\text{comp})}$  on  $(n + m + 5)$  qubits such that for any  $a \in \mathbb{F}_{n_1, m_1}$ ,  $b \in \mathbb{F}_{n_2, m_2}$ ,

$$\tilde{\mathcal{Q}}_{(\text{comp})} : |a\rangle_{n_1+m_1} |b\rangle_{n_2+m_2} |0\rangle_5 \mapsto |a\rangle_{n_1+m_1} |b\rangle_{n_2+m_2} |c_1\rangle |c_2\rangle |c_3\rangle |\text{anc}\rangle_2, \quad (118)$$

where

$$|c_1\rangle |c_2\rangle |c_3\rangle = \begin{cases} |1\rangle |0\rangle |0\rangle, & \text{if } D_{n_1, m_1}(a) > D_{n_2, m_2}(b), \\ |0\rangle |1\rangle |0\rangle, & \text{if } D_{n_1, m_1}(a) < D_{n_2, m_2}(b), \\ |0\rangle |0\rangle |1\rangle, & \text{if } D_{n_1, m_1}(a) = D_{n_2, m_2}(b). \end{cases} \quad (119)$$

The quantum circuit  $\tilde{\mathcal{Q}}_{(\text{comp})}$  requires at most  $175(n + m + 1)^2$  elementary gates. See Figure 13 for the circuit diagram.

*Proof.* The construction of the quantum circuit  $\tilde{\mathcal{Q}}_{(\text{comp})}$  consists of the following steps.

1. We first employ the permutation circuit  $\mathcal{T}_\pi$  from Lemma 4.8 so that

$$\mathcal{T}_\pi : |a\rangle_{n_1+m_1} |b\rangle_{n_2+m_2} |0\rangle_5 \mapsto |0\rangle |0\rangle |a\rangle_{n_1+m_1} |b\rangle_{n_2+m_2} |0\rangle |0\rangle |0\rangle. \quad (120)$$

In this step, the number of elementary gates used is at most  $2(n + m + 5)^2$ .

2. Next, we use the comparison quantum circuit  $\mathcal{Q}_{(\text{comp})}$  from Lemma 4.13 (with  $n_1 \leftarrow n_1 + m_1$ ,  $n_2 \leftarrow n_2 + m_2$  in the notation of Lemma 4.13) to obtain

$$I_2 \otimes \mathcal{Q}_{(\text{comp})} : |0\rangle |0\rangle |a\rangle_{n_1+m_1} |b\rangle_{n_2+m_2} |0\rangle |0\rangle |0\rangle \mapsto |0\rangle |a \boxminus b\rangle_{n_1+m_1+1} |b\rangle_{n_2+m_2} |c_1\rangle |c_2\rangle |c_3\rangle, \quad (121)$$

where  $|c_1\rangle |c_2\rangle |c_3\rangle$  satisfy (119). In this step, the number of elementary gates used is

$$(n_1 + m_1)^2 + 3(n_1 + m_1) + 41 + (n_2 + m_2)(2(n_1 + m_1) - (n_2 + m_2) + 3)/2.$$

3. We use the adder quantum circuit  $\mathcal{Q}_{(+)}$  from Lemma 4.9 (with  $n_1 \leftarrow n_1 + m_1 + 1$ ,  $n_2 \leftarrow n_2 + m_2$  in the notation of Lemma 4.9) to obtain

$$\mathcal{Q}_{(+)} \otimes I_2^{\otimes 3} : |0\rangle |a \boxminus b\rangle_{n_1+m_1+1} |b\rangle_{n_2+m_2} |c_1\rangle |c_2\rangle |c_3\rangle \mapsto |(a \boxminus b) \boxplus b\rangle_{n_1+m_1+2} |b\rangle_{n_2+m_2} |c_1\rangle |c_2\rangle |c_3\rangle. \quad (122)$$

In this step, the number of elementary gates used is

$$(n_1 + m_1 + 1)^2 + 3(n_1 + m_1 + 1) + 18 + \frac{1}{2}((n_2 + m_2)(2(n_1 + m_1 + 1) - (n_2 + m_2) + 3)).$$

4. As elements of  $\mathbb{F}_{n_1+2, m_2}$ , it can be directly checked that

$$(a \boxminus b) \boxplus b = \begin{cases} ((1, 1, a_{n_1-1}, \dots, a_0), (a_{-1}, \dots, a_{-m_1})) & \text{if } D_{n_1, m_1}(a) < 0, \\ ((0, 0, a_{n_1-1}, \dots, a_0), (a_{-1}, \dots, a_{-m_1})) & \text{if } D_{n_1, m_1}(a) \geq 0. \end{cases}$$

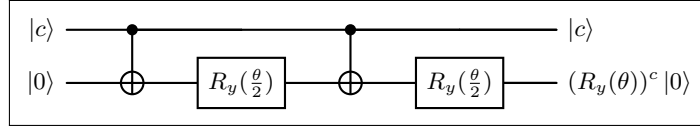


Figure 14: Circuit diagram for  $CR_y(\theta)$  in Lemma 4.15.

In the above expression, we treat the leftmost two bits as ancilla qubits and rewrite  $|(a \boxplus b) \boxplus b\rangle_{n_1+m_1+2} = |\text{anc}\rangle_2 |a\rangle_{n_1+m_1}$ . We apply another permutation circuit  $\mathcal{T}_\pi$  from Lemma 4.8 to obtain

$$\mathcal{T}_\pi : |(a \boxplus b) \boxplus b\rangle_{n_1+m_1+2} |b\rangle_{n_2+m_2} |c_1\rangle |c_2\rangle |c_3\rangle \mapsto |a\rangle_{n_1+m_1} |b\rangle_{n_2+m_2} |c_1\rangle |c_2\rangle |c_3\rangle |\text{anc}\rangle_2. \quad (123)$$

In this step, the number of elementary gates used is at most  $2(n+m+5)^2$ .

The desired quantum circuit  $\tilde{Q}_{(\text{comp})}$  is constructed from the above steps. We note that the number of elementary gates used to construct  $\tilde{Q}_{(\text{comp})}$  is at most

$$\begin{aligned} & 2(n+m+5)^2 + [(n_1+m_1)^2 + 3(n_1+m_1) + 41 + (n_2+m_2)(2(n_1+m_1) - (n_2+m_2) + 3)/2] \\ & + [(n_1+m_1+1)^2 + 3(n_1+m_1+1) + 18 + \frac{1}{2}((n_2+m_2)(2(n_1+m_1+1) - (n_2+m_2) + 3))] + 2(n+m+5)^2 \\ & \leq 2 \cdot 5^2(n+m+1)^2 + [(n+m+1)^2 + 3(n+m+1) + 41 + (n+m+1)^2 + \frac{3}{2}(n+m+1)] \\ & + [(n+m+1)^2 + 3(n+m+1) + 18 + (n+m+1)^2 + \frac{3}{2}(n+m+1)^2 + 2(n+m+1)^2] + 2 \cdot 5^2(n+m+1)^2 \\ & \leq (50 + 48 + 27 + 50)(n+m+1)^2 \\ & = 175(n+m+1)^2. \end{aligned} \quad (124)$$

□

**Lemma 4.15 (Controlled Y-rotations)** For any  $\theta \in (0, 4\pi)$ , there is a controlled Y-rotation gate acting on two qubits that performs the following operation

$$CR_y(\theta) : |c\rangle |0\rangle \mapsto |c\rangle (R_y(\theta))^c |0\rangle = \begin{cases} |c\rangle |0\rangle, & \text{if } c = 0, \\ |c\rangle (\cos(\theta/2) |0\rangle + \sin(\theta/2) |1\rangle), & \text{if } c = 1. \end{cases} \quad (125)$$

The quantum circuit to construct  $CR_y(\theta)$  requires two  $R_y(\theta/2)$  gates (see Example 2.9) and two CNOT gates (see Example 2.10). See Figure 14 for the circuit diagram.

*Proof.* The quantum circuit can be constructed by the following definition

$$CR_y(\theta) = (I_2 \otimes R_y(\theta/2))(\text{CNOT})(I_2 \otimes R_y(\theta/2))(\text{CNOT}). \quad (126)$$

□

### 4.3 Distribution loading

The task of loading an arbitrary  $n$ -qubit state on a quantum computer is known generally to be a hard problem, as highlighted, e.g., in [43]. However, in some cases, the problem of loading states representing certain probability distributions on a quantum computer have been shown to be polynomially tractable. Grover and Rudolph have shown an efficient method to load a discrete approximation of any log-concave probability distributions [35]. Recently, Zoufal et al. have employed the so-called *quantum Generative Adversarial Networks* (qGANs) for learning and loading of probability distributions such as the uniform, normal, or log-normal distributions, including their multivariate versions [76]. For these distributions, it has been shown empirically that the qGANs can well approximate the truncated and discretized distributions, and the gate complexity of the qGANs circuits scale only polynomially in the number of input qubits. In [15], Chakrabarti et. al. constructed a quantum circuit for uploading the discretized multivariate log-normal distributions, where they used the Variational Quantum Eigensolvers (VQE) approach [56] to upload quantum circuits for approximating the cumulative log-return process  $R_t^i$ , defined in (3). It was estimated that loading the discretized multivariate log-normal distribution requires  $O(Ld^2n \log_2(\varepsilon^{-1}))$  gates, where  $L \in \mathbb{N}$  is the depth of each variational quantum circuit for approximating the Gaussian distribution and  $n$  is the number of qubits used in each quantum circuit for approximating the Gaussian distribution, see [15, Appendix E]. Justified by the above examples in the literature, we make the following assumption.

**Assumption 4.16 (Loading of discretized multivariate log-normal distribution)** Let  $n, m \in \mathbb{N}$ , and let  $T > 0$ . For every  $d \in \mathbb{N}$  and  $(t, \mathbf{x}) \in [0, T] \times \mathbb{R}_+^d$  let  $p_d(\cdot, T; \mathbf{x}, t) : \mathbb{R}_+^d \rightarrow \mathbb{R}_+$  be the log-normal transition density given by (4). Then, we assume that there exists a constant  $C_3 \in [1, \infty)$  such that for every  $d \in \mathbb{N}$  and  $\varepsilon > 0$  there exists a quantum circuit  $\mathcal{P}_{d,\varepsilon}$  on  $d(n+m)$  qubits such that the number of elementary gates used to construct  $\mathcal{P}_{d,\varepsilon}$  is at most

$$C_3 d^{C_3} (n+m)^{C_3} (\log_2(\varepsilon^{-1}))^{C_3} \quad (127)$$

and that  $\mathcal{P}_{d,\varepsilon}$  satisfies

$$\mathcal{P}_{d,\varepsilon} |0\rangle_{d(n+m)} = \sum_{\mathbf{i}=(i_1, \dots, i_d) \in \mathbb{F}_{n,m,+}^d} \sqrt{\tilde{p}_{\mathbf{i}}} |i_1\rangle_{n+m} \cdots |i_d\rangle_{n+m}, \quad (128)$$

with coefficients  $\tilde{p}_{\mathbf{i}} \in [0, 1]$  satisfying

$$\sum_{\mathbf{i} \in \mathbb{F}_{n,m,+}^d} \tilde{p}_{\mathbf{i}} = 1 \quad (129)$$

and

$$\sum_{\mathbf{i} \in \mathbb{F}_{n,m,+}^d} |\tilde{p}_{\mathbf{i}} - \gamma^{-1} p_{\mathbf{i},m}| \leq \varepsilon, \quad (130)$$

where

$$p_{\mathbf{i},m} := \int_{Q_{\mathbf{i},m}} p_d(\mathbf{y}, T; \mathbf{x}, t) d\mathbf{y}, \quad Q_{\mathbf{i},m} := [D_{n,m}(i_1), D_{n,m}(i_1) + 2^{-m}] \times \cdots \times [D_{n,m}(i_d), D_{n,m}(i_d) + 2^{-m}], \quad (131)$$

and

$$\gamma := \sum_{\mathbf{i} \in \mathbb{F}_{n,m,+}^d} p_{\mathbf{i},m} \in (0, 1) \quad (132)$$

is a normalization constant.

**Remark 4.17** In case one uses the quantum circuit constructed in [15, Appendix E] to upload the discretized multivariate log-normal distribution, the corresponding constant  $C_3$  defined in Assumption 4.16 can be chosen to be  $C_3 := \max\{2, L\}$ , where  $L \in \mathbb{N}$  is the depth of each variational quantum circuit involved in [15, Appendix E] for approximating the involved Gaussian distributions; we refer to [15, Appendix E] for the precise construction of their quantum circuit.

**Remark 4.18** For general probability distributions, loading of its discretized probability density function (PDFs) remains one of the main problems in quantum computing. In the quantum computing literature, this step is also referred to as quantum state preparation, and it is an important initialization step for many quantum algorithms for pricing options. Recently, there has been new approaches to the quantum state preparation problem in the literature, that are not related to the qGAN or VQE methods reviewed above. In [40], the authors considered the quantum state preparation problem for probability distribution with smooth differentiable density functions, such as the normal distribution, where they proposed an algorithm based on the matrix product state (MPS) approximation method, and provide an error analysis and numerical convergence for the single-variate normal distribution. In [58], the author proposed a quantum binomial tree algorithm to approximate the option prices in a discrete time setting. We refer the reader to [16] for a similar random walk based algorithm, and to [71] for a hybrid classical quantum approach based on deconvolution methods for the quantum state preparation problem. However, to the best of our knowledge, there seems not to be any result in the literature that provides rigorous upper bounds on the quantum circuit complexities and as well as convergence for general multi-variate distributions.

#### 4.4 Loading CPWA payoff functions

The goal of this section is to upload (an approximation of) the payoff function  $h : \mathbb{R}^d \rightarrow \mathbb{R}$  given in (8) to a quantum circuit. To that end, let  $K \in \mathbb{N}$  be the number of component functions of the payoff function  $h$  given in (8), and for  $k = 1, \dots, K$ , let  $h_k : [0, M]^d \rightarrow \mathbb{R}$  be (up to the sign) the corresponding  $k$ -th component of  $h$  given by

$$h_k(\mathbf{x}) = \max\{\mathbf{a}_{k,l} \cdot \mathbf{x} + b_{k,l} : l = 1, \dots, I_k\}, \quad (133)$$

where  $\mathbf{a}_{k,l} \in \mathbb{R}^d$ ,  $b_{k,l} \in \mathbb{R}$  for  $l = 1, \dots, I_k$ . The parameters  $(\mathbf{a}_{k,l}, b_{k,l})$  are approximated by the two's complement method with binary strings of a suitable length. These binary strings are loaded on a qubit register using quantum circuits with  $X$ -gates, see Lemma 4.19. Using the arithmetic quantum circuits that we have constructed in the Section 4.2, we construct a quantum circuit which computes the two's complement-discretized version of the payoff function  $h_k(x)$ . The discrete payoff function is then loaded by a controlled  $Y$ -rotation circuit, see Lemma 4.23 and Proposition 4.24. We have also included quantum circuit diagrams in this section for the ease of understanding of the involved quantum circuits.

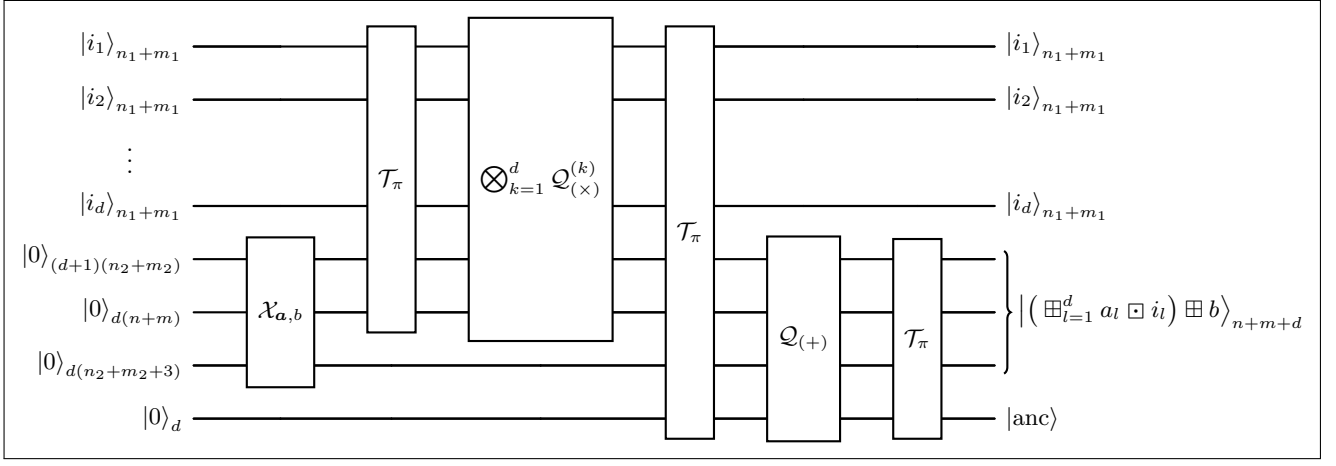


Figure 15: Circuit diagram for  $\mathcal{Q}_+^{d,n,m}$  in Lemma 4.19.

**Lemma 4.19 (Quantum circuit for affine sums)** Let  $d, n_1, n_2 \in \mathbb{N}$ ,  $m_1, m_2 \in \mathbb{N}_0$ . Let  $n := n_1 + n_2$ , and  $m := m_1 + m_2$ . Let  $a_1, \dots, a_d, b \in \mathbb{F}_{n_2, m_2}$ . Then, there is a quantum circuit  $\mathcal{Q}_+^{d,n,m}$  on  $N$  qubits, where

$$N := d(n_1 + m_1) + (d + 1)(n_2 + m_2) + d(n + m) + d(n_2 + m_2 + 3) + d \quad (134)$$

such that for any  $i_1, \dots, i_d \in \mathbb{F}_{n_1, m_1}$ ,

$$\begin{aligned} \mathcal{Q}_+^{d,n,m} : & |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |0\rangle_{(d+1)(n_2+m_2)} |0\rangle_{d(n+m)} |0\rangle_{d(n_2+m_2+3)} |0\rangle_d \\ & \mapsto |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} \left| \left( \bigoplus_{k=1}^d (a_k \boxplus i_k) \right) \boxplus b \right\rangle_{n+m+d} |anc\rangle_p, \end{aligned} \quad (135)$$

with  $p := d(2n_2 + 2m_2 + 3) + (n_2 + m_2) + (d - 1)(n + m)$ , and where  $(\bigoplus_{k=1}^d (a_k \boxplus i_k)) \boxplus b \in \mathbb{F}_{d+n, m}$  is the two's complement binary string representing the affine sum

$$\left( \sum_{k=1}^d (D_{n_2, m_2}(a_k) \cdot D_{n_1, m_1}(i_k)) \right) + D_{n_2, m_2}(b) \in \mathbb{K}_{n+d, m}, \quad (136)$$

(c.f. Definition 4.2). The quantum circuit  $\mathcal{Q}_+^{d,n,m}$  uses at most  $563d^3(n + m + 1)^2$  elementary gates. See Figure 15 for the circuit diagram.

*Proof.* The construction of this circuit involves the following steps:

1. We first load the given two's complement binary strings  $a_1, \dots, a_d, b \in \mathbb{F}_{n_2, m_2}$  on the qubit register  $|0\rangle_{(d+1)(n_2+m_2)} = |0\rangle_{n_2+m_2} \cdots |0\rangle_{n_2+m_2}$ . To that end, we use the Pauli  $X$  gate (see Example 2.6) to flip the bit 0 to 1 according the binary strings  $a_1, \dots, a_d, b$  if necessary, to obtain the state

$$\mathcal{X}_{a,b} : |0\rangle_{n_2+m_2} \cdots |0\rangle_{n_2+m_2} \mapsto |a_1\rangle_{n_2+m_2} \cdots |a_d\rangle_{n_2+m_2} |b\rangle_{n_2+m_2}, \quad (137)$$

where we define

$$\mathcal{X}_{a,b} := \left( \bigotimes_{k=1}^d \bigotimes_{l=-m_2}^{n_2-1} X^{a_k(l)} \right) \otimes \left( \bigotimes_{l=-m_2}^{n_2-1} X^{b(l)} \right), \quad (138)$$

given the binary strings  $a_k = ((a_k(l))_{l=0}^{n_2-1}, (a_k(l))_{l=-m_2}^{-1}) = ((a_k(n_2 - 1), \dots, a_k(0)), (a_k(-1), \dots, a_k(-m_2))) \in \mathbb{F}_{n_2, m_2}$  and  $b = ((b(n_2 - 1), \dots, b(0)), (b(-1), \dots, b(-m_2))) \in \mathbb{F}_{n_2, m_2}$ . Note that we use the convention of  $X^0 = I_2$  for any unitary matrix  $X$ . We define the quantum circuit

$$\tilde{\mathcal{X}}_{a,b} := I_2^{\otimes d(n_1+m_1)} \otimes \mathcal{X}_{a,b} \otimes I_2^{\otimes (d(n+m)+d(n_2+m_2+3)+d)}. \quad (139)$$

Hence, for any  $i_1, \dots, i_d \in \mathbb{F}_{n_1, m_1}$ , we have

$$\begin{aligned} \tilde{\mathcal{X}}_{a,b} : & |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |0\rangle_{(d+1)(n_2+m_2)} |0\rangle_{d(n+m)} |0\rangle_{d(n_2+m_2+3)} |0\rangle_d \\ & \mapsto |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |a_1\rangle_{n_2+m_2} \cdots |a_d\rangle_{n_2+m_2} |b\rangle_{n_2+m_2} |0\rangle_{d(n+m)} |0\rangle_{d(n_2+m_2+3)} |0\rangle_d, \end{aligned} \quad (140)$$

and the number of Pauli  $X$  gates used to construct the quantum circuit  $\tilde{\mathcal{X}}_{a,b}$  is at most

$$(d + 1)(n_2 + m_2). \quad (141)$$

2. Next, we apply the permutation quantum circuit  $\mathcal{T}_\pi$  from Lemma 4.8 to prepare for the upcoming  $d$  multiplications so that

$$\begin{aligned} \mathcal{T}_\pi : & |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |a_1\rangle_{n_2+m_2} \cdots |a_d\rangle_{n_2+m_2} |b\rangle_{n_2+m_2} |0\rangle_{d(n+m)} |0\rangle_{d(n_2+m_2+3)} |0\rangle_d \\ & \mapsto \bigotimes_{k=1}^d (|i_k\rangle_{n_1+m_1} |a_k\rangle_{n_2+m_2} |0\rangle_{n+m} |0\rangle_{n_2+m_2+3}) \otimes |b\rangle_{n_2+m_2} |0\rangle_d \\ & = |i_1\rangle_{n_1+m_1} |a_1\rangle_{n_2+m_2} |0\rangle_{n+m} |0\rangle_{n_2+m_2+3} \cdots |i_d\rangle_{n_1+m_1} |a_d\rangle_{n_2+m_2} |0\rangle_{n+m} |0\rangle_{n_2+m_2+3} |b\rangle_{n_2+m_2} |0\rangle_d. \end{aligned} \quad (142)$$

The number of swap gates used to construct  $\mathcal{T}_\pi$  in this step is at most  $2N^2$ .

3. Next, for each  $k = 1, \dots, d$ , we apply the multiplication quantum circuit  $\mathcal{Q}_{(\times)}^{(k)} := \tilde{\mathcal{Q}}_{(\times)}$  from Corollary 4.12 (with  $n_1 \leftarrow n_1, n_2 \leftarrow n_2, m_1 \leftarrow m_1, m_2 \leftarrow m_2, a \leftarrow i_k, b \leftarrow a_k$  in the notation of Corollary 4.12) on each component  $(|i_k\rangle_{n_1+m_1} |a_k\rangle_{n_2+m_2} |0\rangle_{n+m} |0\rangle_{n_2+m_2+3})$  such that

$$\begin{aligned} \bigotimes_{k=1}^d \mathcal{Q}_{(\times)}^{(k)} : & \bigotimes_{k=1}^d (|i_k\rangle_{n_1+m_1} |a_k\rangle_{n_2+m_2} |0\rangle_{n+m} |0\rangle_{n_2+m_2+3}) \otimes |b\rangle_{n_2+m_2} |0\rangle_d \\ & \mapsto \bigotimes_{k=1}^d (|i_k\rangle_{n_1+m_1} |a_k\rangle_{n_2+m_2} |a_k \boxminus i_k\rangle_{n+m} |\text{anc}\rangle_{n_2+m_2+3}) \otimes |b\rangle_{n_2+m_2} |0\rangle_d \\ & = |i_1\rangle_{n_1+m_1} |a_1\rangle_{n_2+m_2} |a_1 \boxminus i_1\rangle_{n+m} |\text{anc}\rangle_{n_2+m_2+3} \cdots \\ & \quad \cdots |i_d\rangle_{n_1+m_1} |a_d\rangle_{n_2+m_2} |a_d \boxminus i_d\rangle_{n+m} |\text{anc}\rangle_{n_2+m_2+3} |b\rangle_{n_2+m_2} |0\rangle_d. \end{aligned} \quad (143)$$

The number of elementary gates used in this step is at most

$$d \cdot 61(n+m+1)^2. \quad (144)$$

4. We apply the permutation quantum circuit  $\mathcal{T}_\pi$  from Lemma 4.8 to prepare for the upcoming  $d$  additions so that

$$\begin{aligned} \mathcal{T}_\pi : & \bigotimes_{k=1}^d (|i_k\rangle_{n_1+m_1} |a_k\rangle_{n_2+m_2} |a_k \boxminus i_k\rangle_{n+m} |\text{anc}\rangle_{n_2+m_2+3}) \otimes |b\rangle_{n_2+m_2} |0\rangle_d \\ & \mapsto |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |a_1 \boxminus i_1\rangle_{n+m} |a_2 \boxminus i_2\rangle_{n+m} |0\rangle |a_3 \boxminus i_3\rangle_{n+m} |0\rangle \cdots \\ & \quad \cdots |a_d \boxminus i_d\rangle_{n+m} |0\rangle |b\rangle_{n_2+m_2} |0\rangle |\text{anc}\rangle_{d(2n_2+2m_2+3)}. \end{aligned} \quad (145)$$

Here, we consolidate the qubits  $|a_1\rangle_{n_2+m_2}, \dots, |a_d\rangle_{n_2+m_2}$  in the ancilla qubit placeholder  $|\text{anc}\rangle_{d(2n_2+2m_2+3)}$  as we do not need them in the later computations. The number of elementary gates used for this step is at most  $2N^2$ .

5. We perform the following addition inductively on the sums for  $k = 1, \dots, d-1$

$$\boxplus : \mathbb{F}_{n+k-1,m} \times \mathbb{F}_{n,m} \rightarrow \mathbb{F}_{n+k,m}, \quad \left( \boxplus_{l=1}^k (a_l \boxminus i_l), (a_{k+1} \boxminus i_{k+1}) \right) \mapsto \boxplus_{l=1}^{k+1} (a_l \boxminus i_l), \quad (146)$$

and the addition

$$\boxplus : \mathbb{F}_{n+d-1,m} \times \mathbb{F}_{n_2,m_2} \rightarrow \mathbb{F}_{n+d,m}, \quad \left( \boxplus_{l=1}^d (a_l \boxminus i_l), b \right) \mapsto \left( \boxplus_{l=1}^d (a_l \boxminus i_l) \right) \boxplus b, \quad (147)$$

(c.f. Lemma 4.4 for definition of  $\boxplus$ ). That is, we apply the quantum circuit  $\mathcal{Q}_{(+)}^{(k)} := \mathcal{Q}_{(+)}$  from Corollary 4.10 inductively for  $k = 1, \dots, d-1$  (with  $n_1 \leftarrow n+k-1, n_2 \leftarrow n, m_1 \leftarrow m, m_2 \leftarrow m, a \leftarrow \boxplus_{l=1}^k (a_l \boxminus i_l), b \leftarrow a_{k+1} \boxminus i_{k+1}$  in the notation of Corollary 4.10), and we apply the quantum circuit  $\mathcal{Q}_{(+)}^{(b)} := \mathcal{Q}_{(+)}$  (with  $n_1 \leftarrow n+d-1, n_2 \leftarrow n_2, m_1 \leftarrow m, m_2 \leftarrow m_2, a \leftarrow \boxplus_{l=1}^d (a_l \boxminus i_l), b \leftarrow b$  in the notation of Corollary 4.10) so

that

$$\begin{aligned}
& |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |a_1 \boxminus i_1\rangle_{n+m} |a_2 \boxminus i_2\rangle_{n+m} |0\rangle |a_3 \boxminus i_3\rangle_{n+m} |0\rangle \cdots \\
& \cdots |a_d \boxminus i_d\rangle_{n+m} |0\rangle |b\rangle_{n_2+m_2} |0\rangle |\text{anc}\rangle_{d(2n_2+2m_2+3)} \\
& \xrightarrow{\mathcal{Q}_{(+)}^{(1)}} |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |a_2 \boxminus i_2\rangle_{n+m} \left| \bigoplus_{l=1}^2 a_l \boxminus i_l \right\rangle_{n+m+1} |a_3 \boxminus i_3\rangle_{n+m} |0\rangle \cdots \\
& \cdots |a_d \boxminus i_d\rangle_{n+m} |0\rangle |b\rangle_{n_2+m_2} |0\rangle |\text{anc}\rangle_{d(2n_2+2m_2+3)} \\
& \xrightarrow{\mathcal{Q}_{(+)}^{(2)}} |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |a_2 \boxminus i_2\rangle_{n+m} |a_3 \boxminus i_3\rangle_{n+m} \left| \bigoplus_{l=1}^3 a_l \boxminus i_l \right\rangle_{n+m+2} \cdots \\
& \cdots |a_d \boxminus i_d\rangle_{n+m} |0\rangle |b\rangle_{n_2+m_2} |0\rangle |\text{anc}\rangle_{d(2n_2+2m_2+3)} \\
& \vdots \\
& \xrightarrow{\mathcal{Q}_{(+)}^{(d-1)}} |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |a_2 \boxminus i_2\rangle_{n+m} |a_3 \boxminus i_3\rangle_{n+m} \cdots \\
& \cdots |a_d \boxminus i_d\rangle_{n+m} \left| \bigoplus_{l=1}^d a_l \boxminus i_l \right\rangle_{n+m+d-1} |b\rangle_{n_2+m_2} |0\rangle |\text{anc}\rangle_{d(2n_2+2m_2+3)} \\
& \xrightarrow{\mathcal{Q}_{(+)}^{(b)}} |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |a_2 \boxminus i_2\rangle_{n+m} |a_3 \boxminus i_3\rangle_{n+m} \cdots \\
& \cdots |a_d \boxminus i_d\rangle_{n+m} |b\rangle_{n_2+m_2} \left| \left( \bigoplus_{l=1}^d a_l \boxminus i_l \right) \boxplus b \right\rangle_{n+m+d} |\text{anc}\rangle_{d(2n_2+2m_2+3)}.
\end{aligned} \tag{148}$$

The number of elementary gates used for this step is at most

$$\begin{aligned}
& \sum_{k=1}^{d-1} 29[(n+m+k-1) + (n+m) + 1]^2 + 29[(n+m+d-1) + (n_2+m_2) + 1]^2 \\
& \leq 29d \cdot 4d^2(n+m+1)^2 \\
& = 116d^3(n+m+1)^2,
\end{aligned} \tag{149}$$

where we use the fact that  $[(n+m+k-1) + (n+m) + 1]^2 \leq (2n+2m+d)^2 \leq 4d^2(n+m+1)^2$  when  $k \leq d$ .

6. We consolidate the ancillary qubits by combining the qubits (labeled  $|a_2 \boxminus i_2\rangle, \dots, |a_d \boxminus i_d\rangle, |b\rangle$ ) under ancilla qubits  $|\text{anc}\rangle_*$ . The permutation circuit  $\mathcal{T}_\pi$  from Lemma 4.8 performs the following operation

$$\begin{aligned}
& \mathcal{T}_\pi : |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |a_2 \boxminus i_2\rangle_{n+m} \cdots |a_d \boxminus i_d\rangle_{n+m} |b\rangle_{n_2+m_2} \left| \left( \bigoplus_{l=1}^d a_l \boxminus i_l \right) \boxplus b \right\rangle_{n+m+d} |\text{anc}\rangle_{d(2n_2+2m_2+3)} \\
& \mapsto |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} \left| \left( \bigoplus_{l=1}^d a_l \boxminus i_l \right) \boxplus b \right\rangle_{n+m+d} |\text{anc}\rangle_{d(2n_2+2m_2+3) + (n_2+m_2) + (d-1)(n+m)}.
\end{aligned} \tag{150}$$

The number of elementary gates used for this step is at most  $2N^2$ .

The resulting quantum circuit  $\mathcal{Q}_+^{d,n,m}$  is a composition of the quantum circuits from each of the above steps. To deduce its gate complexity, we sum up the number of elementary gates used in each step. We note from the definition of  $n, m \in \mathbb{N}$  and the definition of  $N$  in (134) that

$$\begin{aligned}
N &:= d(n_1+m_1) + (d+1)(n_2+m_2) + d(n+m) + d(n_2+m_2+3) + d \\
&\leq [d+2d+d+3d+d](n+m+1) \\
&= 8d(n+m+1).
\end{aligned} \tag{151}$$

Summing the number of elementary gates used in each step, we find that the number of elementary gates used in total is at most

$$\begin{aligned}
& (d+1)(n_2+m_2) + 2N^2 + 61d(n+m+1)^2 + 2N^2 + 116d^3(n+m+1)^2 + 2N^2 \\
& \leq 2d(n+m+1) + 3 \cdot 2(8d(n+m+1))^2 + 61d(n+m+1)^2 + 116d^3(n+m+1)^2 \\
& \leq (2+3 \cdot 2 \cdot 8^2 + 61+116)d^3(n+m+1)^2 \\
& = 563d^3(n+m+1)^2.
\end{aligned} \tag{152}$$

□



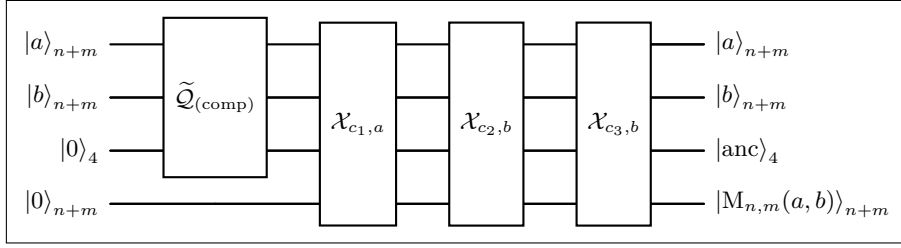


Figure 16: Circuit diagram for  $\mathcal{Q}_{(\max)}^{n,m}$  in Lemma 4.20.

**Lemma 4.20 (Quantum circuit for maximum of two numbers)** Let  $n, m \in \mathbb{N}$ , and let  $M_{n,m} : \mathbb{F}_{n,m} \times \mathbb{F}_{n,m} \rightarrow \mathbb{F}_{n,m}$  be a function defined by

$$M_{n,m}(a, b) := E_{n,m}(\max\{D_{n,m}(a), D_{n,m}(b)\}), \quad \forall a, b \in \mathbb{F}_{n,m}. \quad (153)$$

(c.f. Definition 4.2). Then, there is a quantum circuit  $\mathcal{Q}_{(\max)}^{n,m}$  on  $3(n+m)+5$  qubits such that for any  $a, b \in \mathbb{F}_{n,m}$ ,

$$\mathcal{Q}_{(\max)}^{n,m} : |a\rangle_{n+m} |b\rangle_{n+m} |0\rangle_5 |0\rangle_{n+m} \mapsto |a\rangle_{n+m} |b\rangle_{n+m} |\text{anc}\rangle_5 |M_{n,m}(a, b)\rangle_{n+m}, \quad (154)$$

which uses at most  $1045(n+m+1)^3$  elementary gates. See Figure 16 for the circuit diagram.

*Proof.* The construction of this quantum circuit involves the following steps:

1. We use the comparison quantum circuit  $\tilde{\mathcal{Q}}_{(\text{comp})}$  in Corollary 4.14 (with  $n_1 \leftarrow n, n_2 \leftarrow n, m_1 \leftarrow m, m_2 \leftarrow m, a \leftarrow a$  and  $b \leftarrow b$  in the notation of Corollary 4.14) to obtain

$$\tilde{\mathcal{Q}}_{(\text{comp})} \otimes I_2^{\otimes n+m} : |a\rangle_{n+m} |b\rangle_{n+m} |0\rangle_5 |0\rangle_{n+m} \mapsto |a\rangle_{n+m} |b\rangle_{n+m} |c_1\rangle |c_2\rangle |c_3\rangle |\text{anc}\rangle_2 |0\rangle_{n+m}, \quad (155)$$

where

$$|c_1\rangle |c_2\rangle |c_3\rangle = \begin{cases} |1\rangle |0\rangle |0\rangle, & \text{if } D_{n,m}(a) > D_{n,m}(b), \\ |0\rangle |1\rangle |0\rangle, & \text{if } D_{n,m}(a) < D_{n,m}(b), \\ |0\rangle |0\rangle |1\rangle, & \text{if } D_{n,m}(a) = D_{n,m}(b). \end{cases} \quad (156)$$

The number of gates used in this step is at most  $175(2n+2m+1)^2$  elementary gates.

2. We apply  $(n+m)$  Toffoli gate (circuits) CCNOT defined in Example 2.18 on the control qubit  $|c_1\rangle$ , target qubits  $|a\rangle_{n+m}$ , and output qubits  $|0\rangle_{n+m}$ . More precisely, for  $a = ((a_{n-1}, \dots, a_0), (a_{-1}, \dots, a_{-m})) \in \mathbb{F}_{n,m}$ , we apply for each  $j = -m, \dots, n-1$  a Toffoli gate  $\mathcal{C}_{c_1, a_j} = \text{CCNOT}$  (with  $a \leftarrow c_1, b \leftarrow a_j$ , and  $c \leftarrow 0$  in the notation of Example 2.18)

$$\mathcal{C}_{c_1, a_j} : |c_1\rangle |a_j\rangle |0\rangle \mapsto |c_1\rangle |a_j\rangle X^{c_1 a_j} |0\rangle = \begin{cases} |c_1\rangle |a_j\rangle |1\rangle, & \text{if } c_1 = 1 \text{ and } a_j = 1, \\ |c_1\rangle |a_j\rangle |0\rangle, & \text{otherwise.} \end{cases} \quad (157)$$

For each  $j = -m, \dots, n-1$ , we apply the permutation quantum circuit  $\mathcal{T}_j$  before and after each Toffoli gate  $\mathcal{C}_{c_1, a_j}$ . We define the quantum circuit  $\mathcal{X}_{c_1, j}$  by

$$\mathcal{X}_{c_1, j} := \mathcal{T}_j \left( I_2^{\otimes n+m-1} \otimes I_2^{\otimes n+m} \otimes I_2^{\otimes 4} \otimes I_2^{\otimes n-j-1} \otimes \mathcal{C}_{c_1, a_j} \otimes I_2^{\otimes m+j} \right) \mathcal{T}_j, \quad (158)$$

which computes from (155) the following

$$\begin{aligned} & |a\rangle_{n+m} |b\rangle_{n+m} |c_1\rangle |c_2\rangle |c_3\rangle |\text{anc}\rangle_2 |0\rangle_{n+m} \\ & \xrightarrow{\mathcal{T}_j} |\hat{a}^j\rangle_{n+m-1} |b\rangle_{n+m} |c_2\rangle |c_3\rangle |\text{anc}\rangle_2 |0\rangle_{n-j-1} |c_1\rangle |a_j\rangle |0\rangle_{m+j} \\ & \xrightarrow{\mathcal{C}_{c_1, a_j}} |\hat{a}^j\rangle_{n+m-1} |b\rangle_{n+m} |c_2\rangle |c_3\rangle |\text{anc}\rangle_2 |0\rangle_{n-j-1} |c_1\rangle |a_j\rangle X^{c_1 a_j} |0\rangle_{m+j} \\ & \xrightarrow{\mathcal{T}_j} |a\rangle_{n+m} |b\rangle_{n+m} |c_1\rangle |c_2\rangle |c_3\rangle |\text{anc}\rangle_2 |0\rangle_{n-j-1} X^{c_1 a_j} |0\rangle_{m+j}, \end{aligned} \quad (159)$$

where  $|\hat{a}^j\rangle_{n+m-1} := |a_{n-1}\rangle \cdots |a_{j+1}\rangle |a_{j-1}\rangle \cdots |a_{-m}\rangle$ . Finally, we define the quantum circuit  $\mathcal{X}_{c_1, a}$  by

$$\mathcal{X}_{c_1, a} := \prod_{j=-m}^{n-1} \mathcal{X}_{c_1, j}, \quad (160)$$

and we compute that

$$\begin{aligned} \mathcal{X}_{c_1,a} : & |a\rangle_{n+m} |b\rangle_{n+m} |c_1\rangle |c_2\rangle |c_3\rangle |\text{anc}\rangle_2 |0\rangle_{n+m} \\ \mapsto & |a\rangle_{n+m} |b\rangle_{n+m} |c_1\rangle |c_2\rangle |c_3\rangle |\text{anc}\rangle_2 \left( \bigotimes_{j=-m}^{n-1} X^{c_1 a_j} |0\rangle_{n+m} \right). \end{aligned} \quad (161)$$

Note that the number of elementary gates used in this step is at most  $(n+m)[15 + 2 \cdot 2(3(n+m) + 5)^2]$ , since each Toffoli gate circuit requires 15 gates, and each permutation circuit uses  $2(3(n+m) + 5)^2$  gates.

3. We repeat step 2 but by instead using the control qubit  $|c_2\rangle$  with target qubits  $|b\rangle_{n+m}$ . We define the quantum circuits  $\mathcal{X}_{c_2,b}$  similarly, and we compute that

$$\begin{aligned} \mathcal{X}_{c_2,b} : & |a\rangle_{n+m} |b\rangle_{n+m} |c_1\rangle |c_2\rangle |c_3\rangle |\text{anc}\rangle_2 \left( \bigotimes_{j=-m}^{n-1} X^{c_1 a_j} |0\rangle_{n+m} \right) \\ \mapsto & |a\rangle_{n+m} |b\rangle_{n+m} |c_1\rangle |c_2\rangle |c_3\rangle |\text{anc}\rangle_2 \left( \bigotimes_{j=-m}^{n-1} X^{c_2 b_j} X^{c_1 a_j} |0\rangle_{n+m} \right). \end{aligned} \quad (162)$$

4. We repeat step 2 but by instead using the control qubit  $|c_3\rangle$  with target qubits  $|b\rangle_{n+m}$ . We define the quantum circuits  $\mathcal{X}_{c_3,b}$  similarly, and we compute that

$$\begin{aligned} \mathcal{X}_{c_3,b} : & |a\rangle_{n+m} |b\rangle_{n+m} |c_1\rangle |c_2\rangle |c_3\rangle |\text{anc}\rangle_2 \left( \bigotimes_{j=-m}^{n-1} X^{c_2 b_j} X^{c_1 a_j} |0\rangle_{n+m} \right) \\ \mapsto & |a\rangle_{n+m} |b\rangle_{n+m} |c_1\rangle |c_2\rangle |c_3\rangle |\text{anc}\rangle_2 \left( \bigotimes_{j=-m}^{n-1} X^{c_3 b_j} X^{c_2 b_j} X^{c_1 a_j} |0\rangle_{n+m} \right). \end{aligned} \quad (163)$$

Since the qubits  $|c_1\rangle |c_2\rangle |c_3\rangle$  may only take one of the three possible values as in (156), it holds for each  $j = -m, \dots, n-1$  that

$$X^{c_3 b_j} X^{c_2 b_j} X^{c_1 a_j} = \begin{cases} X^{a_j}, & \text{if } D_{n,m}(a) > D_{n,m}(b), \\ X^{b_j}, & \text{if } D_{n,m}(a) \leq D_{n,m}(b). \end{cases} \quad (164)$$

Thus, we have

$$\begin{aligned} & |a\rangle_{n+m} |b\rangle_{n+m} |c_1\rangle |c_2\rangle |c_3\rangle |\text{anc}\rangle_2 \left( \bigotimes_{j=-m}^{n-1} X^{c_3 b_j} X^{c_2 b_j} X^{c_1 a_j} |0\rangle_{n+m} \right) \\ &= \begin{cases} |a\rangle_{n+m} |b\rangle_{n+m} |c_1\rangle |c_2\rangle |c_3\rangle |\text{anc}\rangle_2 \left( \bigotimes_{j=-m}^{n-1} X^{a_j} |0\rangle_{n+m} \right), & \text{if } D_{n,m}(a) > D_{n,m}(b), \\ |a\rangle_{n+m} |b\rangle_{n+m} |c_1\rangle |c_2\rangle |c_3\rangle |\text{anc}\rangle_2 \left( \bigotimes_{j=-m}^{n-1} X^{b_j} |0\rangle_{n+m} \right), & \text{if } D_{n,m}(a) \leq D_{n,m}(b), \end{cases} \\ &= \begin{cases} |a\rangle_{n+m} |b\rangle_{n+m} |c_1\rangle |c_2\rangle |c_3\rangle |\text{anc}\rangle_2 |a\rangle_{n+m}, & D_{n,m}(a) > D_{n,m}(b), \\ |a\rangle_{n+m} |b\rangle_{n+m} |c_1\rangle |c_2\rangle |c_3\rangle |\text{anc}\rangle_2 |b\rangle_{n+m}, & D_{n,m}(a) \leq D_{n,m}(b), \end{cases} \\ &= |a\rangle_{n+m} |b\rangle_{n+m} |c_1\rangle |c_2\rangle |c_3\rangle |\text{anc}\rangle_2 |M_{n,m}(a, b)\rangle_{n+m}. \end{aligned} \quad (165)$$

The above output is equivalent (154) where we treat  $|c_1\rangle |c_2\rangle |c_3\rangle$  as ancilla qubits. We find that the total number of elementary gates used is at most

$$\begin{aligned} & 175(2n + 2m + 1)^2 + 3 \cdot (n + m)[15 + 2 \cdot 2(3(n + m) + 5)^2] \\ & \leq 175 \cdot 2^2(n + m + 1)^2 + 3 \cdot 15(n + m) + 3(n + m) \cdot 2 \cdot 2 \cdot 5^2(n + m + 1)^2 \\ & \leq (700 + 45 + 300)(n + m + 1)^3 \\ & = 1045(n + m + 1)^3. \end{aligned} \quad (166)$$

□

**Corollary 4.21 (Quantum circuit for maximum of  $I$  numbers)** *Let  $I, n, m \in \mathbb{N}$ . Let  $M_{I,n,m} : \mathbb{F}_{n,m}^I \rightarrow \mathbb{F}_{n,m}$  be a function defined by*

$$M_{I,n,m}(i_1, \dots, i_I) = E_{n,m}(\max\{D_{n,m}(i_1), \dots, D_{n,m}(i_I)\}), \quad \forall (i_1, \dots, i_I) \in \mathbb{F}_{n,m}^I. \quad (167)$$

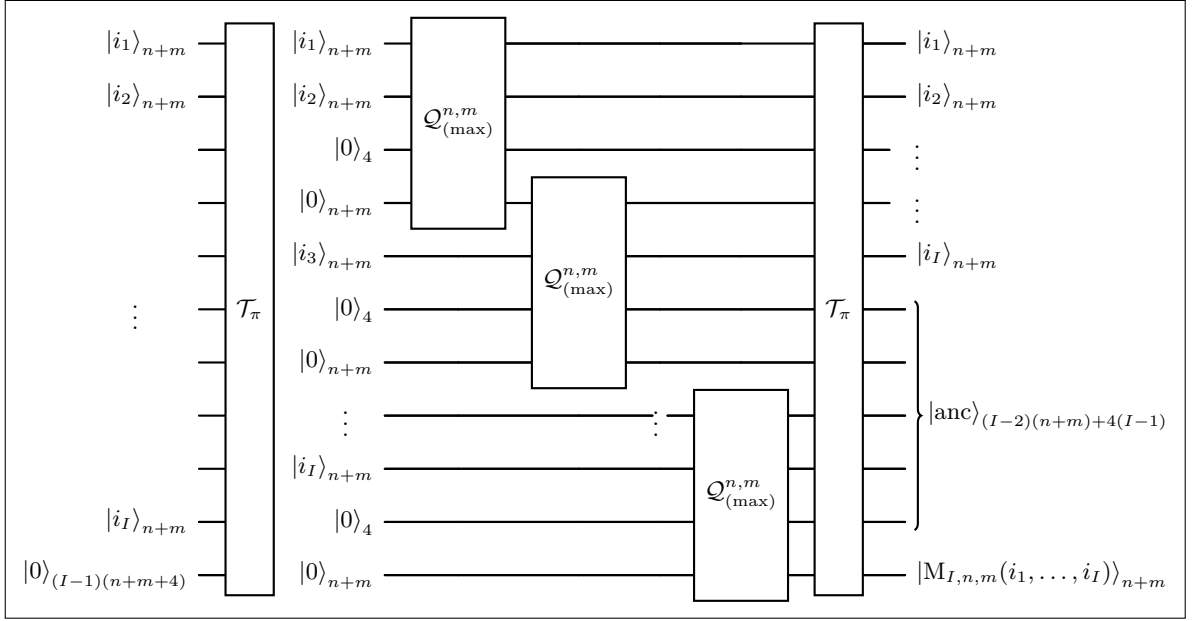


Figure 17: Circuit diagram for  $\mathcal{Q}_{(\max)}^{I,n,m}$  in Corollary 4.21.

(c.f. Definition 4.2). Then, there is a quantum circuit  $\mathcal{Q}_{(\max)}^{I,n,m}$  on  $N$  qubits, where

$$N := I(n+m) + (I-1)(n+m+5) \quad (168)$$

such that for any  $i_1, \dots, i_I \in \mathbb{F}_{n,m}$ ,

$$\begin{aligned} \mathcal{Q}_{(\max)}^{I,n,m} : & |i_1\rangle_{n+m} \cdots |i_I\rangle_{n+m} |0\rangle_{(I-1)(n+m)+5} \\ \mapsto & |i_1\rangle_{n+m} \cdots |i_I\rangle_{n+m} |\text{anc}\rangle_{(I-2)(n+m)+5(I-1)} |M_{I,n,m}(i_1, \dots, i_I)\rangle_{n+m}, \end{aligned} \quad (169)$$

which uses at most  $1189I^2(n+m+1)^3$  elementary gates. See Figure 17 for the circuit diagram.

*Proof.* For any  $i_1, \dots, i_I \in \mathbb{F}_{n,m}$ , observe that the function  $M_{I,n,m}$  can be written recursively by setting

$$\begin{aligned} \mathbf{m}_1 &:= i_1, \\ \mathbf{m}_2 &:= M_{2,n,m}(i_1, i_2), \\ \mathbf{m}_3 &:= M_{2,n,m}(\mathbf{m}_2, i_3) = M_{3,n,m}(i_1, i_2, i_3), \\ &\vdots \\ \mathbf{m}_I &:= M_{2,n,m}(\mathbf{m}_{I-1}, i_I) = \cdots = M_{I,n,m}(i_1, \dots, i_I). \end{aligned} \quad (170)$$

With this setup in mind, we construct the quantum circuit  $\mathcal{Q}_{(\max)}^{I,n,m}$  as follows.

1. We first apply the permutation quantum circuit  $\mathcal{T}_\pi$  from Lemma 4.8 to obtain

$$\begin{aligned} \mathcal{T}_\pi : & |i_1\rangle_{n+m} \cdots |i_I\rangle_{n+m} |0\rangle_{(I-1)(n+m)+5} \\ \mapsto & |i_1\rangle_{n+m} |i_2\rangle_{n+m} |0\rangle_5 |0\rangle_{n+m} |i_3\rangle_{n+m} |0\rangle_5 |0\rangle_{n+m} |i_4\rangle_{n+m} \cdots |i_I\rangle_{n+m} |0\rangle_5 |0\rangle_{n+m}. \end{aligned} \quad (171)$$

The number of elementary gates used is at most  $2N^2$ .

2. We apply inductively the maximum circuit for two TC numbers  $\mathcal{Q}_{(\max)}^{(k)}$  from Lemma 4.20 for  $k = 1, \dots, I-1$  (with  $n \leftarrow n, m \leftarrow m, a \leftarrow \mathbf{m}_k, b \leftarrow i_{k+1}$  in the notation of Lemma 4.20) so that

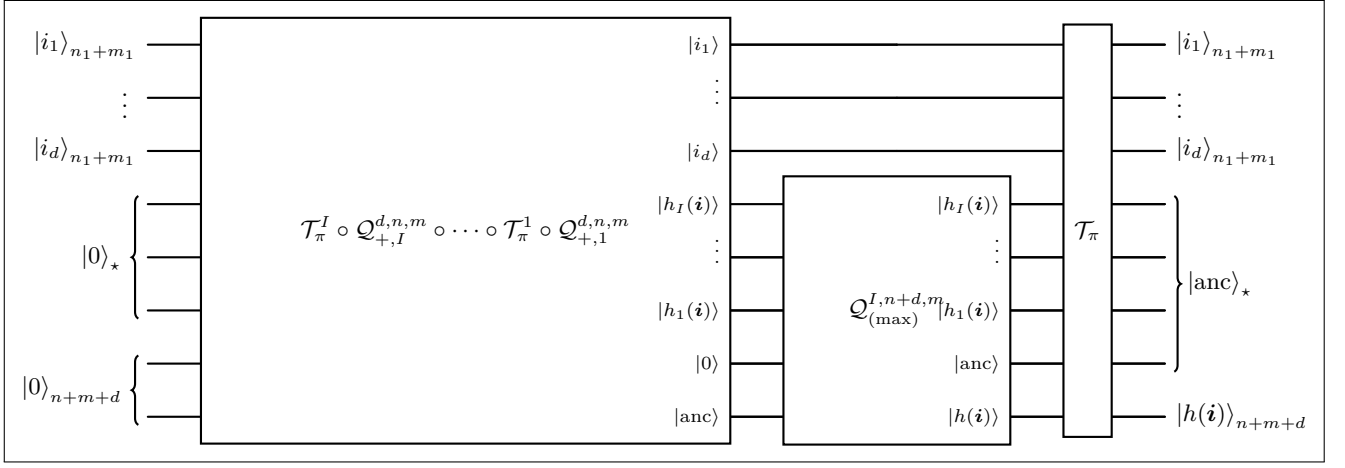


Figure 18: Circuit diagram for  $\mathcal{Q}_h$  in Proposition 4.22.

$$\begin{aligned}
& |i_1\rangle_{n+m} |i_2\rangle_{n+m} |0\rangle_5 |0\rangle_{n+m} |i_3\rangle_{n+m} |0\rangle_5 |0\rangle_{n+m} |i_4\rangle_{n+m} \cdots |i_I\rangle_{n+m} |0\rangle_5 |0\rangle_{n+m} \\
& \xrightarrow{\mathcal{Q}_{(\max)}^{(1)}} |i_1\rangle_{n+m} |i_2\rangle_{n+m} |\text{anc}\rangle_5 |M_{2,n,m}(i_1, i_2)\rangle_{n+m} |i_3\rangle_{n+m} |0\rangle_5 |0\rangle_{n+m} \cdots |i_I\rangle_{n+m} |0\rangle_5 |0\rangle_{n+m} \\
& = |i_1\rangle_{n+m} |i_2\rangle_{n+m} |\text{anc}\rangle_5 |\mathbf{m}_2\rangle_{n+m} |i_3\rangle_{n+m} |0\rangle_5 |0\rangle_{n+m} \cdots |i_I\rangle_{n+m} |0\rangle_5 |0\rangle_{n+m} \\
& \xrightarrow{\mathcal{Q}_{(\max)}^{(2)}} |i_1\rangle_{n+m} |i_2\rangle_{n+m} |\text{anc}\rangle_5 |\mathbf{m}_2\rangle_{n+m} |i_3\rangle_{n+m} |\text{anc}\rangle_5 |M_{2,n,m}(\mathbf{m}_2, i_3)\rangle_{n+m} |i_4\rangle_{n+m} |0\rangle_5 |0\rangle_{n+m} \\
& \quad \cdots |i_I\rangle_{n+m} |0\rangle_5 |0\rangle_{n+m} \\
& =: |i_1\rangle_{n+m} |i_2\rangle_{n+m} |\text{anc}\rangle_5 |\mathbf{m}_2\rangle_{n+m} |i_3\rangle_{n+m} |\text{anc}\rangle_5 |\mathbf{m}_3\rangle_{n+m} |i_4\rangle_{n+m} |0\rangle_5 |0\rangle_{n+m} \\
& \quad \cdots |i_I\rangle_{n+m} |0\rangle_5 |0\rangle_{n+m} \\
& \quad \vdots \quad \quad \quad \vdots \\
& \xrightarrow{\mathcal{Q}_{(\max)}^{(I)}} |i_1\rangle_{n+m} |i_2\rangle_{n+m} |\text{anc}\rangle_5 |\mathbf{m}_2\rangle_{n+m} |i_3\rangle_{n+m} \cdots |\mathbf{m}_{I-1}\rangle_{n+m} |i_I\rangle_{n+m} |\text{anc}\rangle_5 |M_{2,n,m}(\mathbf{m}_{I-1}, i_I)\rangle_{n+m} \\
& =: |i_1\rangle_{n+m} |i_2\rangle_{n+m} |\text{anc}\rangle_5 |\mathbf{m}_2\rangle_{n+m} |i_3\rangle_{n+m} \cdots |\mathbf{m}_{I-1}\rangle_{n+m} |i_I\rangle_{n+m} |\text{anc}\rangle_5 |\mathbf{m}_I\rangle_{n+m}.
\end{aligned} \tag{172}$$

The number of elementary gates used in this step is at most

$$I \cdot 1045(n+m+1)^3. \tag{173}$$

3. We consolidate the ancillary qubits by combining the following qubits:  $(I-1)$  times of  $|\text{anc}\rangle_4$ , and  $(I-1)$  times of  $|\mathbf{m}_2\rangle_{n+m}, \dots, |\mathbf{m}_{I-1}\rangle_{n+m}$  under the placeholder qubit  $|\text{anc}\rangle_{(I-2)(n+m)+4(I-1)}$ . The permutation quantum circuit  $\mathcal{T}_\pi$  from Lemma 4.8 performs the following operation

$$\begin{aligned}
\mathcal{T}_\pi : & |i_1\rangle_{n+m} |i_2\rangle_{n+m} |\text{anc}\rangle_5 |\mathbf{m}_2\rangle_{n+m} |i_3\rangle_{n+m} \cdots |\mathbf{m}_{I-1}\rangle_{n+m} |i_I\rangle_{n+m} |\text{anc}\rangle_5 |\mathbf{m}_I\rangle_{n+m} \\
& \mapsto |i_1\rangle_{n+m} |i_2\rangle_{n+m} \cdots |i_I\rangle_{n+m} |\text{anc}\rangle_{(I-2)(n+m)+5(I-1)} |\mathbf{m}_I\rangle_{n+m}.
\end{aligned} \tag{174}$$

The number of elementary gates used in this step is at most  $2N^2$ .

The resulting quantum circuit  $\mathcal{Q}_{(\max)}^{I,n,m}$  is a composition of the quantum circuits from each of the above steps. The number of elementary gates used in total is the sum of the number of gates used in each step which is at most

$$\begin{aligned}
& 2N^2 + 1045I(n+m+1)^3 + 2N^2 \\
& = 1045I(n+m+1)^3 + 4[I(n+m) + (I-1)(n+m+5)]^2 \\
& \leq 1045I(n+m+1)^3 + 4[I(n+m+1) + 5I(n+m+1)]^2 \\
& \leq 1045I(n+m+1)^3 + 4 \cdot 6^2(I(n+m+1))^2 \\
& \leq 1189I^2(n+m+1)^3.
\end{aligned} \tag{175}$$

□

**Proposition 4.22 (Quantum circuit for loading CPWA component functions)** Let  $I, d, n_1, n_2, m_1, m_2 \in \mathbb{N}$ . Define  $n := n_1 + n_2$ ,  $m := m_1 + m_2$ , and  $p := d(2n_2 + 2m_2 + 3) + (n_2 + m_2) + (d - 1)(n + m)$ . Let  $\{a_{l,j}\}_{l=1,\dots,I;j=1,\dots,d}, \{b_l\}_{l=1,\dots,I} \subset \mathbb{F}_{n_2, m_2}$ . Let  $h_l : \mathbb{F}_{n_1, m_1}^d \rightarrow \mathbb{F}_{n+d, m}$ ,  $l = 1, \dots, I$  be functions defined by

$$h_l(i_1, \dots, i_d) = \bigoplus_{j=1}^d (a_{l,j} \boxplus i_j) \boxplus b_l, \quad \forall (i_1, \dots, i_d) \in \mathbb{F}_{n_1, m_1}^d, \quad (176)$$

let  $M_{I, n+d, m} : \mathbb{F}_{n+d, m}^I \rightarrow \mathbb{F}_{n+d, m}$  be a function defined by

$$M_{I, n+d, m}(i_1, \dots, i_I) = E_{n+d, m}(\max\{D_{n+d, m}(i_1), \dots, D_{n+d, m}(i_I)\}), \quad \forall (i_1, \dots, i_I) \in \mathbb{F}_{n+d, m}^I. \quad (177)$$

Define  $h : \mathbb{F}_{n_1, m_1}^d \rightarrow \mathbb{F}_{n+d, m}$  by

$$h(\mathbf{i}) := M_{I, n+d, m}(h_1(\mathbf{i}), \dots, h_I(\mathbf{i})), \quad \forall \mathbf{i} = (i_1, \dots, i_d) \in \mathbb{F}_{n_1, m_1}^d. \quad (178)$$

Then, there is a quantum circuit  $\mathcal{Q}_h$  on  $N$  qubits, where

$$N := d(n_1 + m_1) + I(n + m + d + p) + (I - 1)(n + m + d + 5), \quad (179)$$

such that for any  $\mathbf{i} = (i_1, \dots, i_d) \in \mathbb{F}_{n_1, m_1}^d$ ,

$$\begin{aligned} \mathcal{Q}_h : & |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |0\rangle_{I(n+m+d+p)+(I-2)(n+m+d)+5(I-1)} |0\rangle_{n+m+d} \\ & \mapsto |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |\text{anc}\rangle_{I(n+m+d+p)+(I-2)(n+m+d)+5(I-1)} |h(\mathbf{i})\rangle_{n+m+d}, \end{aligned} \quad (180)$$

which uses at most  $10651I^3d^3(n + m + 1)^3$  elementary gates. See Figure 18 for the circuit diagram.

*Proof.* The construction of the quantum circuit  $\mathcal{Q}_h$  involves the following steps:

1. We first prepare the  $I$  affine sums  $h_l(i_1, \dots, i_d)$  from (176), using Lemma 4.19.

For  $l = 1, \dots, I$ , we apply the quantum circuits  $(\mathcal{Q}_{+,l}^{d,n,m})_{l=1,\dots,I}$  of Lemma 4.19 (with  $(d, n_1, n_2, m_1, m_2, a_1, \dots, a_d, b) \leftarrow (d, n_1, n_2, m_1, m_2, a_{l,1}, \dots, a_{l,d}, b_l)$  in the notation of Lemma 4.19) followed by an application of the permutation circuit  $\mathcal{T}_\pi$  from Lemma 4.8, where we compute

$$\begin{aligned} & |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |0\rangle_{I(n+m+d+p)} |0\rangle_{(I-1)(n+m+d+5)} \\ & \xrightarrow{\mathcal{Q}_{+,1}^{d,n,m}} |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |h_1(\mathbf{i})\rangle_{n+m+d} |\text{anc}\rangle_p |0\rangle_{(I-1)(n+m+d+p)} |0\rangle_{(I-1)(n+m+d+5)} \\ & \xrightarrow{\mathcal{T}_\pi^1} |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |0\rangle_{n+m+d+p} |h_1(\mathbf{i})\rangle_{n+m+d} |\text{anc}\rangle_p |0\rangle_{(I-2)(n+m+d+p)} |0\rangle_{(I-1)(n+m+d+5)} \\ & \xrightarrow{\mathcal{Q}_{+,2}^{d,n,m}} |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |h_2(\mathbf{i})\rangle_{n+m+d} |\text{anc}\rangle_p |h_1(\mathbf{i})\rangle_{n+m+d} |\text{anc}\rangle_p \\ & \quad \cdot |0\rangle_{(I-2)(n+m+d+p)} |0\rangle_{(I-1)(n+m+d+5)} \\ & \xrightarrow{\mathcal{T}_\pi^2} |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |0\rangle_{n+m+d+p} |h_2(\mathbf{i})\rangle_{n+m+d} |h_1(\mathbf{i})\rangle_{n+m+d} |\text{anc}\rangle_{2p} \\ & \quad \cdot |0\rangle_{(I-3)(n+m+d+p)} |0\rangle_{(I-1)(n+m+d+5)} \\ & \quad \vdots \\ & \xrightarrow{\mathcal{Q}_{+,I}^{d,n,m}} |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |h_I(\mathbf{i})\rangle_{n+m+d} |\text{anc}\rangle_p \\ & \quad \cdots |h_2(\mathbf{i})\rangle_{n+m+d} |h_1(\mathbf{i})\rangle_{n+m+d} |\text{anc}\rangle_{(I-1)p} |0\rangle_{(I-1)(n+m+d+5)} \\ & \xrightarrow{\mathcal{T}_\pi^I} |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |h_I(\mathbf{i})\rangle_{n+m+d} \cdots |h_2(\mathbf{i})\rangle_{n+m+d} |h_1(\mathbf{i})\rangle_{n+m+d} \\ & \quad \cdot |0\rangle_{(I-1)(n+m+d+5)} |\text{anc}\rangle_{Ip}. \end{aligned} \quad (181)$$

In this step, the number of elementary gates used is at most

$$I[2N^2 + 563d^3(n + m + 1)^2]. \quad (182)$$

2. Next, we compute the maximum value amongst the affine sums  $h_1, \dots, h_I$ . We apply the quantum circuit  $\mathcal{Q}_{(\max)}^{I, n+d, m}$  of Corollary 4.21 (with  $I \leftarrow I$ ,  $n \leftarrow d + n$ ,  $m \leftarrow m$ ,  $i_1, \dots, i_I \leftarrow h_1(\mathbf{i}), \dots, h_I(\mathbf{i})$  in the notation of Corollary 4.21) where we have

$$\begin{aligned} & |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |h_I(\mathbf{i})\rangle_{n+m+d} \cdots |h_2(\mathbf{i})\rangle_{n+m+d} |h_1(\mathbf{i})\rangle_{n+m+d} \\ & \quad \cdot |0\rangle_{(I-1)(n+m+d+5)} |\text{anc}\rangle_{Ip} \\ & \xrightarrow{\mathcal{Q}_{(\max)}^{I, n+d, m}} |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |h_I(\mathbf{i})\rangle_{n+m+d} \cdots |h_2(\mathbf{i})\rangle_{n+m+d} |h_1(\mathbf{i})\rangle_{n+m+d} \\ & \quad |\text{anc}\rangle_{(I-2)(n+m+d)+5(I-1)} |h(\mathbf{i})\rangle_{n+m+d} |\text{anc}\rangle_{Ip} \end{aligned} \quad (183)$$

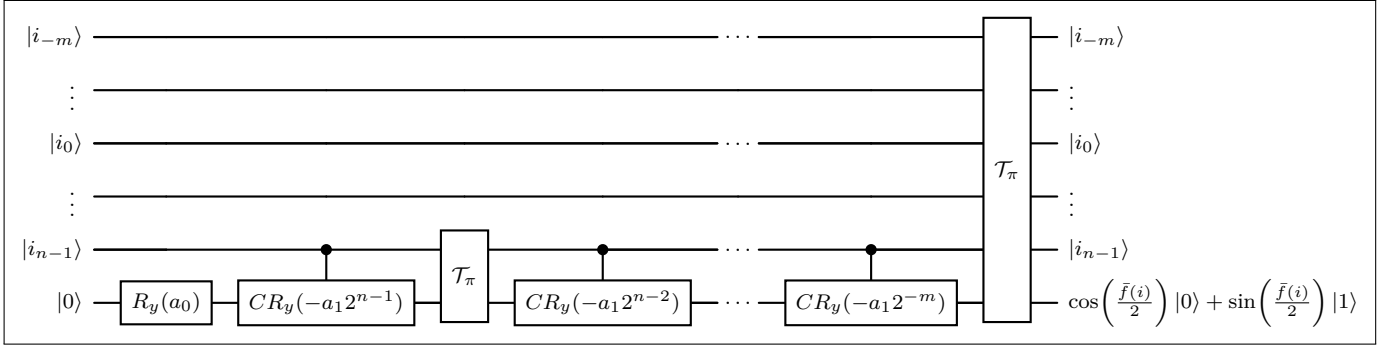


Figure 19: Circuit diagram for  $\mathcal{R}_f$  in Lemma 4.23.

In this step, the number of elementary gates used is at most

$$1189I^2(n + m + d + 1)^3. \quad (184)$$

3. We use the permutation quantum circuit  $\mathcal{T}_\pi$  from Lemma 4.8 and we put the qubits  $|h_I(\mathbf{i})\rangle_{n+m+d} \cdots |h_1(\mathbf{i})\rangle_{n+m+d}$  under  $|\text{anc}\rangle$ , so that we have

$$\begin{aligned} & |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |h_I(\mathbf{i})\rangle_{n+m+d} \cdots |h_1(\mathbf{i})\rangle_{n+m+d} \\ & \quad \cdot |\text{anc}\rangle_{(I-2)(n+m+d)+5(I-1)} |h(\mathbf{i})\rangle_{n+m+d} |\text{anc}\rangle_{Ip} \\ & \xrightarrow{\mathcal{T}_\pi} |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |\text{anc}\rangle_{I(n+m+d+p)+(I-2)(n+m+d)+5(I-1)} |h(\mathbf{i})\rangle_{n+m+d}. \end{aligned} \quad (185)$$

We hence reach the desired state (180). In this step, the number of elementary gates used is at most

$$2N^2. \quad (186)$$

We note that

$$p = d(2n_2 + 2m_2 + 3) + (n_2 + m_2) + (d - 1)(n + m) \leq 4d(n + m + 1). \quad (187)$$

Hence, we obtain that

$$\begin{aligned} N &= d(n_1 + m_1) + I(n + m + d + p) + (I - 1)(n + m + d + 5) \\ &\leq d(n + m + 1) + Id(n + m + 1) + Ip + (d + 5)I(n + m + 1) \\ &\leq (1 + 1 + 4 + 1 + 5)Id(n + m + 1) \\ &= 12Id(n + m + 1). \end{aligned} \quad (188)$$

Thus, the total number of elementary gates used is at most

$$\begin{aligned} & I[2N^2 + 563d^3(n + m + 1)^2] + 1189I^2(n + m + d + 1)^3 + 2N^2 \\ &= (I + 1)2N^2 + 563Id^3(n + m + 1)^2 + 1189I^2(n + m + d + 1)^3 \\ &\leq 4IN^2 + 563Id^3(n + m + 1)^2 + 1189I^2(2d)^3(n + m + 1)^3 \\ &\leq 4I(12Id(n + m + 1))^2 + 563Id^3(n + m + 1)^2 + 1189I^2(2d)^3(n + m + 1)^3 \\ &\leq (4 \cdot 12^2 + 563 + 1189 \cdot 2^3)I^3d^3(n + m + 1)^3 \\ &= 10651I^3d^3(n + m + 1)^3. \end{aligned} \quad (189)$$

□

**Lemma 4.23 (Quantum circuit for  $Y$ -rotation)** Let  $n \in \mathbb{N}$ ,  $m \in \mathbb{N}_0$ ,  $a_0, a_1 \in \mathbb{R}$ , and let  $f(x) = a_1x + a_0$  for  $x \in \mathbb{R}$ . Define

$$\bar{f}(i) := f \circ D_{n,m}(i), \quad \forall i \in \mathbb{F}_{n,m}. \quad (190)$$

Then, there is a quantum circuit  $\mathcal{R}_f$  on  $(n + m + 1)$  qubits such that for any  $i \in \mathbb{F}_{n,m}$ ,

$$\mathcal{R}_f : |i\rangle_{n+m} |0\rangle \mapsto |i\rangle_{n+m} [\cos(\bar{f}(i)/2) |0\rangle + \sin(\bar{f}(i)/2) |1\rangle], \quad (191)$$

which uses  $13(n + m + 1)^3$  elementary gates. See Figure 19 for the circuit diagram.

*Proof.* The quantum circuit  $\mathcal{R}_f$  is constructed from the following steps:

1. We apply the  $Y$ -rotation gate  $R_y(\theta)$  from Example 2.9 (with parameter  $\theta \leftarrow a_0$  in the notation of Example 2.9) to obtain the state

$$I_2^{\otimes n+m} \otimes R_y(a_0) : |i\rangle_{n+m} |0\rangle \mapsto |i\rangle_{n+m} (\cos(a_0/2) |0\rangle + \sin(a_0/2) |1\rangle). \quad (192)$$

2. We apply the controlled  $Y$ -rotation gate  $CR_y(\theta)$  of Lemma 4.15 on the qubit  $|i_{n-1}\rangle$  (with parameter  $\theta \leftarrow -a_1 \cdot 2^{n-1}$ , control qubit  $c \leftarrow i_{n-1}$  and target qubit  $0 \leftarrow (\cos(a_0/2) |0\rangle + \sin(a_0/2) |1\rangle)$  in the notation of Lemma 4.15) to obtain

$$\begin{aligned} & I_2^{\otimes n+m-1} \otimes CR_y(-a_1 \cdot 2^{n-1}) : |i_{-m}\rangle \cdots |i_0\rangle \cdots |i_{n-1}\rangle (\cos(a_0/2) |0\rangle + \sin(a_0/2) |1\rangle) \\ & \mapsto |i_{-m}\rangle \cdots |i_0\rangle \cdots |i_{n-1}\rangle (\cos((-a_1 2^{n-1} i_{n-1} + a_0)/2) |0\rangle + \sin((-a_1 2^{n-1} i_{n-1} + a_0)/2) |1\rangle) \\ & = \begin{cases} |i\rangle_{n+m} (\cos(a_0/2) |0\rangle + \sin(a_0/2) |1\rangle), & \text{if } i_{n-1} = 0, \\ |i\rangle_{n+m} (\cos((-a_1 2^{n-1} + a_0)/2) |0\rangle + \sin((-a_1 2^{n-1} + a_0)/2) |1\rangle), & \text{if } i_{n-1} = 1. \end{cases} \end{aligned} \quad (193)$$

3. Similarly, we inductively apply for  $k = n-2, \dots, 0, \dots, -m$  the permutation quantum circuit  $\mathcal{T}_{k \leftrightarrow n-1}$  from Lemma 4.8, the controlled  $Y$ -rotation gate  $CR_y(\theta)$  (with parameter  $\theta \leftarrow a_1 2^k$  and control qubit  $c \leftarrow i_k$  in the notation of Lemma 4.15), and another permutation circuit  $\mathcal{T}_{k \leftrightarrow n-1}$  to obtain

$$\begin{aligned} & |i_{-m}\rangle \cdots |i_0\rangle \cdots |i_{n-1}\rangle (\cos((-a_1 2^{n-1} i_{n-1} + a_0)/2) |0\rangle + \sin((-a_1 2^{n-1} i_{n-1} + a_0)/2) |1\rangle) \\ & \xrightarrow{\mathcal{T}_{n-2 \leftrightarrow n-1}} |i_{-m}\rangle \cdots |i_0\rangle \cdots |i_{n-1}\rangle |i_{n-2}\rangle (\cos((-a_1 2^{n-1} i_{n-1} + a_0)/2) |0\rangle + \sin((-a_1 2^{n-1} i_{n-1} + a_0)/2) |1\rangle) \\ & \xrightarrow{CR_y(a_1 2^{n-2})} |i_{-m}\rangle \cdots |i_0\rangle \cdots |i_{n-1}\rangle |i_{n-2}\rangle \left( \cos((a_1 (2^{n-2} i_{n-2} - 2^{n-1} i_{n-1}) + a_0)/2) |0\rangle \right. \\ & \quad \left. + \sin((a_1 (2^{n-2} i_{n-2} - 2^{n-1} i_{n-1}) + a_0)/2) |1\rangle \right) \\ & \xrightarrow{\mathcal{T}_{n-2 \leftrightarrow n-1}} |i_{-m}\rangle \cdots |i_0\rangle \cdots |i_{n-2}\rangle |i_{n-1}\rangle \left( \cos((a_1 (2^{n-2} i_{n-2} - 2^{n-1} i_{n-1}) + a_0)/2) |0\rangle \right. \\ & \quad \left. + \sin((a_1 (2^{n-2} i_{n-2} - 2^{n-1} i_{n-1}) + a_0)/2) |1\rangle \right) \\ & \quad \vdots \\ & \mapsto |i\rangle_{n+m} \left( \cos \left( (a_1 (-2^{n-1} i_{n-1} + \sum_{k=-m}^{n-2} 2^k i_k) + a_0)/2 \right) |0\rangle + \sin \left( (a_1 (-2^{n-1} i_{n-1} + \sum_{k=-m}^{n-2} 2^k i_k) + a_0)/2 \right) |1\rangle \right) \\ & = |i\rangle_{n+m} [\cos(\bar{f}(i)/2) |0\rangle + \sin(\bar{f}(i)/2) |1\rangle] \end{aligned} \quad (194)$$

The permutation circuits  $\mathcal{T}_\pi$  from Lemma 4.8 requires  $2(n+m+1)^2$  gates and the controlled  $Y$ -rotation gate  $CR_y(\theta)$  requires 4 elementary gates (c.f. Lemma 4.15). Hence, the total number of gates required for circuit  $\mathcal{R}_f$  is

$$1 + 4 + (n+m-1)[2 \cdot 2(n+m+1)^2 + 4] \leq 13(n+m+1)^3. \quad (195)$$

Thus, we conclude the proof of the lemma.  $\square$

**Proposition 4.24 (Quantum circuit for CPWA payoff function with  $Y$ -rotation)** *Let  $d \in \mathbb{N}$ ,  $n_1, n_2, m_1, m_2 \in \mathbb{N}$ ,  $K \in \mathbb{N}$ ,  $I_1, \dots, I_K \in \mathbb{N}$ ,  $\xi_1, \dots, \xi_K \in \{-1, 1\}$ , and  $s \in (0, 1)$ . Define  $n := n_1 + n_2$ ,  $m = m_1 + m_2$ , and  $p := d(2n_2 + 2m_2 + 3) + (n_2 + m_2) + (d-1)(n+m)$ . Let  $\{a_{k,l,j}\}_{k=1,\dots,K;l=1,\dots,I_k;j=1,\dots,d}$ ,  $\{b_{k,l}\}_{k=1,\dots,K;l=1,\dots,I_k} \subset \mathbb{F}_{n_2, m_2}$ . For  $k = 1, \dots, K$ ,  $l = 1, \dots, I_k$ , let  $h_{k,l} : \mathbb{F}_{n_1, m_1}^d \rightarrow \mathbb{F}_{n+d, m}$  be functions defined by*

$$h_{k,l}(\mathbf{i}) := \bigoplus_{j=1}^d (a_{k,l,j} \boxplus i_j) \boxplus b_{k,l}, \quad \forall \mathbf{i} = (i_1, \dots, i_d) \in \mathbb{F}_{n_1, m_1}^d. \quad (196)$$

For  $k = 1, \dots, K$ , let  $\bar{\xi}_k \in \mathbb{F}_{2,0}$  be defined by

$$\bar{\xi}_k := E_{2,0}(\xi_k), \quad (197)$$

let  $M_{I_k, n+d, m} : \mathbb{F}_{n+d, m}^{I_k} \rightarrow \mathbb{F}_{n+d, m}$  be defined by

$$M_{I_k, n+d, m}(i_1, \dots, i_{I_k}) := E_{n+d, m}(\max\{D_{n+d, m}(i_1), \dots, D_{n+d, m}(i_{I_k})\}), \quad \forall (i_1, \dots, i_{I_k}) \in \mathbb{F}_{n+d, m}^{I_k}, \quad (198)$$

and let  $\mathbf{h}_k : \mathbb{F}_{n_1, m_1}^d \rightarrow \mathbb{F}_{n+d, m}$  be defined by

$$\mathbf{h}_k(\mathbf{i}) := M_{I_k, n+d, m}(h_{k,1}(\mathbf{i}), \dots, h_{k,I_k}(\mathbf{i})), \quad \forall \mathbf{i} = (i_1, \dots, i_d) \in \mathbb{F}_{n_1, m_1}^d. \quad (199)$$

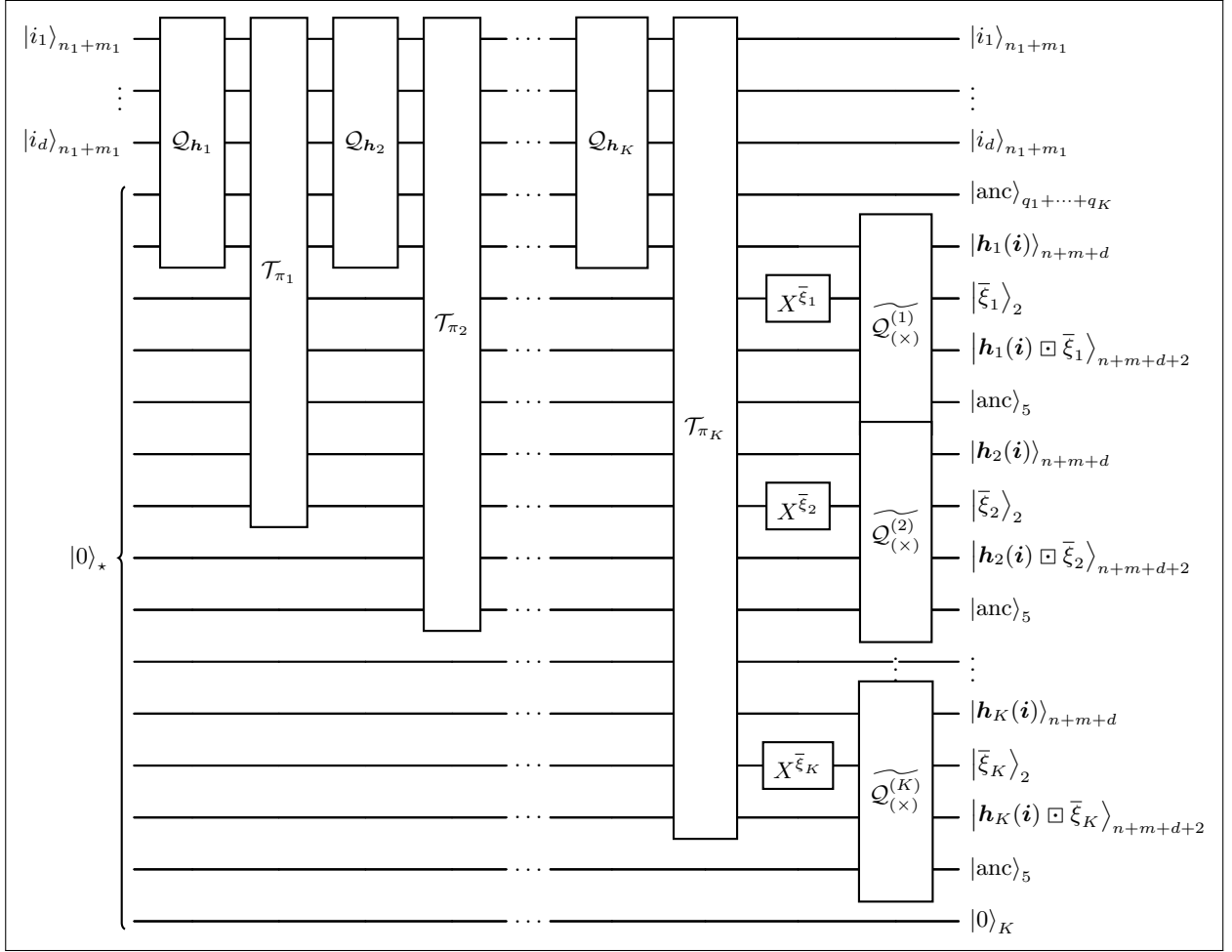


Figure 20: Circuit diagram for  $\mathcal{R}_h$  in Proposition 4.24 (Steps 1–3).

Let  $\mathbf{h} : \mathbb{F}_{n_1, m_1}^d \rightarrow \mathbb{F}_{n+d+K+1, m}$  be defined by

$$\mathbf{h}(\mathbf{i}) := \bigoplus_{k=1}^K (\bar{\xi}_k \boxplus \mathbf{h}_k(\mathbf{i})), \quad \forall \mathbf{i} = (i_1, \dots, i_d) \in \mathbb{F}_{n_1, m_1}^d. \quad (200)$$

Let  $f : \mathbb{R} \rightarrow \mathbb{R}$  be a function defined by  $f(x) = sx + \frac{\pi}{2}$ , and define  $\bar{f} : \mathbb{F}_{n+d+K+1, m} \rightarrow \mathbb{R}$  by

$$\bar{f}(i) := f \circ D_{n+d+K+1, m}(i), \quad \forall i \in \mathbb{F}_{n+d+K+1, m}. \quad (201)$$

Then, there is a quantum circuit  $\mathcal{R}_h$  on  $N$  qubits, where

$$\begin{aligned} N &:= d(n_1 + m_1) + \sum_{k=1}^K q_k + 2K(n + m + d + 5), \\ q_k &:= I_k(n + m + d + p) + (I_k - 2)(n + m + d) + 5(I_k - 1), \quad k = 1, \dots, K, \end{aligned} \quad (202)$$

such that for any  $\mathbf{i} = (i_1, \dots, i_d) \in \mathbb{F}_{n_1, m_1}^d$ ,

$$\begin{aligned} \mathcal{R}_h : & |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |0\rangle_{q_1+\dots+q_K} |0\rangle_{2K(n+m+d+5)} \\ & \mapsto |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |\text{anc}\rangle_{q_1+\dots+q_K+2K(n+m+d+5)-1} [\cos(\bar{f}(\mathbf{h}(\mathbf{i}))/2) |0\rangle + \sin(\bar{f}(\mathbf{h}(\mathbf{i}))/2) |1\rangle], \end{aligned} \quad (203)$$

which uses at most  $16186K^3 (\max_{k=1, \dots, K} \{I_k\})^3 d^3 (n + m + 1)^3$  elementary gates. See Figure 20 and Figure 21 for the circuit diagram.

*Proof.* The construction of the quantum circuit  $\mathcal{R}_h$  involves the following steps:

1. We first prepare the  $K$  component functions  $\mathbf{h}_k$  using Proposition 4.22. For  $k = 1, \dots, K$ , we apply the quantum circuits  $(\mathcal{Q}_{h_k})_{k=1, \dots, K}$  of Proposition 4.22 (with  $I, d, n_1, m_1, n_2, m_2 \leftarrow I_k, d, n_1, m_1, n_2, m_2, a_{l,j} \leftarrow$



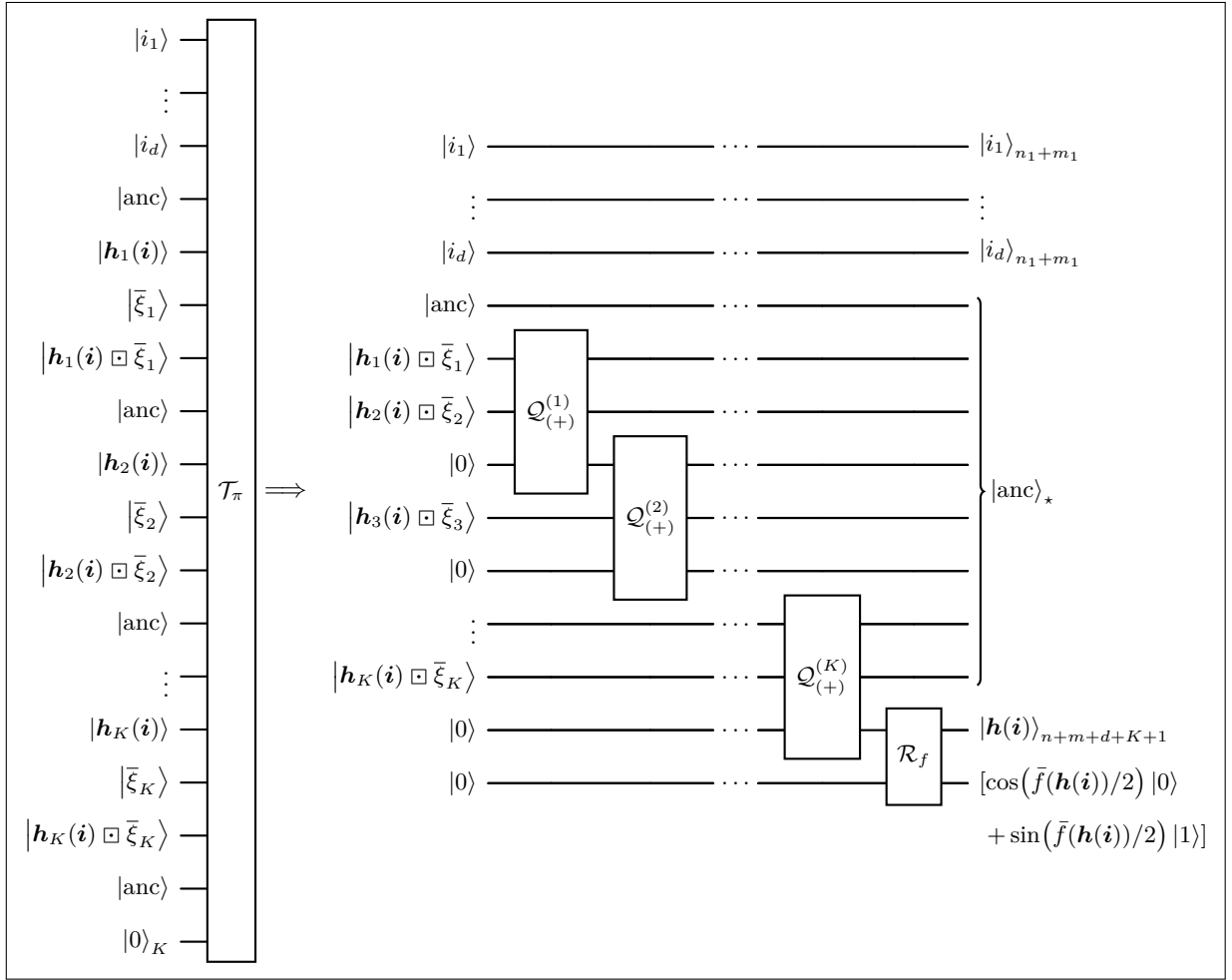


Figure 21: Circuit diagram for  $\mathcal{R}_h$  in Proposition 4.24 (Steps 4–6).

$a_{k,l,j}$ , and  $b_l \leftarrow b_{k,l}$  in the notation of Proposition 4.22) followed by an application of the permutation circuit  $\mathcal{T}_\pi$  from Lemma 4.8, where we compute

$$\begin{aligned}
& |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |0\rangle_{q_1+\dots+q_K} |0\rangle_{2K(n+m+d+5)} \\
&= |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |0\rangle_{q_1} |0\rangle_{n+m+d} |0\rangle_{q_2} |0\rangle_{n+m+d} \cdots |0\rangle_{q_K} |0\rangle_{n+m+d} |0\rangle_{K(n+m+d+10)} \\
&\xrightarrow{\mathcal{Q}_{h_1}} |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |\text{anc}\rangle_{q_1} |\mathbf{h}_1(\mathbf{i})\rangle_{n+m+d} |0\rangle_{q_2} |0\rangle_{n+m+d} \cdots |0\rangle_{q_K} |0\rangle_{n+m+d} |0\rangle_{K(n+m+d+10)} \\
&\xrightarrow{\mathcal{T}_{\pi_1}} |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |0\rangle_{q_2} |0\rangle_{n+m+d} |\text{anc}\rangle_{q_1} |\mathbf{h}_1(\mathbf{i})\rangle_{n+m+d} \cdots |0\rangle_{q_K} |0\rangle_{n+m+d} |0\rangle_{K(n+m+d+10)} \\
&\xrightarrow{\mathcal{Q}_{h_2}} |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |\text{anc}\rangle_{q_2} |\mathbf{h}_2(\mathbf{i})\rangle_{n+m+d} |\text{anc}\rangle_{q_1} |\mathbf{h}_1(\mathbf{i})\rangle_{n+m+d} \cdots |0\rangle_{q_K} |0\rangle_{n+m+d} |0\rangle_{K(n+m+d+10)} \\
&\quad \vdots \\
&\xrightarrow{\mathcal{Q}_{h_K}} |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |\text{anc}\rangle_{q_K} |\mathbf{h}_K(\mathbf{i})\rangle_{n+m+d} \cdots |\text{anc}\rangle_{q_1} |\mathbf{h}_1(\mathbf{i})\rangle_{n+m+d} |0\rangle_{K(n+m+d+10)} \\
&\xrightarrow{\mathcal{T}_{\pi_K}} |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |\text{anc}\rangle_{q_1+\dots+q_K} |\mathbf{h}_1(\mathbf{i})\rangle_{n+m+d} |0\rangle_2 |0\rangle_{n+m+d+2} |0\rangle_5 \\
&\quad \cdot |\mathbf{h}_2(\mathbf{i})\rangle_{n+m+d} |0\rangle_2 |0\rangle_{n+m+d+2} |0\rangle_5 \cdots |\mathbf{h}_K(\mathbf{i})\rangle_{n+m+d} |0\rangle_2 |0\rangle_{n+m+d+2} |0\rangle_5 |0\rangle_K
\end{aligned} \tag{204}$$

In this step, the number of elementary gates used is at most

$$K \cdot 2N^2 + \sum_{k=1}^K 10651 I_k^3 d^3 (n+m+1)^3 \tag{205}$$

2. Next, we load the strings  $\bar{\xi}_k \in \mathbb{F}_{2,0}$  into the qubits  $|0\rangle_2$  for  $k = 1, \dots, K$  using the Pauli  $X$  gates (c.f. Example

2.6)

$$\begin{aligned}
& |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |\text{anc}\rangle_{q_1+\dots+q_K} |\mathbf{h}_1(\mathbf{i})\rangle_{n+m+d} |0\rangle_2 |0\rangle_{n+m+d+2} |0\rangle_5 \\
& \quad \cdot |\mathbf{h}_2(\mathbf{i})\rangle_{n+m+d} |0\rangle_2 |0\rangle_{n+m+d+2} |0\rangle_5 \cdots |\mathbf{h}_K(\mathbf{i})\rangle_{n+m+d} |0\rangle_2 |0\rangle_{n+m+d+2} |0\rangle_5 |0\rangle_K \\
& \mapsto |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |\text{anc}\rangle_{q_1+\dots+q_K} |\mathbf{h}_1(\mathbf{i})\rangle_{n+m+d} |\bar{\xi}_1\rangle_2 |0\rangle_{n+m+d+2} |0\rangle_5 \\
& \quad \cdot |\mathbf{h}_2(\mathbf{i})\rangle_{n+m+d} |\bar{\xi}_2\rangle_2 |0\rangle_{n+m+d+2} |0\rangle_5 \cdots |\mathbf{h}_K(\mathbf{i})\rangle_{n+m+d} |\bar{\xi}_K\rangle_2 |0\rangle_{n+m+d+2} |0\rangle_5 |0\rangle_K.
\end{aligned} \tag{206}$$

The number of elementary gates used in this step is at most

$$2K. \tag{207}$$

3. We perform  $K$  multiplication using the quantum circuit  $\tilde{\mathcal{Q}}_{(\times)}^{(k)} = \tilde{\mathcal{Q}}_{(\times)}$  from Corollary 4.12 (with  $n_1 \leftarrow n+d$ ,  $m_1 \leftarrow m$ ,  $n_2 \leftarrow 2$ ,  $m_2 \leftarrow 0$ ,  $a \leftarrow \mathbf{h}_k(\mathbf{i})$ , and  $b \leftarrow \bar{\xi}_k$  in the notation of Corollary 4.12 for  $k = 1, \dots, K$ )

$$\begin{aligned}
& |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |\text{anc}\rangle_{q_1+\dots+q_K} |\mathbf{h}_1(\mathbf{i})\rangle_{n+m+d} |\bar{\xi}_1\rangle_2 |0\rangle_{n+m+d+2} |0\rangle_5 \\
& \quad \cdot |\mathbf{h}_2(\mathbf{i})\rangle_{n+m+d} |\bar{\xi}_2\rangle_2 |0\rangle_{n+m+d+2} |0\rangle_5 \cdots |\mathbf{h}_K(\mathbf{i})\rangle_{n+m+d} |\bar{\xi}_K\rangle_2 |0\rangle_{n+m+d+2} |0\rangle_5 |0\rangle_K \\
& \xrightarrow{\tilde{\mathcal{Q}}_{(\times)}^{(1)}} |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |\text{anc}\rangle_{q_1+\dots+q_K} |\mathbf{h}_1(\mathbf{i})\rangle_{n+m+d} |\bar{\xi}_1\rangle_2 |\mathbf{h}_1(\mathbf{i}) \boxminus \bar{\xi}_1\rangle_{n+m+d+2} |\text{anc}\rangle_5 \\
& \quad \cdot |\mathbf{h}_2(\mathbf{i})\rangle_{n+m+d} |\bar{\xi}_2\rangle_2 |0\rangle_{n+m+d+2} |0\rangle_5 \cdots |\mathbf{h}_K(\mathbf{i})\rangle_{n+m+d} |\bar{\xi}_K\rangle_2 |0\rangle_{n+m+d+2} |0\rangle_5 |0\rangle_K \\
& \quad \vdots \\
& \xrightarrow{\tilde{\mathcal{Q}}_{(\times)}^{(K)}} |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |\text{anc}\rangle_{q_1+\dots+q_K} |\mathbf{h}_1(\mathbf{i})\rangle_{n+m+d} |\bar{\xi}_1\rangle_2 |\mathbf{h}_1(\mathbf{i}) \boxminus \bar{\xi}_1\rangle_{n+m+d+2} |\text{anc}\rangle_5 \\
& \quad \cdots |\mathbf{h}_K(\mathbf{i})\rangle_{n+m+d} |\bar{\xi}_K\rangle_2 |\mathbf{h}_K(\mathbf{i}) \boxminus \bar{\xi}_K\rangle_{n+m+d+2} |\text{anc}\rangle_5 |0\rangle_K.
\end{aligned} \tag{208}$$

The number of elementary gates used in this step is at most

$$K \cdot 61(n+m+d+3)^2. \tag{209}$$

4. We use the permutation circuit  $\mathcal{T}_\pi$  to reorder the qubits for addition, where

$$\begin{aligned}
& |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |\text{anc}\rangle_{q_1+\dots+q_K} |\mathbf{h}_1(\mathbf{i})\rangle_{n+m+d} |\bar{\xi}_1\rangle_2 |\mathbf{h}_1(\mathbf{i}) \boxminus \bar{\xi}_1\rangle_{n+m+d+2} |\text{anc}\rangle_5 \\
& \quad \cdots |\mathbf{h}_K(\mathbf{i})\rangle_{n+m+d} |\bar{\xi}_K\rangle_2 |\mathbf{h}_K(\mathbf{i}) \boxminus \bar{\xi}_K\rangle_{n+m+d+2} |\text{anc}\rangle_5 |0\rangle_K \\
& \xrightarrow{\mathcal{T}_\pi} |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |\text{anc}\rangle_{q_1+\dots+q_K+K(n+m+d+7)} |\mathbf{h}_1(\mathbf{i}) \boxminus \bar{\xi}_1\rangle_{n+m+d+2} |\mathbf{h}_2(\mathbf{i}) \boxminus \bar{\xi}_2\rangle_{n+m+d+2} |0\rangle \\
& \quad |\mathbf{h}_3(\mathbf{i}) \boxminus \bar{\xi}_3\rangle_{n+m+d+2} |0\rangle \cdots |\mathbf{h}_K(\mathbf{i}) \boxminus \bar{\xi}_K\rangle_{n+m+d+2} |0\rangle |0\rangle.
\end{aligned} \tag{210}$$

The number of elementary gates used in this step is at most

$$2N^2. \tag{211}$$

5. We perform the following addition inductively on the sums for  $k = 1, \dots, K-1$

$$\boxplus : \mathbb{F}_{n+d+2+k-1, m} \times \mathbb{F}_{n+d+2, m} \rightarrow \mathbb{F}_{n+d+2+k, m}, \quad \left( \bigboxplus_{l=1}^k (\mathbf{h}_l(\mathbf{i}) \boxminus \bar{\xi}_l), \mathbf{h}_{k+1}(\mathbf{i}) \boxminus \bar{\xi}_{k+1} \right) \mapsto \bigboxplus_{l=1}^{k+1} (\mathbf{h}_l(\mathbf{i}) \boxminus \bar{\xi}_l), \tag{212}$$

where we apply the quantum circuit  $\mathcal{Q}_{(+)}$  from Corollary 4.10 inductively for  $k = 1, \dots, K-1$  (with  $n_1 \leftarrow n+d+2+k-1$ ,  $m_1 \leftarrow m$ ,  $n_2 \leftarrow n+d+2$ ,  $m_2 \leftarrow m$  in the notation of Corollary 4.10) so that

$$\begin{aligned}
& |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |\text{anc}\rangle_{q_1+\dots+q_K+K(n+m+d+7)} |\mathbf{h}_1(\mathbf{i}) \boxminus \bar{\xi}_1\rangle_{n+m+d+2} |\mathbf{h}_2(\mathbf{i}) \boxminus \bar{\xi}_2\rangle_{n+m+d+2} |0\rangle \\
& \quad |\mathbf{h}_3(\mathbf{i}) \boxminus \bar{\xi}_3\rangle_{n+m+d+2} |0\rangle \cdots |\mathbf{h}_K(\mathbf{i}) \boxminus \bar{\xi}_K\rangle_{n+m+d+2} |0\rangle |0\rangle \\
& \xrightarrow{\mathcal{Q}_{(+)}^{(1)}} |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |\text{anc}\rangle_{q_1+\dots+q_K+K(n+m+d+7)} |\mathbf{h}_2(\mathbf{i}) \boxminus \bar{\xi}_2\rangle_{n+m+d+2} \left| \bigboxplus_{l=1}^2 (\mathbf{h}_l(\mathbf{i}) \boxminus \bar{\xi}_l) \right\rangle_{n+m+d+3} \\
& \quad |\mathbf{h}_3(\mathbf{i}) \boxminus \bar{\xi}_3\rangle_{n+m+d+2} |0\rangle \cdots |\mathbf{h}_K(\mathbf{i}) \boxminus \bar{\xi}_K\rangle_{n+m+d+2} |0\rangle |0\rangle \\
& \quad \vdots \\
& \xrightarrow{\mathcal{Q}_{(+)}^{(K)}} |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |\text{anc}\rangle_{q_1+\dots+q_K+K(n+m+d+7)} |\mathbf{h}_2(\mathbf{i}) \boxminus \bar{\xi}_2\rangle_{n+m+d+2} |\mathbf{h}_3(\mathbf{i}) \boxminus \bar{\xi}_3\rangle_{n+m+d+2} \\
& \quad \cdots |\mathbf{h}_K(\mathbf{i}) \boxminus \bar{\xi}_K\rangle_{n+m+d+2} \left| \bigboxplus_{l=1}^K (\mathbf{h}_l(\mathbf{i}) \boxminus \bar{\xi}_l) \right\rangle_{n+m+d+K+1} |0\rangle \\
& =: |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |\text{anc}\rangle_{q_1+\dots+q_K+K(n+m+d+7)+(K-1)(n+m+d+2)} |\mathbf{h}(\mathbf{i})\rangle_{n+m+d+K+1} |0\rangle.
\end{aligned} \tag{213}$$

The number of elementary gates used in this step is at most

$$\sum_{k=1}^{K-1} 29[(n+m+d+2+k-1) + (n+m+d+2) + 1]^2 \quad (214)$$

6. Lastly, we apply the quantum circuit  $\mathcal{R}_f$  from Lemma 4.23 (with  $a_1 \leftarrow s$ ,  $a_0 \leftarrow \pi/2$ ,  $n \leftarrow n+d+K+1$ ,  $m \leftarrow m$  in the notation of Lemma 4.23)

$$\begin{aligned} & |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |\text{anc}\rangle_{q_1+\cdots+q_k+K(n+m+d+7)+(K-1)(n+m+d+2)} |\mathbf{h}(\mathbf{i})\rangle_{n+m+d+K+1} |0\rangle \\ & \xrightarrow{\mathcal{R}_f} |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |\text{anc}\rangle_{q_1+\cdots+q_k+K(n+m+d+7)+(K-1)(n+m+d+2)} |\mathbf{h}(\mathbf{i})\rangle_{n+m+d+K+1} \\ & \quad [\cos(\bar{f}(\mathbf{h}(\mathbf{i}))/2) |0\rangle + \sin(\bar{f}(\mathbf{h}(\mathbf{i}))/2) |1\rangle] \\ & =: |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |\text{anc}\rangle_{q_1+\cdots+q_K+2K(n+m+d+5)-1} [\cos(\bar{f}(\mathbf{h}(\mathbf{i}))/2) |0\rangle + \sin(\bar{f}(\mathbf{h}(\mathbf{i}))/2) |1\rangle]. \end{aligned} \quad (215)$$

This is the desired state (203). The number of elementary gates used in this step is at most

$$13(n+d+K+1+m+1)^3. \quad (216)$$

Note that  $p \leq 4d(n+m+1)$  (c.f. (187)), hence, for each  $k = 1, \dots, K$ ,

$$\begin{aligned} q_k &= I_k(n+m+d+p) + (I_k-2)(n+m+d) + 5(I_k-1) \\ &\leq I_k(n+m+d+4d(n+m+1)) + (I_k-2)(n+m+d) + 5(I_k-1) \\ &\leq I_k(d(n+m+1) + 4d(n+m+1)) + 6I_k d(n+m+1) \\ &= 11I_k d(n+m+1). \end{aligned} \quad (217)$$

Hence,

$$\begin{aligned} N &= d(n_1+m_1) + \sum_{k=1}^K q_k + 2K(n+m+d+5) \\ &\leq d(n+m+1) + K \cdot \max_{k=1,\dots,K} \{I_k\} \cdot 11d(n+m+1) + 12Kd(n+m+1) \\ &\leq 24K \cdot \max_{k=1,\dots,K} \{I_k\} \cdot d(n+m+1). \end{aligned} \quad (218)$$

Thus, the total number of elementary gates used is at most

$$\begin{aligned} & K \cdot 2N^2 + \sum_{k=1}^K 10651I_k^3 d^3(n+m+1)^3 + 2K \\ & + K \cdot 61(n+m+d+3)^2 + 2N^2 \\ & + \sum_{k=1}^{K-1} 29[(n+m+d+2+k-1) + (n+m+d+2) + 1]^2 + 13(n+d+K+1+m+1)^3 \\ & \leq 2 \cdot 24^2 K^3 \left( \max_{k=1,\dots,K} \{I_k\} \right)^2 d^2(n+m+1)^2 + 10651K \left( \max_{k=1,\dots,K} \{I_k\} \right)^3 d^3(n+m+1)^3 + 2K \\ & + 61 \cdot 4^2 K d^2(n+m+1)^2 + 2 \cdot 24^2 K^3 \left( \max_{k=1,\dots,K} \{I_k\} \right)^2 d^2(n+m+1)^2 \\ & + 29K[3Kd(n+m+1) + 3d(n+m+1) + 1]^2 + 13(4Kd(n+m+1))^3 \\ & \leq 2 \cdot 24^2 K^3 \left( \max_{k=1,\dots,K} \{I_k\} \right)^2 d^2(n+m+1)^2 + 10651K \left( \max_{k=1,\dots,K} \{I_k\} \right)^3 d^3(n+m+1)^3 + 2K \\ & + 61 \cdot 4^2 K d^2(n+m+1)^2 + 2 \cdot 24^2 K^3 \left( \max_{k=1,\dots,K} \{I_k\} \right)^2 d^2(n+m+1)^2 \\ & + 29K[7Kd(n+m+1)]^2 + 13 \cdot 4^3 K^3 d^3(n+m+1)^3 \\ & \leq (2 \cdot 24^2 + 10651 + 2 + 61 \cdot 4^2 + 2 \cdot 24^2 + 29 \cdot 7^2 + 13 \cdot 4^3) K^3 \left( \max_{k=1,\dots,K} \{I_k\} \right)^3 d^3(n+m+1)^3 \\ & = 16186K^3 \left( \max_{k=1,\dots,K} \{I_k\} \right)^3 d^3(n+m+1)^3. \end{aligned} \quad (219)$$

□

## 5 Error analysis

In this section, we provide the detailed error analysis of the steps of Algorithm 1 outlined in Section 2.4.1. We begin the error analysis with a few basic lemmas.

**Lemma 5.1 (Lipschitz constant of CPWA functions)** *Let  $h : \mathbb{R}_+^d \rightarrow \mathbb{R}$  be a CPWA function given by (8). Let Assumption 2.4 hold with corresponding constant  $C_2 \in [1, \infty)$ . Then,  $h : \mathbb{R}_+^d \rightarrow \mathbb{R}$  is Lipschitz continuous with Lipschitz constant*

$$L := \sum_{k=1}^K \max_{1 \leq l \leq I_k} \{\|\mathbf{a}_{k,l}\|_\infty\} \sqrt{d} \leq C_2^2 d^{\frac{3}{2}}, \quad (220)$$

i.e.

$$\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}_+^d : |h(\mathbf{x}) - h(\mathbf{y})| \leq C_2^2 d^{\frac{3}{2}} \|\mathbf{x} - \mathbf{y}\|_2. \quad (221)$$

*Proof.* Following the proof of Lemma 3.3 in [53], the CPWA function  $h : \mathbb{R}_+^d \rightarrow \mathbb{R}$  admits the following representation:

$$h(\mathbf{x}) = \begin{cases} \mathbf{a}'_1 \cdot \mathbf{x} + b'_1, & \text{if } \mathbf{x} \in \Omega_1, \\ \vdots \\ \mathbf{a}'_J \cdot \mathbf{x} + b'_J, & \text{if } \mathbf{x} \in \Omega_J, \end{cases} \quad (222)$$

where  $J := \prod_{k=1}^K I_k \in \mathbb{N}$  and the coefficients  $\{\mathbf{a}'_j, b'_j : j = 1, \dots, J\}$  are of the form

$$\mathbf{a}'_j := \sum_{k=1}^K \xi_k \mathbf{a}_{k,l_k^*}, \quad b'_j := \sum_{k=1}^K \xi_k b_{k,l_k^*}, \quad (223)$$

for some  $l_k^* \in \{1, 2, \dots, I_k\}$  (the specific choice of index  $l_k^*$  can be found in the proof of Lemma 3.3 in [53]), and where  $\Omega_1, \dots, \Omega_J$  are polyhedrons whose union is  $\mathbb{R}_+^d$ . Note that some of these sets  $\Omega_j$  can be empty. We claim that

$$\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}_+^d : |h(\mathbf{x}) - h(\mathbf{y})| \leq \max_{1 \leq j \leq J} \{\|\mathbf{a}'_j\|_\infty\} \cdot \sqrt{d} \|\mathbf{x} - \mathbf{y}\|_2. \quad (224)$$

Let  $\mathbf{x}, \mathbf{y} \in \mathbb{R}_+^d$  be fixed. Consider the line segment from  $\mathbf{x}$  to  $\mathbf{y}$  given by the set  $\Gamma := \{\gamma(t) := \mathbf{x} + t(\mathbf{y} - \mathbf{x}) : t \in [0, 1]\}$ . If the line segment  $\Gamma$  lies entirely in one of polyhedron  $\Omega_{j^*}$ , then by linearity of  $h$  in  $\Omega_{j^*}$  and the Hölder's inequality, it follows that

$$|h(\mathbf{x}) - h(\mathbf{y})| = |\mathbf{a}'_{j^*} \cdot (\mathbf{x} - \mathbf{y})| \leq \|\mathbf{a}'_{j^*}\|_\infty \|\mathbf{x} - \mathbf{y}\|_1 \leq \max_{1 \leq j \leq J} \{\|\mathbf{a}'_j\|_\infty\} \cdot \sqrt{d} \|\mathbf{x} - \mathbf{y}\|_2.$$

In the general case, the line segment  $\Gamma$  may be contained in some  $n \geq 2$  polyhedrons  $\Omega_{j_1}, \dots, \Omega_{j_n}$ , such that there are  $n + 1$  points  $\{\gamma(t_0), \gamma(t_1), \dots, \gamma(t_n)\} \subset \Gamma$  with a partition  $\{0 =: t_0 < t_1 < \dots < t_n := 1\}$ , satisfying  $\gamma(t_m) \in \Omega_{j_m} \cap \Omega_{j_{m+1}}$  for  $m = 1, \dots, n - 1$ . Again by the same argument, it holds for all  $m = 1, \dots, n - 1$  that

$$|h(\gamma(t_m)) - h(\gamma(t_{m-1}))| = |\mathbf{a}'_{j_m} \cdot (\gamma(t_m) - \gamma(t_{m-1}))| \leq \|\mathbf{a}'_{j_m}\|_\infty (t_m - t_{m-1}) \sqrt{d} \|\mathbf{x} - \mathbf{y}\|_2.$$

Hence, summing over  $m$ , we have

$$\begin{aligned} |h(\mathbf{x}) - h(\mathbf{y})| &\leq \sum_{m=1}^n \|\mathbf{a}'_{j_m}\|_\infty (t_m - t_{m-1}) \sqrt{d} \|\mathbf{x} - \mathbf{y}\|_2 \\ &\leq \max_{1 \leq j \leq J} \{\|\mathbf{a}'_j\|_\infty\} \cdot \sqrt{d} \|\mathbf{x} - \mathbf{y}\|_2. \end{aligned}$$

Thus, the claim (224) is proven. Next, by (223), it follows that

$$\max_{1 \leq j \leq J} \{\|\mathbf{a}'_j\|_\infty\} \leq \sum_{k=1}^K \max_{1 \leq l \leq I_k} \{\|\mathbf{a}_{k,l}\|_\infty\}. \quad (225)$$

This, (224) and Assumption 2.4 concludes the lemma.  $\square$

**Lemma 5.2 (Linear growth of CPWA)** *Let  $h : \mathbb{R}_+^d \rightarrow \mathbb{R}$  be a CPWA function given by (8). Let Assumption 2.4 hold with corresponding constant  $C_2 \in [1, \infty)$ . Then, for all  $\mathbf{x} \in \mathbb{R}_+^d$ , it holds that*

$$|h(\mathbf{x})| \leq C_2^2 d^{\frac{3}{2}} (1 + \|\mathbf{x}\|_2). \quad (226)$$

*Proof.* Note that Lemma 5.1 and an application of the triangle equality shows that

$$|h(\mathbf{x})| \leq |h(\mathbf{x}) - h(0)| + |h(0)| \leq C_2^2 d^{\frac{3}{2}} \|\mathbf{x}\|_2 + |h(0)|. \quad (227)$$

Moreover, by (8) and Assumption 2.4, we have for all  $k = 1, \dots, K$  that

$$|h(0)| \leq K \cdot \max\{|b_{k,l}| : l = 1, \dots, I_k\} \leq C_2^2 d. \quad (228)$$

This and (227) implies (226).  $\square$

## 5.1 Step 1: Truncation error bounds

**Lemma 5.3** *Let  $Y$  be a log-normal random variable with parameters  $\mu \in \mathbb{R}$  and  $\sigma^2 \in (0, \infty)$ . Let  $Z$  be a standard normal random variable. Then it holds that*

(i) *for all  $k \in \mathbb{N}$ ,*

$$\mathbb{E}[Y^k] = e^{k\mu + \frac{k^2\sigma^2}{2}}, \quad (229)$$

(ii) *for all  $y \in [0, \infty)$  that*

$$\mathbb{P}(Z \geq y) \leq \frac{1}{2}e^{-\frac{y^2}{2}}. \quad (230)$$

*Proof.* Item (i) is proven in [3, Chapter 2.3], and Item (ii) is proved in [17, Eq. 6].  $\square$

**Proposition 5.4 (Truncation error)** *Let  $\varepsilon \in (0, 1)$ ,  $d \in \mathbb{N}$ ,  $r, T \in (0, \infty)$ , and  $(t, \mathbf{x}) \in [0, T] \times \mathbb{R}_+^d$ . Let  $h : \mathbb{R}_+^d \rightarrow \mathbb{R}$  be a CPWA function given by (8). Let Assumption 2.2 and Assumption 2.4 hold with respective constants  $C_1, C_2 \in [1, \infty)$ . Let  $p(\cdot, T; \mathbf{x}, t) : \mathbb{R}_+^d \rightarrow \mathbb{R}_+$  be the log-normal transition density function given by (4). Let  $M_{d,\varepsilon} \in [1, \infty)$  satisfy*

$$M_{d,\varepsilon} = 2C_2^2 d^{\frac{5}{2}} \varepsilon^{-1} e^{4C_1^2 T^2} e^{2rT} \max_{i=1,\dots,d} \{1, x_i^2\}. \quad (231)$$

*Let  $u : [0, T] \times \mathbb{R}_+^d \rightarrow \mathbb{R}$  be the option price given by*

$$u(t, \mathbf{x}) := e^{-r(T-t)} \int_{\mathbb{R}_+^d} h(\mathbf{y}) p(\mathbf{y}, T; \mathbf{x}, t) d\mathbf{y}, \quad (232)$$

*and for every  $M \geq M_{d,\varepsilon}$ , let  $\bar{u}_{M,t,\mathbf{x}} \in \mathbb{R}$  be the truncated solution given by*

$$\bar{u}_{M,t,\mathbf{x}} := e^{-r(T-t)} \int_{[0,M]^d} h(\mathbf{y}) p(\mathbf{y}, T; \mathbf{x}, t) d\mathbf{y}. \quad (233)$$

*Then, the truncation solution satisfies the following estimate*

$$|u(t, \mathbf{x}) - \bar{u}_{M,t,\mathbf{x}}| \leq \varepsilon. \quad (234)$$

*Proof.* Let  $\mathbf{\Sigma} \equiv \mathbf{\Sigma}_d := (T-t)\mathbf{C}_d \in \mathbb{R}^{d \times d}$ , and  $\boldsymbol{\mu} \equiv \boldsymbol{\mu}_d \in \mathbb{R}^d$  denote the log-covariance and log-mean parameters for the multivariate log-normal random variable  $\mathbf{Y}$  with the probability density function  $p(\cdot, T; \mathbf{x}, t)$  given by Lemma 2.1. Using Lemma 5.2, (232), (233), the fact that  $e^{-r(T-t)} \leq 1$ , and Cauchy-Schwarz inequality,

$$\begin{aligned} |u(t, \mathbf{x}) - \bar{u}_{M,t,\mathbf{x}}| &= e^{-r(T-t)} \left| \int_{\mathbb{R}_+^d} h(\mathbf{y}) p(\mathbf{y}, T; \mathbf{x}, t) d\mathbf{y} - \int_{[0,M]^d} h(\mathbf{y}) p(\mathbf{y}, T; \mathbf{x}, t) d\mathbf{y} \right| \\ &\leq C_2^2 d^{\frac{3}{2}} \mathbb{E}[(1 + \|\mathbf{Y}\|_2)^2]^{1/2} \mathbb{P}(\mathbf{Y} \notin [0, M]^d)^{1/2}, \end{aligned} \quad (235)$$

where

$$\mathbb{P}(\mathbf{Y} \notin [0, M]^d) = \int_{\mathbb{R}_+^d \setminus [0,M]^d} p(\mathbf{y}, T; \mathbf{x}, t) d\mathbf{y}.$$

Let  $\mathbf{X} = (X_1, \dots, X_d) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$  be the multivariate Gaussian random variable and recall that  $X_i = \ln(Y_i)$  for  $i = 1, \dots, d$ . Using Lemma 5.3 (i), we have

$$\mathbb{E}[\|\mathbf{Y}\|_2^2] = \sum_{i=1}^d \mathbb{E}[Y_i^2] = \sum_{i=1}^d \mathbb{E}[e^{2X_i}] = \sum_{i=1}^d e^{2\mu_i + 2\sigma_{ii}^2}, \quad (236)$$

where  $\mu_i = \ln(x_i) + (r - \frac{1}{2}\sigma_{ii}^2)(T-t)$  and  $\sigma_{ii} = \mathbf{\Sigma}_{i,i}$ . We use the bound  $e^{2\mu_i} \leq \max_{i=1,\dots,d} \{1, x_i^2\} e^{2rT}$  and  $e^{2\sigma_{ii}^2} \leq e^{2C_1^2 T^2}$  by Assumption 2.2 and conclude that

$$\mathbb{E}[\|\mathbf{Y}\|_2^2] \leq d e^{2C_1^2 T^2} e^{2rT} \max_{i=1,\dots,d} \{1, x_i^2\}. \quad (237)$$

By Cauchy-Schwarz inequality, we have  $(1 + \|\mathbf{Y}\|_2)^2 \leq 2(1 + \|\mathbf{Y}\|_2^2)$ . Also, note that  $\max_i \{1, x_i^2\}^{1/2} = \max_i \{1, |x_i|\}$  and  $1 \leq d e^{2C_1^2 T^2} e^{2rT} \max_i \{1, x_i^2\}$ . Combining the above bounds, we arrive at

$$\mathbb{E}[(1 + \|\mathbf{Y}\|_2)^2]^{1/2} \leq 2d^{1/2} e^{C_1^2 T^2} e^{rT} \max_{i=1,\dots,d} \{1, |x_i|\}. \quad (238)$$

Hence, we have

$$|u(t, \mathbf{x}) - \bar{u}_{M,t,\mathbf{x}}| \leq 2C_2^2 d^2 e^{C_1^2 T^2} e^{rT} \max_{i=1,\dots,d} \{1, |x_i|\} \cdot \mathbb{P}(\mathbf{Y} \notin [0, M]^d)^{1/2}. \quad (239)$$

Moreover, for all  $i = 1, \dots, d$ , we have the inclusions

$$\{Y_i \leq M\} = \{X_i \leq \ln(M)\} \supseteq \{|X_i - \mu_i| \leq \ln(M) - \mu_i\}. \quad (240)$$

Note that since  $M \geq M_{d,\varepsilon} \geq e^{\mu_i}$ , we have that  $\ln(M) \geq \mu_i$ . By Sidak's correlation inequality [68, Corollary 1], we have

$$\mathbb{P}\left(\bigcap_{i=1}^d \{|X_i - \mu_i| \leq \ln(M) - \mu_i\}\right) \geq \prod_{i=1}^d \mathbb{P}(\{|X_i - \mu_i| \leq \ln(M) - \mu_i\}), \quad (241)$$

and hence,

$$\mathbb{P}(\mathbf{Y} \notin [0, M]^d) = 1 - \mathbb{P}\left(\bigcap_{i=1}^d \{Y_i \leq M\}\right) \leq 1 - \prod_{i=1}^d \mathbb{P}(\{|X_i - \mu_i| \leq \ln(M) - \mu_i\}). \quad (242)$$

Denote by  $Z$  the standard normal random variable. Using Lemma 5.3 (ii) and Assumption 2.2, we have

$$\mathbb{P}(\{|X_i - \mu_i| \leq \ln(M) - \mu_i\}) = 1 - 2\mathbb{P}(Z \geq \frac{\ln(M) - \mu_i}{\sigma_{ii}}) \geq 1 - e^{-\frac{(\ln(M) - \mu_i)^2}{2\sigma_{ii}^2}} \geq 1 - e^{-\frac{(\ln(M) - \mu_i)^2}{2C_1^2 T^2}}. \quad (243)$$

Moreover, using  $\ln(M) > 0$  and  $M \geq M_{d,\varepsilon} \geq \max_{i=1,\dots,d} \{1, x_i^2\} e^{2rT} e^{4C_1^2 T^2} \geq e^{2\mu_i} e^{4C_1^2 T^2}$ , we have

$$\begin{aligned} \ln(M) &\geq 2\mu_i + 4C_1^2 T^2 \\ \iff (\ln(M))^2 &\geq (2\mu_i + 4C_1^2 T^2) \ln(M) \\ \iff (\ln(M))^2 - 2\mu_i \ln(M) + \mu_i^2 &\geq 4C_1^2 T^2 \ln(M) + \mu_i^2 \\ \implies (\ln(M) - \mu_i)^2 &\geq 4C_1^2 T^2 \ln(M) \\ \iff -\frac{(\ln(M) - \mu_i)^2}{2C_1^2 T^2} &\leq -2\ln(M) \\ \iff e^{-\frac{(\ln(M) - \mu_i)^2}{2C_1^2 T^2}} &\leq M^{-2}. \end{aligned} \quad (244)$$

Using Bernoulli's inequality, that is  $(1+z)^d \geq 1+dz$  for any  $z \in [-1, \infty)$ , and the fact that  $-M^{-2} \in [-1, 0)$ , we have

$$\mathbb{P}(\mathbf{Y} \notin [0, M]^d) \leq 1 - (1 - M^{-2})^d \leq dM^{-2}. \quad (245)$$

Thus, by (239), (245), and (231), we conclude that

$$|u(t, \mathbf{x}) - \bar{u}_{M,t,\mathbf{x}}| \leq 2C_2^2 d^{\frac{5}{2}} e^{C_1^2 T^2} e^{rT} \max_{i=1,\dots,d} \{1, |x_i|\} M^{-1} \leq 2C_2^2 d^{\frac{5}{2}} e^{C_1^2 T^2} e^{rT} \max_{i=1,\dots,d} \{1, |x_i|\} M_{d,\varepsilon}^{-1} \leq \varepsilon. \quad (246)$$

□

## 5.2 Step 2: Quadrature error bounds

**Proposition 5.5 (Quadrature error)** *Let  $d \in \mathbb{N}$ ,  $r, T \in (0, \infty)$ ,  $n \in \mathbb{N} \cap \{2, 3, \dots\}$ ,  $m \in \mathbb{N}$ , and  $(t, \mathbf{x}) \in [0, T] \times \mathbb{R}_+^d$ . Let  $M \in [1, \infty)$  be defined by  $M := 2^{n-1}$ . Let  $h : \mathbb{R}_+^d \rightarrow \mathbb{R}$  be the CPWA function given by (8). Let Assumption 2.4 hold with corresponding constant  $C_2 \in [1, \infty)$ . Let  $p(\cdot, T; \mathbf{x}, t) : \mathbb{R}_+^d \rightarrow \mathbb{R}_+$  be the log-normal transition density given by (4). Let  $\tilde{u}_{n,t,\mathbf{x}} \in \mathbb{R}$  be the truncated solution given by*

$$\tilde{u}_{n,t,\mathbf{x}} := e^{-r(T-t)} \int_{[0,M]^d} h(\mathbf{y}) p(\mathbf{y}, T; \mathbf{x}, t) d\mathbf{y}, \quad (247)$$

and let  $\tilde{u}_{n,m,t,\mathbf{x}} \in \mathbb{R}$  be the truncated quadrature solution given by

$$\tilde{u}_{n,m,t,\mathbf{x}} := e^{-r(T-t)} \sum_{\mathbf{j} \in \mathbb{K}_{n,m,+}^d} h(\mathbf{j}) p_{\mathbf{j},m}, \quad (248)$$

where for  $\mathbf{j} = (j_1, \dots, j_d) \in \mathbb{K}_{n,m,+}^d$  (c.f Definition 4.1),

$$p_{\mathbf{j},m} := \int_{Q_{\mathbf{j},m}} p(\mathbf{y}, T; \mathbf{x}, t) d\mathbf{y}, \quad \text{and} \quad Q_{\mathbf{j},m} := [j_1, j_1 + 2^{-m}) \times \dots \times [j_d, j_d + 2^{-m}). \quad (249)$$

Then,

$$|\tilde{u}_{n,t,\mathbf{x}} - \tilde{u}_{n,m,t,\mathbf{x}}| \leq C_2^2 d^2 2^{-m}. \quad (250)$$

*Proof.* Let  $[\cdot]_m : \mathbb{R}_+^d \rightarrow (2^{-m}\mathbb{Z})^d$  be a function defined by

$$[\mathbf{y}]_m = \left( \frac{\lfloor 2^m y_1 \rfloor}{2^m}, \dots, \frac{\lfloor 2^m y_d \rfloor}{2^m} \right), \quad (251)$$

where  $\lfloor \cdot \rfloor$  is the floor function. With this function, it holds for every  $\mathbf{j} \in \mathbb{K}_{n,m,+}^d$  that

$$\forall \mathbf{y} \in Q_{\mathbf{j},m}, \quad [\mathbf{y}]_m = \mathbf{j}. \quad (252)$$

Moreover, since

$$[0, M]^d = \bigsqcup_{\mathbf{j} \in \mathbb{K}_{n,m,+}^d} Q_{\mathbf{j},m} \quad (253)$$

is a disjoint union of sets, it follows that

$$\tilde{u}_{n,m,t,\mathbf{x}} = e^{-r(T-t)} \sum_{\mathbf{j} \in \mathbb{K}_{n,m,+}^d} \int_{Q_{\mathbf{j},m}} h([\mathbf{y}]_m) p(\mathbf{y}, T; \mathbf{x}, t) d\mathbf{y} = e^{-r(T-t)} \int_{[0,M]^d} h([\mathbf{y}]_m) p(\mathbf{y}, T; \mathbf{x}, t) d\mathbf{y}. \quad (254)$$

Furthermore, observe that

$$\forall \mathbf{y} \in \mathbb{R}_+^d, \quad \|\mathbf{y} - [\mathbf{y}]_m\|_1 \leq d2^{-m}. \quad (255)$$

Hence, by definition of  $\tilde{u}_{n,t,\mathbf{x}}$  in (247), (254), by the fact that  $e^{-r(T-t)} \leq 1$ , by the Lipschitz continuity of  $h(\cdot)$  in Lemma 5.2, by Assumption 2.4, (255), and by the fact that  $p(\mathbf{y}, T; \mathbf{x}, t)$  is a probability density function supported on  $\mathbb{R}_+^d$ , we conclude that

$$\begin{aligned} |\tilde{u}_{n,t,\mathbf{x}} - \tilde{u}_{n,m,t,\mathbf{x}}| &\leq e^{-r(T-t)} \int_{[0,M]^d} |h(\mathbf{y}) - h([\mathbf{y}]_m)| p(\mathbf{y}, T; \mathbf{x}, t) d\mathbf{y} \\ &\leq \int_{[0,M]^d} C_2^2 d \|\mathbf{y} - [\mathbf{y}]_m\|_1 p(\mathbf{y}, T; \mathbf{x}, t) d\mathbf{y} \\ &\leq C_2^2 d^2 2^{-m}. \end{aligned} \quad (256)$$

□

**Corollary 5.6 (Truncation and quadrature errors)** *Let  $\varepsilon \in (0, 1)$ ,  $d \in \mathbb{N}$ ,  $r, T \in (0, \infty)$ , and  $(t, \mathbf{x}) \in [0, T] \times \mathbb{R}_+^d$ . Let  $u(t, \mathbf{x})$  be the option price given by (7). Let  $h : \mathbb{R}_+^d \rightarrow \mathbb{R}$  be the CPWA function given by (8). Let Assumption 2.2 and Assumption 2.4 hold with respective constants  $C_1, C_2 \in [1, \infty)$ . For every  $\eta \in (0, 1)$ , let  $M_{d,\eta} \in [1, \infty)$  be given by (231). For every  $n, m \in \mathbb{N}$  satisfying*

$$n \geq 1 + \log_2(M_{d,\varepsilon/2}), \quad (257)$$

$$m \geq \log_2(C_2^2 d^2 (\varepsilon/2)^{-1}) \quad (258)$$

*let  $\tilde{u}_{n,m,t,\mathbf{x}}$  be the truncated quadrature solution given as in (248). Then*

$$|u(t, \mathbf{x}) - \tilde{u}_{n,m,t,\mathbf{x}}| \leq \varepsilon. \quad (259)$$

*Proof.* By (257), it holds that

$$M := 2^{n-1} \geq 2^{n_{d,\varepsilon}-1} \geq M_{d,\varepsilon/2}.$$

Let  $\tilde{u}_{n,t,\mathbf{x}}$  be the truncated solution given by (247). By Proposition 5.4 (with  $\varepsilon \leftarrow \varepsilon/2$  and  $M \leftarrow 2^{n-1}$  in the notation of Proposition 5.4), it follows that

$$|u(t, \mathbf{x}) - \tilde{u}_{n,t,\mathbf{x}}| \leq \varepsilon/2. \quad (260)$$

Moreover, by (258), it holds that  $C_2^2 d^2 2^{-m} \leq \varepsilon/2$ . Hence, by Proposition 5.5, it follows that

$$|\tilde{u}_{n,t,\mathbf{x}} - \tilde{u}_{n,m,t,\mathbf{x}}| \leq \varepsilon/2. \quad (261)$$

Thus, (259) follows from triangle inequality. □

### 5.3 Step 3: Approximation error bounds for payoff function

**Lemma 5.7 (Sublinear property of the max-function)** Let  $\varepsilon > 0$ ,  $n \in \mathbb{N}$ , and let  $(a_i)_{i=1}^n, (\tilde{a}_i)_{i=1}^n \subset \mathbb{R}$  satisfy for all  $i = 1, \dots, n$ :

$$|a_i - \tilde{a}_i| \leq \varepsilon. \quad (262)$$

Then,

$$\left| \max_{i=1, \dots, n} \{a_i\} - \max_{i=1, \dots, n} \{\tilde{a}_i\} \right| \leq \varepsilon. \quad (263)$$

*Proof.* Let  $i \in \{1, \dots, n\}$  be arbitrary. The fact that  $\tilde{a}_i \leq \max_{j=1, \dots, n} \{\tilde{a}_j\}$  and (262) imply that

$$a_i - \max_{j=1, \dots, n} \{\tilde{a}_j\} \leq a_i - \tilde{a}_i \leq |a_i - \tilde{a}_i| \leq \varepsilon. \quad (264)$$

Since  $i$  was arbitrarily chosen, taking maximum over  $i = 1, \dots, n$  yields

$$\max_{i=1, \dots, n} \{a_i\} - \max_{j=1, \dots, n} \{\tilde{a}_j\} \leq \varepsilon. \quad (265)$$

Repeating the argument by symmetry concludes (263).  $\square$

**Proposition 5.8 (Approximating payoff function)** Let  $d \in \mathbb{N}$ ,  $r, T \in (0, \infty)$ ,  $(t, \mathbf{x}) \in [0, T] \times \mathbb{R}_+^d$ ,  $n_1 \in \mathbb{N} \cap \{2, 3, \dots\}$ , and  $m_1 \in \mathbb{N}_0$ . Let  $h : \mathbb{R}_+^d \rightarrow \mathbb{R}$  be the CPWA function given by (8). i.e.

$$h(\mathbf{x}) = \sum_{k=1}^K \xi_k \max\{\mathbf{a}_{k,l} \cdot \mathbf{x} + b_{k,l} : l = 1, \dots, I_k\}.$$

Let Assumption 2.4 hold with corresponding constant  $C_2 \in [1, \infty)$ , and let  $n_2 \in \mathbb{N} \cap \{2, 3, \dots\}$  be defined by

$$n_2 := 1 + \lceil \log_2(C_2) \rceil. \quad (266)$$

For every  $m_2 \in \mathbb{N}_0$  define the function  $[\cdot]_{m_2} : \mathbb{R} \rightarrow 2^{-m_2}\mathbb{Z}$  by  $[x]_{m_2} := \frac{\lfloor 2^{m_2}x \rfloor}{2^{m_2}}$ ,  $x \in \mathbb{R}$ . For every  $m_2 \in \mathbb{N}_0$ ,  $k = 1, \dots, K$ , and  $l = 1, \dots, I_k$ , define  $\tilde{\mathbf{a}}_{n_2, m_2, k, l} = (\tilde{a}_{n_2, m_2, k, l, 1}, \dots, \tilde{a}_{n_2, m_2, k, l, d}) \in \mathbb{R}^d$  and  $\tilde{b}_{n_2, m_2, k, l} \in \mathbb{R}$  by

$$\begin{aligned} \tilde{a}_{n_2, m_2, k, l, i} &:= [(\mathbf{a}_{k,l})_i]_{m_2}, \quad i = 1, \dots, d, \\ \tilde{b}_{n_2, m_2, k, l} &:= [b_{k,l}]_{m_2}, \end{aligned} \quad (267)$$

and define the function  $\tilde{h}_{n_2, m_2} : \mathbb{R}^d \rightarrow \mathbb{R}$  by

$$\tilde{h}_{n_2, m_2}(\mathbf{x}) := \sum_{k=1}^K \xi_k \max\{\tilde{\mathbf{a}}_{n_2, m_2, k, l} \cdot \mathbf{x} + \tilde{b}_{n_2, m_2, k, l} : l = 1, \dots, I_k\}. \quad (268)$$

Let  $\tilde{u}_{n_1, m_1, t, \mathbf{x}} \in \mathbb{R}$  be the truncated quadrature solution given as in (248). i.e.

$$\tilde{u}_{n_1, m_1, t, \mathbf{x}} := e^{-r(T-t)} \sum_{\mathbf{j} \in \mathbb{K}_{n_1, m_1, +}^d} h(\mathbf{j}) p_{\mathbf{j}, m_1}.$$

For every  $m_2 \in \mathbb{N}_0$  let  $\tilde{u}_{n_1, m_1, n_2, m_2, t, \mathbf{x}} \in \mathbb{R}$  be the truncated quadrature solution with an approximated payoff function given by

$$\tilde{u}_{n_1, m_1, n_2, m_2, t, \mathbf{x}} := e^{-r(T-t)} \sum_{\mathbf{j} \in \mathbb{K}_{n_1, m_1, +}^d} \tilde{h}_{n_2, m_2}(\mathbf{j}) p_{\mathbf{j}, m_1} \quad (269)$$

Then, for every  $m_2 \in \mathbb{N}_0$  it holds that  $(\tilde{\mathbf{a}}_{n_2, m_2, k, l}, \tilde{b}_{n_2, m_2, k, l}) \in \mathbb{K}_{n_2, m_2}^{d+1}$  and that

$$|\tilde{u}_{n_1, m_1, t, \mathbf{x}} - \tilde{u}_{n_1, m_1, n_2, m_2, t, \mathbf{x}}| \leq C_2 d^2 2^{n_1 - m_2}. \quad (270)$$

*Proof.* By Assumption 2.4, it holds that

$$\forall k = 1, \dots, K, \forall l = 1, \dots, I_k, \forall i = 1, \dots, d : |(\mathbf{a}_{k,l})_i|, |b_{k,l}| \leq C_2. \quad (271)$$

Hence, by Definition 4.1, (267), and (268), it follows that  $(\tilde{\mathbf{a}}_{n_2, m_2, k, l}, \tilde{b}_{n_2, m_2, k, l}) \in \mathbb{K}_{n_2, m_2}^{d+1}$ . Furthermore, observe that

$$\forall c > 0 : |c - [c]_{m_2}| \leq 2^{-m_2}. \quad (272)$$

Hence, for all  $k = 1, \dots, K$ ,  $l = 1, \dots, I_k$ , and  $\mathbf{x} \in \mathbb{R}_+^d$ , it follows that

$$|\mathbf{a}_{k,l} \cdot \mathbf{x} + b_{k,l} - (\tilde{\mathbf{a}}_{n_2, m_2, k, l} \cdot \mathbf{x} + \tilde{b}_{n_2, m_2, k, l})| \leq (d+1)2^{-m_2} \max\{1, \max\{(\mathbf{x})_i : i = 1, \dots, d\}\}. \quad (273)$$



Moreover, since  $\mathbb{K}_{n_1, m_1} \subset [-2^{n_1-1}, 2^{n_1-1}]$  (c.f. Definition 4.1), it follows for all  $k = 1, \dots, K$ ,  $l = 1, \dots, I_k$  that

$$\forall \mathbf{j} \in \mathbb{K}_{n_1, m_1, +}^d : |\mathbf{a}_{k,l} \cdot \mathbf{j} + b_{k,l} - (\tilde{\mathbf{a}}_{n_2, m_2, k, l} \cdot \mathbf{j} + \tilde{b}_{n_2, m_2, k, l})| \leq (d+1)2^{n_1-1}2^{-m_2}. \quad (274)$$

Hence, by Lemma 5.7, it holds for all  $k = 1, \dots, K$  that

$$|\max\{\mathbf{a}_{k,l} \cdot \mathbf{j} + b_{k,l} : l = 1, \dots, I_k\} - \max\{\tilde{\mathbf{a}}_{n_2, m_2, k, l} \cdot \mathbf{j} + \tilde{b}_{n_2, m_2, k, l} : l = 1, \dots, I_k\}| \leq (d+1)2^{n_1-1}2^{-m_2}. \quad (275)$$

Hence, by Assumption 2.4, it follows that

$$\forall \mathbf{j} \in \mathbb{K}_{n_1, m_1, +}^d : |h(\mathbf{j}) - \tilde{h}_{n_2, m_2}(\mathbf{j})| \leq K(d+1)2^{n_1-1}2^{-m_2} \leq C_2 d^2 2^{n_1-m_2}. \quad (276)$$

Furthermore, since  $e^{-r(T-t)} \leq 1$  and

$$0 < \sum_{\mathbf{j} \in \mathbb{K}_{n_1, m_1, +}^d} p_{\mathbf{j}, m_1} = \int_{[0, 2^{n_1-1}]^d} p(\mathbf{y}, T; \mathbf{x}, t) d\mathbf{y} \leq 1,$$

we conclude that

$$|\tilde{u}_{n_1, m_1, t, \mathbf{x}} - \tilde{u}_{n_1, m_1, n_2, m_2, t, \mathbf{x}}| \leq C_2 d^2 2^{n_1-m_2+1} e^{-r(T-t)} \cdot \sum_{\mathbf{j} \in \mathbb{K}_{n_1, m_1, +}^d} p_{\mathbf{j}, m_1} \leq C_2 d^2 2^{n_1-m_2}. \quad (277)$$

□

**Corollary 5.9 (Truncation and quadrature with approximated payoff errors)** *Let  $\varepsilon \in (0, 1)$ ,  $d \in \mathbb{N}$ ,  $r, T \in (0, \infty)$ , and  $(t, \mathbf{x}) \in [0, T] \times \mathbb{R}_+^d$ . Let  $u(t, \mathbf{x})$  be the option price given by (7). Let  $h : \mathbb{R}_+^d \rightarrow \mathbb{R}$  be the CPWA function given by (8). Let Assumption 2.2 and Assumption 2.4 hold with respective constants  $C_1, C_2 \in [1, \infty)$ , and let  $n_2 := 1 + \lceil \log_2(C_2) \rceil$ . For every  $\eta \in (0, 1)$ , let  $M_{d, \eta} \in [1, \infty)$  be given by (231). For every  $n_1, m_1, m_2 \in \mathbb{N}$  satisfying*

$$n_1 \geq 1 + \log_2(M_{d, \varepsilon/3}), \quad (278)$$

$$m_1 \geq \log_2(C_2^2 d^2 (\varepsilon/3)^{-1}), \quad (279)$$

$$m_2 \geq 1 + \log_2(M_{d, \varepsilon/3}) + \log_2(C_2 d^2 (\varepsilon/3)^{-1}), \quad (280)$$

*let  $\tilde{u}_{n_1, m_1, n_2, m_2, t, \mathbf{x}} \in \mathbb{R}$  be the truncated quadrature solution with an approximated payoff function be given by (269). Then we have that*

$$|u(t, \mathbf{x}) - \tilde{u}_{n_1, m_1, n_2, m_2, t, \mathbf{x}}| \leq \varepsilon. \quad (281)$$

*Proof.* Let  $\tilde{u}_{n_1, m_1, t, \mathbf{x}}$  be the quadrature solution given by (248). By (278), (279), and Corollary 5.6 (with  $\varepsilon \leftarrow 2\varepsilon/3$  in the notation of Corollary 5.6), it follows that

$$|u(t, \mathbf{x}) - \tilde{u}_{n_1, m_1, t, \mathbf{x}}| \leq 2\varepsilon/3. \quad (282)$$

Moreover, by Proposition 5.8 (with  $m_2 \leftarrow m_2$  in the notation of Proposition 5.8) and (280), it follows that

$$|\tilde{u}_{n_1, m_1, t, \mathbf{x}} - \tilde{u}_{n_1, m_1, n_2, m_2, t, \mathbf{x}}| \leq C_2 d^2 2^{n_1-m_2} \leq \varepsilon/3. \quad (283)$$

Thus, the conclusion follows from the triangle inequality.

□

## 5.4 Step 4: Distribution loading error bounds

**Proposition 5.10 (Distribution loading errors)** *Let  $\varepsilon \in (0, 1)$ ,  $d \in \mathbb{N}$ ,  $r, T \in (0, \infty)$ , and  $(t, \mathbf{x}) \in [0, T] \times \mathbb{R}_+^d$ . Let  $u(t, \mathbf{x})$  be the option price given by (7). Let  $h : \mathbb{R}_+^d \rightarrow \mathbb{R}$  be the CPWA function given by (8). Let Assumption 2.2 and Assumption 2.4 hold with respective constants  $C_1, C_2 \in [1, \infty)$ , and let  $n_2 := 1 + \lceil \log_2(C_2) \rceil$ . For every  $\eta \in (0, 1)$ , let  $M_{d, \eta} \in [1, \infty)$  be given by (231) and for every  $n_1, m_1 \in \mathbb{N}$  let  $\{\tilde{p}_{\mathbf{j}, \eta} : \mathbf{j} \in \mathbb{K}_{n_1, m_1, +}^d\} \subset [0, 1]$  satisfy*

$$\sum_{\mathbf{j} \in \mathbb{K}_{n_1, m_1, +}^d} \tilde{p}_{\mathbf{j}, \eta} = 1 \quad (284)$$

and

$$\sum_{\mathbf{j} \in \mathbb{K}_{n_1, m_1, +}^d} |\tilde{p}_{\mathbf{j}, \eta} - \gamma^{-1} p_{\mathbf{j}, m_1}| \leq \frac{\eta}{C_2^2 d^2 2^{n_1+1}}, \quad (285)$$

where for all  $\mathbf{j} = (j_1, \dots, j_d) \in \mathbb{K}_{n_1, m_1, +}^d$ ,

$$p_{\mathbf{j}, m_1} := \int_{Q_{\mathbf{j}, m_1}} p(\mathbf{y}, T; \mathbf{x}, t) d\mathbf{y}, \quad Q_{\mathbf{j}, m_1} := [j_1, j_1 + 2^{-m_1}) \times \dots \times [j_d, j_d + 2^{-m_1}), \quad (286)$$

and

$$\gamma := \sum_{\mathbf{j} \in \mathbb{K}_{n_1, m_1, +}^d} p_{\mathbf{j}, m_1} \in (0, 1). \quad (287)$$

Moreover, for every  $n_1, m_1, m_2 \in \mathbb{N}$  satisfying

$$n_1 \geq 1 + \log_2(M_{d, \varepsilon/4}), \quad (288)$$

$$m_1 \geq \log_2(C_2^2 d^2 (\varepsilon/4)^{-1}), \quad (289)$$

$$m_2 \geq 1 + \log_2(M_{d, \varepsilon/4}) + \log_2(C_2 d^2 (\varepsilon/4)^{-1}), \quad (290)$$

let  $\tilde{h}_{n_2, m_2} : \mathbb{R}^d \rightarrow \mathbb{R}$  be given by (268), and let  $\tilde{u}_{n_1, m_1, n_2, m_2, p, t, \mathbf{x}} \in \mathbb{R}$  be the truncated quadrature solution with approximated payoff and loaded distribution given by

$$\tilde{u}_{n_1, m_1, n_2, m_2, p, t, \mathbf{x}} := \gamma e^{-r(T-t)} \sum_{\mathbf{j} \in \mathbb{K}_{n_1, m_1, +}^d} \tilde{p}_{\mathbf{j}, \varepsilon/4} \tilde{h}_{n_2, m_2}(\mathbf{j}), \quad (291)$$

Then,

$$|u(t, \mathbf{x}) - \tilde{u}_{n_1, m_1, n_2, m_2, p, t, \mathbf{x}}| \leq \varepsilon. \quad (292)$$

*Proof.* Let  $\tilde{u}_{n_1, m_1, n_2, m_2, t, \mathbf{x}} \in \mathbb{R}$  be the quadrature solution with an approximated payoff function be given by (269). Note that by Corollary 5.9 (with  $\varepsilon \leftarrow 3\varepsilon/4$  in the notation of Corollary 5.9), it holds that

$$|u(t, \mathbf{x}) - \tilde{u}_{n_1, m_1, n_2, m_2, t, \mathbf{x}}| \leq 3\varepsilon/4. \quad (293)$$

Moreover, by Lemma 5.2 and (276), it holds for every  $\mathbf{j} \in \mathbb{K}_{n_1, m_1, +}^d$  that

$$\begin{aligned} |\tilde{h}_{n_2, m_2}(\mathbf{j})| &\leq |h(\mathbf{j}) - \tilde{h}_{n_2, m_2}(\mathbf{j})| + |h(\mathbf{j})| \\ &\leq C_2 d^2 2^{n_1 - m_2} + C_2^2 d(1 + \|\mathbf{j}\|_1) \\ &\leq C_2 d^2 2^{n_1 - m_2} + C_2^2 d(1 + d 2^{n_1 - 1}) \\ &\leq C_2 d^2 2^{n_1 - m_2} + C_2^2 d^2 2^{n_1} \\ &\leq C_2^2 d^2 2^{n_1 + 1}. \end{aligned} \quad (294)$$

Hence, by (269) and (291), using (285), the fact that  $0 < \gamma, e^{-r(T-t)} \leq 1$ , and the above estimate, we have

$$\begin{aligned} |\tilde{u}_{n_1, m_1, n_2, m_2, t, \mathbf{x}} - \tilde{u}_{n_1, m_1, n_2, m_2, p, t, \mathbf{x}}| &\leq \gamma e^{-r(T-t)} \cdot \max_{\mathbf{j} \in \mathbb{K}_{n_1, m_1, +}^d} |\tilde{h}_{n_2, m_2}(\mathbf{j})| \cdot \sum_{\mathbf{j} \in \mathbb{K}_{n_1, m_1, +}^d} |\tilde{p}_{\mathbf{j}, \varepsilon/4} - \gamma^{-1} p_{\mathbf{j}, m_1}| \\ &\leq \varepsilon/4. \end{aligned} \quad (295)$$

Thus, the conclusion follows from triangle inequality.  $\square$

## 5.5 Step 5: Rotation error bounds

**Proposition 5.11 (Rotation error)** Let  $\varepsilon \in (0, 1)$ ,  $d \in \mathbb{N}$ ,  $r, T \in (0, \infty)$ , and  $(t, \mathbf{x}) \in [0, T) \times \mathbb{R}_+^d$ . Let  $u(t, \mathbf{x})$  be the option price given by (7). Let  $h : \mathbb{R}_+^d \rightarrow \mathbb{R}$  be the CPWA function given by (8). Let Assumption 2.2 and Assumption 2.4 hold with respective constants  $C_1, C_2 \in [1, \infty)$ , and let  $n_2 := 1 + \lceil \log_2(C_2) \rceil$ . For every  $\eta \in (0, 1)$  let  $M_{d, \eta} \in [1, \infty)$  be given by (231), for every  $n_1, m_1, m_2 \in \mathbb{N}$  let  $\{\tilde{p}_{\mathbf{j}, \eta} : \mathbf{j} \in \mathbb{K}_{n_1, m_1, +}^d\} \subset [0, 1]$  satisfy

$$\sum_{\mathbf{j} \in \mathbb{K}_{n_1, m_1, +}^d} \tilde{p}_{\mathbf{j}, \eta} = 1 \quad (296)$$

and

$$\sum_{\mathbf{j} \in \mathbb{K}_{n_1, m_1, +}^d} |\tilde{p}_{\mathbf{j}, \eta} - \gamma^{-1} p_{\mathbf{j}, m_1}| \leq \frac{\eta}{C_2^2 d^2 2^{n_1 + 1}}, \quad (297)$$

where for all  $\mathbf{j} = (j_1, \dots, j_d) \in \mathbb{K}_{n_1, m_1, +}^d$ ,

$$p_{\mathbf{j}, m_1} := \int_{Q_{\mathbf{j}, m_1}} p(\mathbf{y}, T; \mathbf{x}, t) d\mathbf{y}, \quad Q_{\mathbf{j}, m_1} := [j_1, j_1 + 2^{-m_1}) \times \dots \times [j_d, j_d + 2^{-m_1}), \quad (298)$$

and

$$\gamma := \sum_{\mathbf{j} \in \mathbb{K}_{n_1, m_1, +}^d} p_{\mathbf{j}, m_1} \in (0, 1), \quad (299)$$

let  $\mathfrak{s}_{d,\eta} \in (0, \infty)$  be defined by

$$\mathfrak{s}_{d,\eta} := \sqrt{\frac{\eta}{(C_2^2 d^2 2^{n_1+1})^3}}, \quad (300)$$

let  $\tilde{h}_{n_2,m_2} : \mathbb{R}^d \rightarrow \mathbb{R}$  be given by (268), and let  $a_{n_1,n_2,m_1,m_2,\eta} \in [0, 1]$  be the amplitude given by

$$a_{n_1,n_2,m_1,m_2,\eta} = \sum_{\mathbf{j} \in \mathbb{K}_{n_1,m_1,+}^d} \tilde{p}_{\mathbf{j},\eta} \sin^2 \left( \frac{\mathfrak{s}_{d,\eta} \tilde{h}_{n_2,m_2}(\mathbf{j})}{2} + \frac{\pi}{4} \right). \quad (301)$$

Moreover, for every  $n_1, m_1, m_2 \in \mathbb{N}$  satisfying

$$n_1 \geq 1 + \log_2(M_{d,\varepsilon/5}), \quad (302)$$

$$m_1 \geq \log_2(C_2^2 d^2 (\varepsilon/5)^{-1}), \quad (303)$$

$$m_2 \geq 1 + \log_2(M_{d,\varepsilon/5}) + \log_2(C_2 d^2 (\varepsilon/5)^{-1}), \quad (304)$$

let  $\tilde{u}_{n_1,m_1,n_2,m_2,p,a,t,\mathbf{x}} \in \mathbb{R}$  be the truncated quadrature solution with approximated payoff and loaded distribution with rotation given by

$$\tilde{u}_{n_1,m_1,n_2,m_2,p,a,t,\mathbf{x}} := \mathfrak{s}^{-1} \gamma e^{-r(T-t)} (2a - 1), \quad (305)$$

where here

$$\mathfrak{s} := \mathfrak{s}_{d,\varepsilon/5}, \quad \text{and} \quad a := a_{n_1,n_2,m_1,m_2,\varepsilon/5}.$$

Then, the following holds:

(i)

$$\tilde{u}_{n_1,m_1,n_2,m_2,p,a,t,\mathbf{x}} = \mathfrak{s}^{-1} \gamma e^{-r(T-t)} \sum_{\mathbf{j} \in \mathbb{K}_{n_1,m_1,+}^d} \tilde{p}_{\mathbf{j},\varepsilon/5} \sin \left( \mathfrak{s} \tilde{h}_{n_2,m_2}(\mathbf{j}) \right). \quad (306)$$

(ii)

$$|u(t, \mathbf{x}) - \tilde{u}_{n_1,m_1,n_2,m_2,p,a,t,\mathbf{x}}| \leq \varepsilon. \quad (307)$$

*Proof.* First, recall the trigonometric identity that for all  $x \in \mathbb{R}$

$$\sin^2 \left( \frac{x}{2} + \frac{\pi}{4} \right) = 1 - \cos^2 \left( \frac{x}{2} + \frac{\pi}{4} \right) = 1 - \frac{1}{2} (1 + \cos(x + \pi/2)) = \frac{1}{2} + \frac{1}{2} \sin(x). \quad (308)$$

This, (301), (305), and the fact that  $\sum_{\mathbf{j} \in \mathbb{K}_{n_1,m_1,+}^d} \tilde{p}_{\mathbf{j},\varepsilon/5} = 1$  imply that

$$\begin{aligned} \tilde{u}_{n_1,m_1,n_2,m_2,p,a,t,\mathbf{x}} &= \mathfrak{s}^{-1} \gamma e^{-r(T-t)} \sum_{\mathbf{j} \in \mathbb{K}_{n_1,m_1,+}^d} \tilde{p}_{\mathbf{j},\varepsilon/5} \left[ 2 \sin^2 \left( \frac{\mathfrak{s} \tilde{h}_{n_2,m_2}(\mathbf{j})}{2} + \frac{\pi}{4} \right) - 1 \right] \\ &= \mathfrak{s}^{-1} \gamma e^{-r(T-t)} \sum_{\mathbf{j} \in \mathbb{K}_{n_1,m_1,+}^d} \tilde{p}_{\mathbf{j},\varepsilon/5} \sin \left( \mathfrak{s} \tilde{h}_{n_2,m_2}(\mathbf{j}) \right), \end{aligned}$$

which proves Item (i). Next, for the proof of Item (ii), note that by Taylor expansion, one has for any  $x \in [-1, 1]$  the estimate

$$|\sin(x) - x| \leq \left( \frac{|x|^3}{3!} + \frac{|x|^5}{5!} + \frac{|x|^7}{7!} + \dots \right) \leq |x|^3 (e - [1 + \frac{1}{1!} + \frac{1}{2!}]) \leq |x|^3. \quad (309)$$

Moreover, since  $|\tilde{h}_{n_2,m_2}(\mathbf{j})| \leq C_2^2 d^2 2^{n_1+1}$  (c.f. (294)), by (300), it holds for all  $\mathbf{j} \in \mathbb{K}_{n_1,m_1,+}^d$  that

$$|\mathfrak{s} \tilde{h}_{n_2,m_2}(\mathbf{j})| \leq \sqrt{\varepsilon/5} \leq 1. \quad (310)$$

This, (309), (300), and the fact that  $|\tilde{h}_{n_2,m_2}(\mathbf{j})| \leq C_2^2 d^2 2^{n_1+1}$  hence ensure for all  $\mathbf{j} \in \mathbb{K}_{n_1,m_1,+}^d$  that

$$\mathfrak{s}^{-1} |\sin(\mathfrak{s} \tilde{h}_{n_2,m_2}(\mathbf{j})) - \mathfrak{s} \tilde{h}_{n_2,m_2}(\mathbf{j})| \leq \mathfrak{s}^{-1} |\mathfrak{s} \tilde{h}_{n_2,m_2}(\mathbf{j})|^3 = \mathfrak{s}^2 |\tilde{h}_{n_2,m_2}(\mathbf{j})|^3 \leq \varepsilon/5. \quad (311)$$

Let here  $\tilde{u}_{n_1,m_1,n_2,m_2,p,t,\mathbf{x}} \in \mathbb{R}$  be the truncated quadrature solution with approximated payoff and loaded distribution given by

$$\tilde{u}_{n_1,m_1,n_2,m_2,p,t,\mathbf{x}} := \gamma e^{-r(T-t)} \sum_{\mathbf{j} \in \mathbb{K}_{n_1,m_1,+}^d} \tilde{p}_{\mathbf{j},\varepsilon/5} \tilde{h}_{n_2,m_2}(\mathbf{j}).$$

This, (311), (306), and the fact that

$$0 < \gamma e^{-r(T-t)} \sum_{\mathbf{j} \in \mathbb{K}_{n_1,m_1,+}^d} \tilde{p}_{\mathbf{j},\varepsilon/5} \leq 1$$

imply that

$$|\tilde{u}_{n_1,m_1,n_2,m_2,p,t,\mathbf{x}} - \tilde{u}_{n_1,m_1,n_2,m_2,p,a,t,\mathbf{x}}| \leq \varepsilon/5. \quad (312)$$

Furthermore, by Proposition (5.10) (with  $\varepsilon \leftarrow 4\varepsilon/5$  in the notation of Proposition (5.10)), it holds that

$$|u(t, \mathbf{x}) - \tilde{u}_{n_1,m_1,n_2,m_2,p,t,\mathbf{x}}| \leq 4\varepsilon/5. \quad (313)$$

Hence, the conclusion follows from the triangle inequality.  $\square$

## 5.6 Step 6: Quantum amplitude estimation error bounds

**Proposition 5.12 (Combined errors)** Let  $\varepsilon \in (0, 1)$ ,  $d \in \mathbb{N}$ ,  $r, T \in (0, \infty)$ , and  $(t, \mathbf{x}) \in [0, T] \times \mathbb{R}_+^d$ . Let  $u(t, \mathbf{x})$  be the option price given by (7). Let  $h : \mathbb{R}_+^d \rightarrow \mathbb{R}$  be the CPWA function given by (8). Let Assumption 2.2 and Assumption 2.4 hold with respective constants  $C_1, C_2 \in [1, \infty)$ , and let

$$n_2 := 1 + \lceil \log_2(C_2) \rceil \quad (314)$$

For every  $\eta \in (0, 1)$ , let  $M_{d,\eta} \in [1, \infty)$  be given by (231). Let  $n_{1,d,\varepsilon}, m_{1,d,\varepsilon}, m_{2,d,\varepsilon} \in (0, \infty)$  be defined by

$$n_{1,d,\varepsilon} := 1 + \log_2(M_{d,\varepsilon/6}), \quad (315)$$

$$m_{1,d,\varepsilon} := \log_2(C_2^2 d^2 (\varepsilon/6)^{-1}), \quad (316)$$

$$m_{2,d,\varepsilon} := 1 + \log_2(M_{d,\varepsilon/6}) + \log_2(C_2 d^2 (\varepsilon/6)^{-1}). \quad (317)$$

Moreover, for every  $n_1, m_1, m_2 \in \mathbb{N}$  satisfying  $n_1 \geq n_{1,d,\varepsilon}$ ,  $m_1 \geq m_{1,d,\varepsilon}$ , and  $m_2 \geq m_{2,d,\varepsilon}$ , let  $\{\tilde{p}_{\mathbf{j},\varepsilon/6} : \mathbf{j} \in \mathbb{K}_{n_1,m_1,+}^d\} \subset [0, 1]$  satisfy

$$\sum_{\mathbf{j} \in \mathbb{K}_{n_1,m_1,+}^d} \tilde{p}_{\mathbf{j},\varepsilon/6} = 1 \quad (318)$$

and

$$\sum_{\mathbf{j} \in \mathbb{K}_{n_1,m_1,+}^d} |\tilde{p}_{\mathbf{j},\varepsilon/6} - \gamma^{-1} p_{\mathbf{j},m_1}| \leq \frac{\varepsilon}{6C_2^2 d^2 2^{n_1+1}}, \quad (319)$$

where for all  $\mathbf{j} = (j_1, \dots, j_d) \in \mathbb{K}_{n_1,m_1,+}^d$ ,

$$p_{\mathbf{j},m_1} := \int_{Q_{\mathbf{j},m_1}} p(\mathbf{y}, T; \mathbf{x}, t) d\mathbf{y}, \quad Q_{\mathbf{j},m_1} := [j_1, j_1 + 2^{-m_1}) \times \dots \times [j_d, j_d + 2^{-m_1}), \quad (320)$$

and

$$\gamma := \sum_{\mathbf{j} \in \mathbb{K}_{n_1,m_1,+}^d} p_{\mathbf{j},m_1} \in (0, 1), \quad (321)$$

let  $\mathfrak{s} \equiv \mathfrak{s}_{d,\varepsilon/6} \in (0, \infty)$  be defined by

$$\mathfrak{s} \equiv \mathfrak{s}_{d,\varepsilon/6} := \sqrt{\frac{\varepsilon/6}{(C_2^2 d^2 2^{n_1+1})^3}}, \quad (322)$$

let  $\tilde{h}_{n_2,m_2} : \mathbb{R}^d \rightarrow \mathbb{R}$  be given by (268), let  $a \equiv a_{n_1,n_2,m_1,m_2,\varepsilon/6} \in [0, 1]$  be the amplitude given by

$$a \equiv a_{n_1,n_2,m_1,m_2,\mathfrak{s},\varepsilon/6} := \sum_{\mathbf{j} \in \mathbb{K}_{n_1,m_1,+}^d} \tilde{p}_{\mathbf{j},\varepsilon/6} \sin^2 \left( \frac{\mathfrak{s} \tilde{h}_{n_2,m_2}(\mathbf{j})}{2} + \frac{\pi}{4} \right), \quad (323)$$

let  $\hat{a} \in [0, 1]$  satisfy

$$|a - \hat{a}| \leq \frac{\varepsilon \mathfrak{s}}{12}, \quad (324)$$

and let  $\tilde{U}_{t,\mathbf{x}}$  be the approximated solution given by

$$\tilde{U}_{t,\mathbf{x}} := \mathfrak{s}^{-1} \gamma e^{-r(T-t)} (2\hat{a} - 1). \quad (325)$$

Then,

$$|u(t, \mathbf{x}) - \tilde{U}_{t,\mathbf{x}}| \leq \varepsilon. \quad (326)$$

*Proof.* Let here  $\tilde{u}_{n_1,m_1,n_2,m_2,p,a,t,\mathbf{x}} \in \mathbb{R}$  be the truncated quadrature solution with approximated payoff and loaded distribution with rotation given by

$$\tilde{u}_{n_1,m_1,n_2,m_2,p,a,t,\mathbf{x}} := \mathfrak{s}^{-1} \gamma e^{-r(T-t)} (2a - 1).$$

By Proposition 5.11 item (ii) (with  $\varepsilon \leftarrow 5\varepsilon/6$  in the notation of Proposition 5.11), it holds that

$$|u(t, \mathbf{x}) - \tilde{u}_{n_1,m_1,n_2,m_2,p,a,t,\mathbf{x}}| \leq 5\varepsilon/6. \quad (327)$$

Using  $0 < \gamma e^{-r(T-t)} \leq 1$  and (324), it follows that

$$|\tilde{u}_{n_1,m_1,n_2,m_2,p,a,t,\mathbf{x}} - \tilde{U}_{t,\mathbf{x}}| \leq 2\mathfrak{s}^{-1} |a - \hat{a}| \leq \varepsilon/6. \quad (328)$$

Hence, we conclude (326).  $\square$

## 6 Proof of Theorem 1

In this section, we provide the proof of Theorem 1.

*Proof of Theorem 1.* First, let  $n_1, n_2, m_1, m_2, N, \gamma, \mathfrak{s}$  be defined as in line 2–3 of Algorithm 1, and set  $n := n_1 + n_2$  and  $m := m_1 + m_2$ . Let  $\mathbf{h} : \mathbb{F}_{n_1, m_1}^d \rightarrow \mathbb{F}_{n+d+K+1+m}$  be defined as in (200), and let  $p := d(2n_2 + 2m_2 + 3)$  and  $N, \{q_k\}_{k=1}^K \in \mathbb{N}$  be given by (202) from Proposition 4.24. By Assumption 4.16, it holds that

$$\mathcal{P} |0\rangle_{d(n_1+m_1)} = \sum_{\mathbf{i} \in \mathbb{F}_{n_1, m_1, +}^d} \sqrt{\tilde{p}_i} |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1}. \quad (329)$$

Hence, together with Proposition 4.24, the circuit  $\mathcal{A} := \mathcal{R}_h(\mathcal{P} \otimes I_2^{\otimes(N-d(n_1+m_1))})$  satisfies

$$\begin{aligned} \mathcal{A} |0\rangle_N &= \sum_{\mathbf{i} \in \mathbb{F}_{n_1, m_1, +}^d} \sqrt{\tilde{p}_i} |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |\text{anc}\rangle_{q_1+\dots+q_K+2K(n+m+d+5)-1} [\cos(\bar{f}(\mathbf{h}(\mathbf{i}))/2) |0\rangle + \sin(\bar{f}(\mathbf{h}(\mathbf{i}))/2) |1\rangle] \\ &= \sum_{\mathbf{i} \in \mathbb{F}_{n_1, m_1, +}^d} \sqrt{\tilde{p}_i} \cos(\bar{f}(\mathbf{h}(\mathbf{i}))/2) |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |\text{anc}\rangle_{q_1+\dots+q_K+2K(n+m+d+5)-1} |0\rangle \\ &\quad + \sum_{\mathbf{i} \in \mathbb{F}_{n_1, m_1, +}^d} \sqrt{\tilde{p}_i} \sin(\bar{f}(\mathbf{h}(\mathbf{i}))/2) |i_1\rangle_{n_1+m_1} \cdots |i_d\rangle_{n_1+m_1} |\text{anc}\rangle_{q_1+\dots+q_K+2K(n+m+d+5)-1} |1\rangle \\ &=: \sqrt{1-a} |\psi_0\rangle_{N-1} |0\rangle + \sqrt{a} |\psi_1\rangle_{N-1} |1\rangle, \end{aligned} \quad (330)$$

where as in Proposition 4.24,  $\bar{f} : \mathbb{F}_{n+d+K+1,m} \rightarrow \mathbb{R}$  is defined by  $\bar{f}(i) = f \circ D_{n+d+K+1,m}(i)$  and  $f(x) = \mathfrak{s}x + \frac{\pi}{2}$ , and  $a \in [0, 1]$  is given by

$$a := \sum_{\mathbf{i} \in \mathbb{F}_{n_1, m_1, +}^d} \tilde{p}_i \sin^2(\bar{f}(\mathbf{h}(\mathbf{i}))/2). \quad (331)$$

By Proposition 4.24 and Proposition 5.8, we note that the function  $\tilde{h}_{n_2, m_2}$  given in (268) coincides with the function  $D_{n+d+K+1,m}(\mathbf{h})$  when restricted to the domain  $\mathbb{F}_{n_1, m_1}^d$ . Using this and Proposition 5.12 (with  $\varepsilon \leftarrow \frac{\varepsilon}{6C_2^2 d^2 2^{n_1+1}}$  in the notation of Assumption 4.16), we have

$$a = \sum_{\mathbf{i} \in \mathbb{F}_{n_1, m_1, +}^d} \tilde{p}_i \sin^2 \left( \frac{\mathfrak{s} D_{n+d+K+1,m}(\mathbf{h}(\mathbf{i}))}{2} + \frac{\pi}{4} \right) = a_{n_1, n_2, m_1, m_2, \mathfrak{s}, \varepsilon/6}, \quad (332)$$

where  $a_{n_1, n_2, m_1, m_2, \mathfrak{s}, \varepsilon/6}$  is defined in (323). By Proposition 2.20, the output  $\hat{a}$  from line 7 of Algorithm 1 satisfies the bound

$$|a - \hat{a}| \leq \varepsilon \mathfrak{s}/12, \quad \text{with probability at least } 1 - \alpha. \quad (333)$$

Thus, the estimate (59) follows from Proposition 5.12. Next, we count the total number of qubits and elementary gates used to construct circuit  $\mathcal{A}$ . Let  $M_{d, \varepsilon/6}$  be the constant given by (231), and note that

$$M_{d, \varepsilon/6} = 6\mathfrak{c} d^{\frac{5}{2}} \varepsilon^{-1}. \quad (334)$$

Moreover, recall that for any  $v \in \mathbb{R}$ , one has  $\lceil v \rceil \leq v + 1$ . Hence, by (315)–(317) and the bound on  $M_{d, \varepsilon/6}$ , we have the following bounds

$$n_1 = \lceil n_{1, d, \varepsilon} \rceil \leq 2 + \log_2(M_{d, \varepsilon/6}) = 2 + \log_2(2 \cdot 3\mathfrak{c} d^{\frac{5}{2}} \varepsilon^{-1}), \quad (335)$$

$$n_2 \leq 2 + \log_2(C_2) \leq 2 + \log_2(\mathfrak{c}^{\frac{1}{2}}), \quad (336)$$

$$m_1 = \lceil m_{1, d, \varepsilon} \rceil \leq 1 + \log_2(C_2^2 d^2 (\varepsilon/6)^{-1}) \leq 1 + \log_2(2 \cdot 3\mathfrak{c} d^2 \varepsilon^{-1}), \quad (337)$$

$$m_2 = \lceil m_{2, d, \varepsilon} \rceil \leq 2 + \log_2(M_{d, \varepsilon/6}) + \log_2(C_2 d^2 (\varepsilon/6)^{-1}) \leq 2 + \log_2(2^2 3^2 \mathfrak{c}^{\frac{3}{2}} d^{\frac{9}{2}} \varepsilon^{-2}). \quad (338)$$

Furthermore,

$$n + m + 1 = n_1 + n_2 + m_1 + m_2 + 1 \leq 8 + \log_2(2^4 3^4 \mathfrak{c}^4 d^9 \varepsilon^{-4}) = \log_2(2^7 3^4 \mathfrak{c}^4 d^9 \varepsilon^{-4}). \quad (339)$$

By Proposition 4.24 and the fact that  $\mathcal{A} = \mathcal{R}_h(\mathcal{P} \otimes I_2^{\otimes(N-d(n_1+m_1))})$ , the quantum circuit  $\mathcal{A}$  uses  $N$  qubits, where  $N$  is given by (202). Using the upper bound (218) for  $N$ , Assumption 2.4, and (339), we thus have the following

bound on the number of qubits used for the circuit  $\mathcal{A}$

$$\begin{aligned}
N &\leq 24K \cdot \max_{k=1,\dots,K} I_k \cdot d(n+m+1) \\
&\leq 24C_2d \cdot d(n+m+1) \\
&\leq 24C_2d^2 \log_2(2^7 3^4 \mathfrak{c}^4 d^9 \varepsilon^{-4}) \\
&\leq 24C_2d^2 (\log_2(2^7 3^4) + 4\log_2(\mathfrak{c}) + 9\log_2(d\varepsilon^{-1})) \\
&\leq 24C_2d^2 (14 + 4\log_2(\mathfrak{c}) + 9\log_2(d\varepsilon^{-1})) \\
&\leq 24C_2d^2 \cdot 27\log_2(\mathfrak{c})(1 + \log_2(d\varepsilon^{-1})) \\
&\leq 648C_2\log_2(\mathfrak{c})d^2(1 + \log_2(d\varepsilon^{-1})) \\
&=: \mathfrak{C}_1 d^2(1 + \log_2(d\varepsilon^{-1})).
\end{aligned} \tag{340}$$

Next, we count the number of elementary gates used to construct the quantum circuit  $\mathcal{A}$ , which will be denoted by  $N_{\mathcal{A}}$ . By Assumption 4.16, the number of elementary gates used to construct  $\mathcal{P}$  is at most (127) with  $(n \leftarrow n_1, m \leftarrow m_1, \varepsilon \leftarrow \frac{\varepsilon}{6C_2^2 d^2 2^{n_1+1}})$  in the notation of Assumption 4.16). Hence, using (335) and (337), the number of elementary gates used to construct  $\mathcal{P}$  is bounded by

$$\begin{aligned}
&C_3(n_1 + m_1)^{C_3} d^{C_3} (\log_2(6\varepsilon^{-1} C_2^2 d^2 2^{n_1+1}))^{C_3} \\
&= C_3(n_1 + m_1)^{C_3} d^{C_3} (\log_2(6C_2^2 d^2 \varepsilon^{-1}) + n_1 + 1)^{C_3} \\
&\leq C_3(3 + \log_2(2^2 3^2 \mathfrak{c}^2 d^{\frac{9}{2}} \varepsilon^{-2}))^{C_3} d^{C_3} (\log_2(2 \cdot 3\mathfrak{c} d^2 \varepsilon^{-1}) + 3 + \log_2(2 \cdot 3\mathfrak{c} d^{\frac{5}{2}} \varepsilon^{-1}))^{C_3} \\
&= C_3 d^{C_3} (\log_2(2^5 3^2 \mathfrak{c}^2 d^{\frac{9}{2}} \varepsilon^{-2}))^{2C_3}.
\end{aligned} \tag{341}$$

By Proposition 4.24, Assumption (2.4), and the bound (339), the number of elementary gates used to construct  $\mathcal{R}_h$  is estimated by at most

$$16186K^3 (\max\{I_1, \dots, I_k\})^3 d^3 (n+m+1)^3 \leq 16186(C_2d)^3 d^3 (\log_2(2^7 3^4 \mathfrak{c}^4 d^9 \varepsilon^{-4}))^3. \tag{342}$$

Hence, summing up (341) and (342), the number of elementary gates used to construct quantum circuit  $\mathcal{A}$  is at most

$$N_{\mathcal{A}} \leq C_3 d^{C_3} (\log_2(2^5 3^2 \mathfrak{c}^2 d^{\frac{9}{2}} \varepsilon^{-2}))^{2C_3} + 16186(C_2d)^3 d^3 (\log_2(2^7 3^4 \mathfrak{c}^4 d^9 \varepsilon^{-4}))^3. \tag{343}$$

Next, we count the number of elementary gates used in line 7 of Algorithm 1, which by Remark 2.21 Item 2. coincides with the number of elementary gates used to construct the quantum circuit  $\mathcal{Q}^{k_t} \mathcal{A}$  in the Modified IQAE algorithm. Note that this number is also the number of elementary gates used in Algorithm 1. Using (322) and (335), the number  $\mathfrak{s}^{-1}$  is bounded by

$$\mathfrak{s}^{-1} = (6\varepsilon^{-1} (C_2^2 d^2 2^{n_1+1})^3)^{\frac{1}{2}} \leq \sqrt{6}\varepsilon^{-\frac{1}{2}} C_2 d \left( 2^{3+\log_2(6\mathfrak{c} d^{\frac{5}{2}} \varepsilon^{-1})} \right)^{\frac{3}{2}} = 2^{\frac{13}{2}} 3^2 C_2 \mathfrak{c}^{\frac{3}{2}} d^{\frac{19}{4}} \varepsilon^{-2}. \tag{344}$$

Thus, by using Proposition 2.20 Item 3. (with  $\mathcal{A} \leftarrow \mathcal{A}$ ,  $\varepsilon \leftarrow \varepsilon \mathfrak{s}/12$ ,  $n \leftarrow N-1$ , and  $N \leftarrow N_{\mathcal{A}}$  in the Notation of Proposition 2.20) together with (340) and (343), we conclude that the number of elementary gates used in

Algorithm 1 is bounded by

$$\begin{aligned}
& \frac{\pi}{4^{\frac{\varepsilon 5}{12}}} (8N^2 + 23 + N_{\mathcal{A}}) \\
& \leq 2^{\frac{13}{2}} 3^3 \pi C_2 \mathfrak{c}^{\frac{3}{2}} d^{\frac{19}{4}} \varepsilon^{-3} \left[ 8(24C_2 d^2 \log_2(2^7 3^4 \mathfrak{c}^4 d^9 \varepsilon^{-4}))^2 + 23 \right. \\
& \quad \left. + C_3 d^{C_3} (\log_2(2^5 3^2 \mathfrak{c}^2 d^{\frac{9}{2}} \varepsilon^{-2}))^{2C_3} + 16186 (C_2 d)^3 d^3 (\log_2(2^7 3^4 \mathfrak{c}^4 d^9 \varepsilon^{-4}))^3 \right] \\
& \leq 2^{\frac{13}{2}} 3^3 \pi C_2^4 C_3 d^{\max\{10.75, 4.75 + C_3\}} \varepsilon^{-3} \left[ 8 \cdot 24^2 (\log_2(2^7 3^4 \mathfrak{c}^4 d^9 \varepsilon^{-4}))^2 + 23 \right. \\
& \quad \left. + (\log_2(2^5 3^2 \mathfrak{c}^2 d^{\frac{9}{2}} \varepsilon^{-2}))^{2C_3} + 16186 (\log_2(2^7 3^4 \mathfrak{c}^4 d^9 \varepsilon^{-4}))^3 \right] \\
& \leq 2^{\frac{13}{2}} 3^3 \pi C_2^4 C_3 d^{\max\{10.75, 4.75 + C_3\}} \varepsilon^{-3} \left[ 8 \cdot 24^2 (\log_2(2^7 3^4) + 4 \log_2(\mathfrak{c}) + 9)^2 (1 + \log_2(d\varepsilon^{-1}))^2 + 23 \right. \\
& \quad \left. + (\log_2(2^5 3^2) + 2 \log_2(\mathfrak{c}) + \frac{9}{2})^{2C_3} (1 + \log_2(d\varepsilon^{-1}))^{2C_3} \right. \\
& \quad \left. + 16186 (\log_2(2^7 3^4) + 4 \log_2(\mathfrak{c}) + 9)^3 (1 + \log_2(d\varepsilon^{-1}))^3 \right] \tag{345} \\
& \leq 2^{\frac{13}{2}} 3^3 \pi \left[ 8 \cdot 24^2 \cdot ((14 + 4 + 9) \log_2(\mathfrak{c}))^2 + 23 + ((8.2 + 2 + 4.5) \log_2(\mathfrak{c}))^{2C_3} \right. \\
& \quad \left. + 16186 ((14 + 4 + 9) \log_2(\mathfrak{c}))^3 \right] C_2^4 C_3 d^{\max\{10.75, 4.75 + C_3\}} \varepsilon^{-3} (1 + \log_2(d\varepsilon^{-1}))^{\max\{3, 2C_3\}} \\
& \leq 2^{\frac{13}{2}} 3^3 \pi \left[ 8 \cdot 24^2 + 23 + 1 + 16186 \right] \\
& \quad \cdot C_2^4 C_3 (27 \log_2(\mathfrak{c}))^{\max\{3, 2C_3\}} d^{\max\{10.75, 4.75 + C_3\}} \varepsilon^{-3} (1 + \log_2(d\varepsilon^{-1}))^{\max\{3, 2C_3\}} \\
& \leq (1.6 \times 10^8) C_2^4 C_3 (27 \log_2(\mathfrak{c}))^{\max\{3, 2C_3\}} d^{\max\{10.75, 4.75 + C_3\}} \varepsilon^{-3} (1 + \log_2(d\varepsilon^{-1}))^{\max\{3, 2C_3\}} \\
& =: \mathfrak{C}_2 d^{\max\{10.75, 4.75 + C_3\}} \varepsilon^{-3} (1 + \log_2(d\varepsilon^{-1}))^{\max\{3, 2C_3\}}.
\end{aligned}$$

Lastly, we count the number of applications on  $\mathcal{A}$ . Using (47) (with  $\varepsilon \leftarrow \varepsilon 5/12$  and  $\alpha \leftarrow \alpha$  in the notation Proposition 2.20) and bound for  $\mathfrak{s}^{-1}$  (c.f. (344)), the number of applications on  $\mathcal{A}$  is at most

$$\frac{62 \cdot 12}{\varepsilon 5} \ln\left(\frac{21}{\alpha}\right) \leq 62 \cdot 12 \cdot 2^{\frac{13}{2}} 3^2 C_2 \mathfrak{c}^{\frac{3}{2}} d^{\frac{19}{4}} \varepsilon^{-3} \ln\left(\frac{21}{\alpha}\right) \leq (6.1 \times 10^5) C_2 \mathfrak{c}^{\frac{3}{2}} d^{4.75} \varepsilon^{-3} \ln\left(\frac{21}{\alpha}\right) =: \mathfrak{C}_3 d^{4.75} \varepsilon^{-3} \ln\left(\frac{21}{\alpha}\right). \tag{346}$$

□

## 7 Conclusion

In this paper we have developed with Algorithm 1 a quantum Monte Carlo algorithm to approximately solve (potentially high-dimensional) Black-Scholes PDEs. The contributions of this paper are the following.

First, our algorithm allows the payoff function to be of general form and is only required to be CPWA. From a financial point of view, this is not very restrictive, as most European options are CPWA, see also the various relevant examples provided in Example 2.5. This extends the existing quantum algorithms which typically require the CPWA payoff function to be either one-dimensional or to be a basket option.

Moreover, we provided a mathematical rigorous error and complexity analysis of Algorithm 1, which we see as our main contribution of the paper. This allows us to prove that the computational complexity of the algorithm only grows polynomially in the space dimension  $d$  of the PDE and the prescribed reciprocal of the accuracy  $\varepsilon$ . In addition, we see that for CPWA payoff functions which are uniformly bounded, the computational running time of Algorithm 1 scales  $O(\varepsilon^{-3/2})$ . Therefore, compared to classical (i.e. non quantum-based) Monte Carlo algorithms which scale  $O(\varepsilon^{-2})$ , we indeed have proved that Algorithm 1 provides a speed-up.

Furthermore, we have developed a package we named **qfinance** within the **Qiskit** framework which can be used to run Algorithm 1 on a computer for the case  $d = 1, 2$ . The **OptionPricing** class within this package enables the user to input all parameters of the underlying stocks, to choose the class of CPWA payoff functions within the ones presented in Example 2.5, as well as to specify to error tolerance level. We numerically demonstrated the applicability of our algorithm in this low-dimensional setting. Moreover, we have discussed the scalability of our algorithm by explaining how one could extend our code for the general  $d$ -dimensional setting. We also highlighted that the limitation of the numerical simulation are not caused by our developed Algorithm 1, but due to the limited quantum computing hardware currently available.

We emphasize that the outline of Algorithm 1, namely to approximate the solution of the Black-Scholes PDE via its Feynman-Kac representation by first uploading the transition probability of the underlying (log-normally distributed) SDE, followed by the uploading of the payoff function, and then applying a Quantum amplitude estimation algorithm to estimate the solution of the PDE is not new and has been already applied, e.g., in [15, 63, 69]. However, so far, no mathematical rigorous error and complexity analysis of such a quantum Monte Carlo algorithm to solve Black-Scholes PDEs, or any quantum based algorithm to solve PDEs, has been provided in the literature, which was the main goal of this paper.

# Acknowledgment

Financial support by the Nanyang Assistant Professorship Grant (NAP Grant) *Machine Learning based Algorithms in Finance and Insurance* and the grant NRF2021-QEP2-02-P06 is gratefully acknowledged.

# References

- [1] Scott Aaronson and Patrick Rall. “Quantum approximate counting, simplified”. In: (2020), pp. 24–32.
- [2] Yves Achdou and Olivier Pironneau. *Computational methods for option pricing*. SIAM, 2005.
- [3] J. Aitchison and J.A.C. Brown. *The Lognormal Distribution*. Cambridge University Press, 1957.
- [4] JJ Alvarez-Sanchez, JV Álvarez-Bravo, and LM Nieto. “A quantum architecture for multiplying signed integers”. In: *Journal of Physics: Conference Series* 128 (2008), p. 012013. DOI: 10.1088/1742-6596/128/1/012013.
- [5] Dong An, Noah Linden, Jin-Peng Liu, Ashley Montanaro, Changpeng Shao, and Jiasu Wang. “Quantum-accelerated multilevel Monte Carlo methods for stochastic differential equations in mathematical finance”. In: *Quantum* 5 (2021), p. 481.
- [6] Juan Miguel Arrazola, Timjan Kalajdzievski, Christian Weedbrook, and Seth Lloyd. “Quantum algorithm for nonhomogeneous linear partial differential equations”. In: *Physical Review A* 100.3 (2019), p. 032306.
- [7] Amine Assouel, Antoine Jacquier, and Alexei Kondratyev. “A quantum generative adversarial network for distributions”. In: *Quantum Machine Intelligence* 4.2 (2022), p. 28.
- [8] Christian Beck, Sebastian Becker, Patrick Cheridito, Arnulf Jentzen, and Ariel Neufeld. “Deep splitting method for parabolic PDEs”. In: *SIAM Journal on Scientific Computing* 43.5 (2021), A3135–A3154.
- [9] Julius Berner, Philipp Grohs, and Arnulf Jentzen. “Analysis of the generalization error: Empirical risk minimization over deep artificial neural networks overcomes the curse of dimensionality in the numerical approximation of Black–Scholes partial differential equations”. In: *SIAM Journal on Mathematics of Data Science* 2.3 (2020), pp. 631–657.
- [10] Fischer Black and Myron Scholes. “The pricing of options and corporate liabilities”. In: *Journal of political economy* 81.3 (1973), pp. 637–654.
- [11] Phelim P Boyle. “Options: A monte carlo approach”. In: *Journal of financial economics* 4.3 (1977), pp. 323–338.
- [12] Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. “Quantum amplitude amplification and estimation”. In: *Contemporary Mathematics* 305 (2002), pp. 53–74.
- [13] Randal E. Bryant and David R. O’Hallaron. *Computer Systems: A Programmer’s Perspective*. Prentice Hall, 2003. ISBN: 9780131784567.
- [14] Giuseppe Campolieti and Roman N. Makarov. *Financial Mathematics: A Comprehensive Treatment*. Textbooks in Mathematics. CRC Press, 2014. ISBN: 9781439892435.
- [15] Shouvanik Chakrabarti, Rajiv Krishnakumar, Guglielmo Mazzola, Nikitas Stamatopoulos, Stefan Woerner, and William J Zeng. “A threshold for quantum advantage in derivative pricing”. In: *Quantum* 5 (2021), p. 463.
- [16] Yen-Jui Chang, Wei-Ting Wang, Hao-Yuan Chen, Shih-Wei Liao, and Ching-Ray Chang. “A novel approach for quantum financial simulation and quantum state preparation”. In: *arXiv preprint arXiv:2308.01844* (2023).
- [17] Marco Chiani, Davide Dardari, and Marvin K. Simon. “New exponential bounds and approximations for the computation of error probability in fading channels”. In: *IEEE Transactions on Wireless Communications* 2.4 (2003), pp. 840–845. DOI: 10.1109/TWC.2003.814350.
- [18] Andrew M Childs, Jin-Peng Liu, and Aaron Ostrander. “High-precision quantum algorithms for partial differential equations”. In: *Quantum* 5 (2021), p. 574.
- [19] Michael G Crandall, Hitoshi Ishii, and Pierre-Louis Lions. “User’s guide to viscosity solutions of second order partial differential equations”. In: *Bulletin of the American mathematical society* 27.1 (1992), pp. 1–67.
- [20] Steven A. Cuccaro, Thomas G. Draper, Samuel A. Kutin, and David Petrie Moulton. “A new quantum ripple-carry addition circuit”. In: (2004). arXiv: quant-ph/0410184 [quant-ph].
- [21] Josh Dees, Antoine Jacquier, and Sylvain Laizet. “Unsupervised Random Quantum Networks for PDEs”. In: *arXiv preprint arXiv:2312.14975* (2023).
- [22] Cirq Developers. *Cirq*. Version v1.1.0. See full list of authors on Github: <https://github.com/quantumlib/Cirq/graphs/contributors>. Dec. 2022. DOI: 10.5281/zenodo.7465577.
- [23] João F Doriguello, Alessandro Luongo, Jinge Bao, Patrick Rebentrost, and Miklos Santha. “Quantum algorithm for stochastic optimal stopping problems with applications in finance”. In: *17th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.
- [24] Thomas Draper. “Addition on a Quantum Computer”. In: (Sept. 2000).
- [25] David S. Dummit and Richard M. Foote. *Abstract Algebra*. Wiley, 2003. ISBN: 9780471433347.
- [26] Daniel J Egger, Claudio Gambella, Jakub Marecek, Scott McFaddin, Martin Mevissen, Rudy Raymond, Andrea Simonetto, Stefan Woerner, and Elena Yndurain. “Quantum computing for finance: State-of-the-art and future prospects”. In: *IEEE Transactions on Quantum Engineering* 1 (2020), pp. 1–24.
- [27] Dennis Elbrächter, Philipp Grohs, Arnulf Jentzen, and Christoph Schwab. “DNN expression rate analysis of high-dimensional PDEs: Application to option pricing”. In: *Constructive Approximation* 55.1 (2022), pp. 3–71.
- [28] Filipe Fontanela, Antoine Jacquier, and Mugad Oumgari. “A quantum algorithm for linear PDEs arising in finance”. In: *SIAM Journal on Financial Mathematics* 12.4 (2021), SC98–SC114.
- [29] Shion Fukuzawa, Christopher Ho, Sandy Irani, and Jasen Zion. “Modified iterative quantum amplitude estimation is asymptotically optimal”. In: *2023 Proceedings of the Symposium on Algorithm Engineering and Experiments (ALENEX)*. SIAM, 2023, pp. 135–147.
- [30] Manfred Gilli, Dietmar Maringer, and Enrico Schumann. *Numerical methods and optimization in finance*. Academic Press, 2019.



- [31] Tudor Giurgica-Tiron, Iordanis Kerenidis, Farrokh Labib, Anupam Prakash, and William Zeng. “Low depth algorithms for quantum amplitude estimation”. In: *Quantum* 6 (2022), p. 745.
- [32] Lukas Gonon and Antoine Jacquier. “Universal approximation theorem and error bounds for quantum neural networks and quantum reservoirs”. In: *arXiv preprint arXiv:2307.12904* (2023).
- [33] Dmitry Grinko, Julien Gacon, Christa Zoufal, and Stefan Woerner. “Iterative quantum amplitude estimation”. In: *npj Quantum Information* 7.1 (2021), pp. 1–6.
- [34] Philipp Grohs, Fabian Hornung, Arnulf Jentzen, and Philippe von Wurstemberger. *A proof that artificial neural networks overcome the curse of dimensionality in the numerical approximation of Black–Scholes partial differential equations*. Vol. 284. 1410. American Mathematical Society, 2023.
- [35] Lov Grover and Terry Rudolph. “Creating superpositions that correspond to efficiently integrable probability distributions”. In: (Sept. 2002).
- [36] Lov K. Grover. “A Fast Quantum Mechanical Algorithm for Database Search.” In: *STOC*. Ed. by Gary L. Miller. ACM, 1996, pp. 212–219. ISBN: 0-89791-785-5.
- [37] Jiequn Han, Arnulf Jentzen, and Weinan E. “Solving high-dimensional partial differential equations using deep learning”. In: *Proceedings of the National Academy of Sciences* 115.34 (2018), pp. 8505–8510.
- [38] Steven Herbert. “Quantum Monte Carlo integration: the full advantage in minimal circuit depth”. In: *Quantum* 6 (2022), p. 823.
- [39] Martin Hutzenthaler, Arnulf Jentzen, and Philippe von Wurstemberger. “Overcoming the curse of dimensionality in the approximate pricing of financial derivatives with default risks”. In: *Electronic Journal of Probability* 25 (2020), pp. 1–73.
- [40] Jason Iaconis, Sonika Johri, and Elton Yechao Zhu. “Quantum State Preparation of Normal Distributions using Matrix Product States”. In: *arXiv preprint arXiv:2303.01562* (2023).
- [41] Antoine Jacquier, Oleksiy Kondratyev, Gordon Lee, and Mugad Oumgari. “Quantum Computing for Financial Mathematics”. In: *arXiv preprint arXiv:2311.06621* (2023).
- [42] Antoine Jacquier, Oleksiy Kondratyev, Alexander Lipton, and Marcos Lopez de Prado. *Quantum Machine Learning and Optimisation in Finance: On the Road to Quantum Advantage*. Packt Publishing Ltd, 2022.
- [43] Emanuel Knill. “Approximation by quantum circuits”. In: *arXiv preprint quant-ph/9508006* (1995).
- [44] Kenji Kubo, Koichi Miyamoto, Kosuke Mitarai, and Keisuke Fujii. “Pricing Multi-asset Derivatives by Variational Quantum Algorithms”. In: *IEEE Transactions on Quantum Engineering* (2023).
- [45] Noah Linden, Ashley Montanaro, and Changpeng Shao. “Quantum vs. classical algorithms for solving the heat equation”. In: *Communications in Mathematical Physics* 395.2 (2022), pp. 601–641.
- [46] Alberto Manzano, Andrés Gómez, and CESGA Carlos Vázquez. “D5. 7: Update of review of state-of-the-art for Pricing and Computation of VaR”. In: (2023).
- [47] Alberto Manzano, Daniele Musso, and Álvaro Leitao. “Real quantum amplitude estimation”. In: *EPJ Quantum Technology* 10.1 (2023), pp. 1–24.
- [48] Dan C. Marinescu. *Classical and Quantum Information*. Elsevier Science, 2011. ISBN: 9780123838759.
- [49] Robert C Merton. “Theory of rational option pricing”. In: *The Bell Journal of economics and management science* (1973), pp. 141–183.
- [50] Ashley Montanaro. “Quantum speedup of Monte Carlo methods”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 471.2181 (2015), p. 20150301. DOI: 10.1098/rspa.2015.0301. eprint: <https://royalsocietypublishing.org/doi/pdf/10.1098/rspa.2015.0301>.
- [51] Ashley Montanaro and Sam Pallister. “Quantum algorithms and the finite element method”. In: *Physical Review A* 93.3 (2016), p. 032324.
- [52] Kouhei Nakaji. “Faster amplitude estimation”. In: *Quantum Information and Computation* 20.13&14 (2020), pp. 1109–1122. DOI: 10.26421/QIC20.13-14-2. URL: <https://doi.org/10.26421/QIC20.13-14-2>.
- [53] Ariel Neufeld, Antonis Papapantoleon, and Qikun Xiang. “Model-free bounds for multi-asset options using option-implied information and their exact computation”. In: *Management Science* (2022).
- [54] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. 10th. USA: Cambridge University Press, 2011. ISBN: 1107002176.
- [55] Román Orús, Samuel Mugel, and Enrique Lizaso. “Quantum computing for finance: Overview and prospects”. In: *Reviews in Physics* 4 (2019), p. 100028.
- [56] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L O’Brien. “A variational eigenvalue solver on a photonic quantum processor”. In: *Nature communications* 5.1 (2014), pp. 1–7.
- [57] Kirill Plekhanov, Matthias Rosenkranz, Mattia Fiorentini, and Michael Lubasch. “Variational quantum amplitude estimation”. In: *Quantum* 6 (2022), p. 670.
- [58] Rafał Pracht. “Quantum Binomial Tree, an effective method for probability distribution loading for derivative pricing”. In: *Available at SSRN 4216595* (2023).
- [59] Qiskit contributors. *Qiskit: An Open-source Framework for Quantum Computing*. 2023. DOI: 10.5281/zenodo.2573505.
- [60] Patrick Rall and Bryce Fuller. “Amplitude Estimation from Quantum Signal Processing”. In: *Quantum* 7 (2023), p. 937.
- [61] Sergi Ramos-Calderer, Adrián Pérez-Salinas, Diego García-Martín, Carlos Bravo-Prieto, Jorge Cortada, Jordi Planaguma, and José I Latorre. “Quantum unary approach to option pricing”. In: *Physical Review A* 103.3 (2021), p. 032414.
- [62] Pooja Rao, Kwangmin Yu, Hyunkyung Lim, Dasol Jin, and Deokkyu Choi. “Quantum amplitude estimation algorithms on IBM quantum devices”. In: *Quantum Communications and Quantum Imaging XVIII*. Vol. 11507. SPIE. 2020, pp. 49–60.
- [63] Patrick Rebentrost, Brajesh Gupta, and Thomas R. Bromley. “Quantum computational finance: Monte Carlo pricing of financial derivatives”. In: *Phys. Rev. A* 98 (2 2018), p. 022321. DOI: 10.1103/PhysRevA.98.022321.
- [64] Patrick Rebentrost and Seth Lloyd. “Quantum computational finance: quantum algorithm for portfolio optimization”. In: *arXiv preprint arXiv:1811.03975* (2018).

- [65] Lidia Ruiz-Perez and Juan Carlos Garcia-Escartin. “Quantum arithmetic with the quantum Fourier transform”. In: *Quantum Information Processing* 16.6 (2017). issn: 1573-1332. DOI: 10.1007/s11128-017-1603-1.
- [66] Mehdi Saeedi and Massoud Pedram. “Linear-depth quantum circuits for n-qubit Toffoli gates with no ancilla”. In: *Physical Review A* 87.6 (2013), p. 062318.
- [67] Engin Şahin. “Quantum arithmetic operations based on quantum fourier transform on signed integers”. In: *International Journal of Quantum Information* 18.06 (2020), p. 2050035.
- [68] Zbyněk Šidák. “Rectangular Confidence Regions for the Means of Multivariate Normal Distributions”. In: *Journal of the American Statistical Association* 62.318 (1967), pp. 626–633. DOI: 10.1080/01621459.1967.10482935. eprint: <https://doi.org/10.1080/01621459.1967.10482935>.
- [69] Nikitas Stamatopoulos, Daniel J Egger, Yue Sun, Christa Zoufal, Raban Iten, Ning Shen, and Stefan Woerner. “Option pricing using quantum computers”. In: *Quantum* 4 (2020), p. 291.
- [70] Yohichi Suzuki, Shumpei Uno, Rudy Raymond, Tomoki Tanaka, Tamiya Onodera, and Naoki Yamamoto. “Amplitude estimation without phase estimation”. In: *Quantum Information Processing* 19.2 (2020), pp. 1–17.
- [71] Camille de Valk, Ankur Raina, Julian van Velzen, et al. “Quantum state preparation for bell-shaped probability distributions using deconvolution methods”. In: *arXiv preprint arXiv:2310.05044* (2023).
- [72] Vlatko Vedral, Adriano Barenco, and Artur Ekert. “Quantum networks for elementary arithmetic operations”. In: *Physical Review A* 54.1 (1996), 147–153. issn: 1094-1622. DOI: 10.1103/physreva.54.147.
- [73] Chu-Ryang Wie. “Simpler quantum counting”. In: *Quantum Information and Computation* 19.11 and 12 (), pp. 0967–0983.
- [74] Stefan Woerner and Daniel J. Egger. “Quantum risk analysis”. In: *npj Quantum Information* 5.1 (2019). issn: 2056-6387. DOI: 10.1038/s41534-019-0130-6.
- [75] Yunpeng Zhao, Haiyan Wang, Kuai Xu, Yue Wang, Ji Zhu, and Feng Wang. “Adaptive Algorithm for Quantum Amplitude Estimation”. In: *arXiv preprint arXiv:2206.08449* (2022).
- [76] Christa Zoufal, Aurélien Lucchi, and Stefan Woerner. “Quantum Generative Adversarial Networks for learning and loading random distributions”. In: *npj Quantum Information* 5.1 (2019), p. 103. issn: 2056-6387. DOI: 10.1038/s41534-019-0223-2.