

QUANTUM MONTE CARLO INTEGRATION FOR SIMULATION-BASED OPTIMISATION

JINGJING CUI ², PHILIPPE J.S. DE BROUWER ³, STEVEN HERBERT ¹, PHILIP INTALLURA ⁴, CAHIT KARGI ¹, GEORGIOS KORPAS ^{5,6,7}, ALEXANDRE KRAJENBRINK ², WILLIAM SHOOSMITH ⁴, IFAN WILLIAMS ¹, BAN ZHENG ⁸

¹ Quantinuum, Terrington House, 13–15 Hills Road, Cambridge CB2 1NL, United Kingdom

² Quantinuum, Partnership House, Carlisle Place, London SW1P 1BX, United Kingdom

³ HSBC Service Delivery Sp. z o.o., Kapelanka 42A, Krakow, Poland

⁴ Quantum Technologies Group, Emerging Technology, Innovation & Ventures, HSBC, 8 Canada Square, E14 5HQ, London, United Kingdom

⁵ Quantum Technologies Group, Emerging Technology, Innovation & Ventures, HSBC, 20 Pasir Panjang Road, 117439, Singapore

⁶ Department of Computer Science, Czech Technical University in Prague, Karlovo nam. 13, Prague 2, Czech Republic

⁷ Archimedes Research Unit on AI, Data Science and Algorithms, Athena Research Center, 15125, Marousi, Greece

⁸ HSBC Global Asset Management, Esplanade Du Gal De Gaulle, Courbevoie Hauts-De-Seine, 92400, Paris, France

ABSTRACT. We investigate the feasibility of integrating quantum algorithms as subroutines of simulation-based optimisation problems with relevance to and potential applications in mathematical finance. To this end, we conduct a thorough analysis of all systematic errors arising in the formulation of quantum Monte Carlo integration in order to better understand the resources required to encode various distributions such as a Gaussian, and to evaluate statistical quantities such as the Value-at-Risk (VaR) and Conditional-Value-at-Risk (CVaR) of an asset. Finally, we study the applicability of quantum Monte Carlo integration for fundamental financial use cases in terms of simulation-based optimisations, notably Mean-Conditional-Value-at-Risk (Mean-CVaR) and (risky) Mean-Variance (Mean-Var) optimisation problems. In particular, we study the Mean-Var optimisation problem in the presence of noise on a quantum device, and benchmark a quantum error mitigation method that applies to quantum amplitude estimation – a key subroutine of quantum Monte Carlo integration – showcasing the utility of such an approach.

CONTENTS

1. Introduction	2
2. Glossary and definitions of financial concepts	6
3. Quantum enhancements of Monte Carlo integration	7

4. Systematic errors in quantum Monte Carlo integration	12
5. Estimating (Conditional-)Value-at-Risk using the quantum Monte Carlo integration engine	15
6. Mean-CVaR simulation-based portfolio optimisation over stable distributions	25
7. Asset allocation under uncertainty and Mean-Variance optimisation	27
8. Conclusions	35
Acknowledgements	36
References	36
Appendix A. Analytical derivation of the utility function and minimum for the Mean-Var portfolio problem	42
Appendix B. Stable distributions and extended interesting distributions	44
Appendix C. Error analysis for the Lévy distribution	45
Appendix D. Portfolio optimisation with Conditional-Drawdown-at-Risk	48

1. INTRODUCTION

Financial institutions routinely face a multitude of problems that involve numerical computations. These include forecasting problems, such as pricing, risk estimation, anomaly detection, and understanding customer preferences, as well as optimisation problems such as portfolio selection, optimal trading-strategy development, and hedging. More often than not, such problems include significant theoretical and practical challenges that are difficult to overcome. These challenges arise from both the complexity associated with the underlying financial models and the limitations of traditional computational methods, even in the current era of advanced parallel computing platforms enabled by modern GPU architectures. Investigating how such challenges could be tackled using quantum computing (QC) is currently an overarching effort of the quantum community, and in particular applications of QC for financial modeling are of great promise [1, 2].

Quantum computing for financial modeling. Both academia and industry have made remarkable progress in exploring the potential of QC to address problems that are classically computationally resource intensive or difficult to solve, offering speedups ranging from polynomial to exponential [3–5]. The focus has predominantly been on protocols for optimisation, machine learning, and stochastic simulation, leading to the development of various QC algorithms tailored to financial applications that make use of the above protocols as underlying subroutines. For instance, quantum algorithms for Monte Carlo integration (MCI) promise a quadratic speedup in convergence compared to classical methods, offering significant prospects in risk modelling, where Monte Carlo methods are vital [6–10]. For optimisation and machine learning, certain quantum algorithms have shown advantages in discrete optimisation, dynamic programming, boosting, and clustering, sometimes demonstrating polynomially-improved complexity compared to classical algorithms [11]. Furthermore, heuristic approaches, frequently employing classical-quantum hybrid methodologies that make use of variational quantum circuits, have been actively researched [11, 12]. Given the broad applicability of these algorithms, great promise is held for financial applications. Within finance, the inherent strengths of QC make it particularly suited for addressing challenges in computational finance with examples such as risk analysis, risk assessment and portfolio management [13–16], option and derivative pricing [17–19], and collateral optimisation [20]. Although achieving fully fault-tolerant quantum computation remains an ongoing challenge, progress is being made [21–24], with the aim of running practical applications on quantum machines [25].

While the potential of QC for finance is promising [14, 26], it is important to acknowledge that there exist several current limitations and challenges. On the quantum-algorithmic side, solving

optimisation problems using quantum computers remains challenging, often requiring the use of heuristic algorithms run on near-term hardware [11], which offer no guarantees on convergence times or approximation ratios. Nevertheless, there are continual development of new protocols for quantum optimisation that provide empirical confidence for potential computational advantage; see e.g., Ref. [27]. In this paper, we focus on the application of QC to problems related to optimisation in financial modeling, and thus we must first introduce the topic and discuss some of its unique challenges.

Optimisation in financial modeling. Optimisation in financial modeling is of great importance for applications within finance, however it is made difficult by a number of fundamental challenges. One challenge is that data is often high dimensional. For example, in portfolio optimisation and risk management, often one encounters the problem of estimating the covariance matrix (or its inverse) of the asset returns in the portfolio. For instance, with 500 stocks under consideration for asset allocation, the covariance matrix has 125, 250 parameters [28]. Furthermore, phenomena that occur in high-dimensional data, such as collinearities or spurious correlations, add significant complications. This challenge is often referred to as the curse of dimensionality and it is a fundamental limitation, especially in applications within risk management and derivative pricing. Intensive Monte Carlo simulations [29,30] are the main tool employed in these scenarios, and are considered well suited for tackling this problem – often facilitated by modern GPU architectures to expedite computations. nevertheless, while GPUs can accelerate such computations, they do not mitigate the fundamental superpolynomial or exponential growth in complexity associated with increasing dimensionality, and thus can at best offer constant factor speedups.

Another fundamental challenge is that non-convex problems are usually **NP-hard**. For example, the following simple concave quadratic program

$$\begin{aligned} & \underset{x \in \mathbb{R}^n}{\text{maximize}} && \frac{1}{2}x^\top Qx + c^\top x \\ & \text{subject to} && Ax \leq b, \end{aligned}$$

where $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ and Q is a $n \times n$ symmetric negative semi-definite matrix, is **NP-hard** [31]. Even if one relaxes the condition on Q , such that it is a symmetric matrix with only a single negative eigenvalue, the problem remains **NP-hard** [32]. Such non-convex problems do not provide guarantees that any local minimum is a global minimum, which significantly complicates obtaining globally optimal solutions.

In financial modeling, a simple example of an **NP-hard** problem is the two-player zero-sum game (related to other models studied later in this paper)

$$\begin{aligned} & \underset{x \in \mathbb{R}^n}{\text{minimize}} && \max_{\mu \in [\hat{\mu} - \delta, \hat{\mu} + \delta]} \left(R^{(f)} - \mu^\top x \right) - \frac{\gamma}{2} x^\top \Sigma x \\ & \text{subject to} && \mathbf{1}^\top x = 1, \\ & && x \geq 0, \end{aligned}$$

where x is a vector of decision variables, $\mu \in \mathbb{R}^n$ (equivalently, $\hat{\mu}$) is the (estimated or expected) mean return vector of the assets under consideration, $\gamma \in \mathbb{R}_+$ is the so-called risk-aversion coefficient, $\Sigma \in S_+^n$ is the covariance matrix, $\delta \in \mathbb{R}_+$ is a scalar that quantifies the uncertainty or allowable deviation in the mean return vector μ and $R^{(f)}$ is the risk-free return. In this problem, one agent seeks to minimise the maximum potential *regret*, defined as the difference between the best possible outcome and the actual outcome, while an adversarial agent aims to maximise this regret by selecting the least favorable expected returns. The non-convexity arises due to the max-min structure; while the inner maximisation problem is convex (linear in μ), when combined

with the outer minimisation over the binary decision variables, it results in a non-convex problem overall [33].

An extension of this optimisation problem can be obtained by considering other measures of uncertainty, market fluctuations, and the underlying distributions and their parameters. Optimising in the presence of such uncertainties can broadly be formulated as the following optimisation problem:

$$\begin{aligned} & \underset{x \in \mathbb{X}}{\text{minimize}} \quad f(x) + \mathbb{E}[g_0(x, \xi)] \\ & \text{subject to} \quad \mathbb{E}[g_i(x, \xi)] \leq 0, \quad i = 1, \dots, k, \end{aligned} \tag{1.1}$$

where $x \in \mathbb{X} \subseteq \mathbb{R}^n$ is a vector of (possibly discrete) decision variables, $\xi \in \Xi \subseteq \mathbb{R}^m$ a random variable that models the uncertainty of the environment in the problem of interest, $f : \mathbb{X} \rightarrow \mathbb{R}$ a real-valued deterministic function of x and $g_i : \mathbb{X} \times \Xi \rightarrow \mathbb{R}$, $i = 0, 1, \dots, k$ with $k \in \mathbb{N}$ and $k \leq \infty$ is a real-valued stochastic function of x and ξ that captures the uncertainty of the model. Here, $g_0(x, \xi)$ represents the stochastic component in the objective function and $g_i(x, \xi)$, $i = 1, \dots, k$ represent the stochastic components in the constraints which can be independent or correlated. Assuming that the uncertainty $\xi \sim P$, for some known distribution P that is independent of x , then the stochastic term g_0 could be, for example, the value-at-risk (VaR), conditional value-at-risk (CVaR) or some high-order moments of ξ . In practice, these optimisation problems correspond to a wide range of risk measure calculations and portfolio optimisations. For example, “Maslowian Portfolio Theory” argues that a coherent risk measure such as CVaR can be used for the optimisation of investment portfolios, see Refs. [34–36].

The general framework for such optimisation problems falls under the umbrella of stochastic programming (SP) [37, 38]. SP is a well-established approach specifically designed to address optimisation problems under such conditions. Unlike deterministic optimisation, SP is able to take into account inherent uncertainty in financial markets, and thus seeks solutions that are robust for a range of possible scenarios [39]. It turns out that there exists a family of SP problems for optimisation that are particularly amenable to be solved using QC, that of simulation-based optimisation (SBO).

Quantum computing for simulation-based optimisation. SBO, in which stochastic approximations are employed to evaluate functions of random variables within the context of an otherwise deterministic optimisation problem, can be seen as the set of SP problems for which the computation of the terms that involve random variables cannot be performed analytically. The goal is to simulate a sequence of system configurations, each corresponding to specific settings of the decision variables, in order to identify an optimal or near-optimal configuration. This approach allows for the evaluation of a small subset of possible configurations, thereby avoiding the computational infeasibility of exhaustive enumeration [40]. This is therefore useful for families of optimisation problems that involve scenarios where in the objective function – and potentially in the constraints – one encounters quantities that depend on moments of random variables [41, 42]. Reference [43] first considered how quantum amplitude estimation (QAE) and quantum Monte Carlo integration (QMCI) can be used to accelerate SBO over both continuous and discrete variables, introducing a quantum-enhanced SBO algorithm. Concretely, the problem of interest therein is essentially similar to that of (1.1), but assuming no access to the analytical form of $\mathbb{E}[g_i(\xi)]$. The authors of Ref. [43]¹ discriminate between two approaches depending on the nature of the search variable x ; continuous or discrete. In the former case they use modern variations of the QAE algorithm; see Section 3 for a brief exposition of the ones relevant to our work. In the latter, where the search variable x

¹Comparing the notation of [43] to ours the equivalence is $y \rightarrow x$ and $X \rightarrow \xi$. Their formulation also does include deterministic terms on the objective or stochastic terms as constraints.

is discrete, they solve the optimisation part of the problem by means of a parametrised quantum circuit (PQC).

The interest in SBO arises due to its wide range of applications in asset management, including optimal asset allocation, risk-return trade-offs, asset-liability management (e.g., cash flow matching), portfolio optimisation (e.g., robust portfolio construction and risk management), stress testing, risk assessment, and scenario analysis (scenario-based optimisation); see [44–46]. Given the difficulty of the problem, it is only natural to ask whether a practical realisation of quantum-enabled SBO for these applications is possible, and answering this question is the primary motivation for this paper.

Our contribution. Going a step beyond what is considered “usual” in the quantum finance literature, our work aims to investigate how QMCI can assist optimisation problems that involve elements of risk factors; that is, include more a more realistic approach to optimisation problems that appear in financial modelling. To use QMCI for practical applications it is essential to understand all the sources of systematic error that are introduced at each stage of the workflow. This ensures that the final results are interpretable and that the precise bottlenecks for a particular use case are understood. To the best of our knowledge, a thorough analysis of all systematic errors introduced when running QMCI has not previously been carried out – we refer to [47] for previous initial studies in this direction. In addition, for near-term applications of QMCI to optimisation, then given the limitations of current hardware the negative effects of noise will be significant; this can be considered as a significant future “bottleneck” to successfully deploying QMCI for optimisation in the near term. Thus exploring methods of quantum error mitigation that can reduce the impact of noise will be of key importance.

To this end, we explore the strengths and limitations of quantum-enhanced SBO for some well-known financial modeling problems that are often associated with quantum finance, namely the family of “risky” portfolio-optimisation problems. This contributes to the ongoing exploration of how to integrate QC into financial modelling, paving the way for further advancements for applications in portfolio construction. Our main contributions are as follows:

- (1) We identify and define the systematic errors involved in QMCI, which can easily be extended to state preparation of any quantum computation, then we provide detailed analyses of the systematic errors to provide insights into the performance of QMCI. The systematic errors are both related to the distribution that is integrated over and also to the operation of the algorithm that is the integration over a specified region. To this end, we decompose the quantity that is *actually* being estimated into *intermediate* quantities, which are helpful when we want to write down and calculate the general error definitions for a specific function that is integrated.
- (2) We examine the feasibility of using QMCI under resource constraints – particularly limited qubit counts – and apply our systematic error analyses to different use cases revealing further insights into the performance of QMCI and its practical bottlenecks. In particular, we investigate (C)VaR estimations and the related detailed analyses of the specific systematic errors that arise in this calculation; these not only fully explain the observed numerical results – functions of the various bottlenecks that exist for the calculations – but also suggest how these bottlenecks can be overcome. Our key conclusion is that QMCI provides accurate estimations and exhibits the expected quadratic advantage in terms of error scaling as compared to CMCI, but realising this advantage in practice is contingent on suppressing systematic errors to a sufficient degree.
- (3) We investigate practical SBO problems using QMCI, in which we make use of QMCI as a subroutine for the SBO. Given the relative lack of detailed SBO studies that exist in the literature, following [43] we consider two different use cases concerning asset allocation

under uncertainty and provide numerical results for each use case. We start by solving a mean-CVaR portfolio optimisation problem, showcasing the effectiveness of QMCI as compared to classical Monte Carlo integration (CMCI). We then solve a Mean-Var optimisation problem for asset allocation, in which we benchmark results in the presence of device noise using a method of quantum error mitigation that applies to the QAE subroutine of QMCI, demonstrating a more realistic example of how one might actually perform SBO on near-term quantum hardware.² This is also the first time that this method of error mitigation has been applied to a real-world problem that makes use of QMCI as a subroutine. Based on this study we conclude that our method of error mitigation for QAE does improve the expected results when considering estimation of the objective function. However, when considering the full SBO problem – which involves both optimisation and estimation – then we cannot determine conclusively if there is any improvement, given the limited statistics available for the study and the additional complexity of the full SBO problem, where error contributions from sources both related and unrelated to noise are compounded.

It is worth noting that our work in general is significantly differentiated from that of [43]. For example, in terms of portfolio optimisation [43] studied a VaR-based binary optimisation problem for asset allocation, while in our work we focus on the estimation of VaR and CVaR over a number of distributions.

Organisation. In order to provide the right level of context and mathematical tools, we introduce a glossary of financial concepts in Section 2 and we introduce the relevant quantum tools, notably QMCI, used throughout this work in Section 3. Subsequently, we provide a detailed description of the systematic errors involved in the methodology of QMCI in Section 4 followed by (C)VaR estimations in Section 5 as well as the related systematic error analyses. We then investigate two practical SBO problems using QMCI to solve a mean-CVaR portfolio optimisation problem in Section 6 and then to solve a Mean-Var optimisation problem for asset allocation in the presence of device noise in Section 7. We conclude the paper in Section 8.

2. GLOSSARY AND DEFINITIONS OF FINANCIAL CONCEPTS

To bridge the gap between the mathematical finance literature and the quantum computing literature, we present in this section a glossary containing relevant definitions.

We denote the general variables:

- (1) The return of a risk-free asset or portfolio $R^{(f)}$. A risk-free asset or portfolio is a theoretical financial instrument that is assumed to have no risk of financial loss and is expected to return its full value with interest at maturity. When the risk-free return is considered over different periods, we add an subscript $R_T^{(f)}$ to refer to the time period T .
- (2) The return of a portfolio X which represents the gain or loss on the portfolio over a specific period, often expressed as a percentage. This return can be calculated based on the weighted average of the returns of the individual assets within the portfolio, where the weights correspond to the proportion of each asset's value relative to the total value of the portfolio.
- (3) The return of a portfolio admits different decompositions, a possible one is $X = r + R^{(f)}$ where r is the risk premium or excess return of the portfolio - it describes the fluctuations induced by the risk.
- (4) For a portfolio of a single asset, the value/wealth of the asset at time T is denoted as V_T .
- (5) The return of the risky asset during the period T is denoted as X_T .

²Indeed, we can consider the effects of noise as an additional type of error for near-term QMCI calculations.

- (6) Logarithmic returns, also known as log-returns, which measures the percentage change in the value of an asset over a period of time. The log-returns can be calculated by $X_{T+1} = \log\left(\frac{V_{T+1}}{V_T}\right)$.
- (7) For a portfolio of multiple assets, the return of asset i is denoted as X_i .
- (8) The utility function U over which we optimise.

The following probabilistic quantities have the following meaning in the context of this work:

- (9) $\mathbb{E}[X]$ the expectation of the value of the return/wealth of an asset.
- (10) VaR the Value-at-Risk of a portfolio. For a random variable X , $\text{VaR}_\alpha(X) := \inf\{\gamma : \mathbb{P}(X \leq \gamma) \geq \alpha\}$ for the confidence level $\alpha \in (0, 1)$. $\text{VaR}_\alpha(X)$ is a lower percentile of the random variable X for a given loss distribution.
- (11) CVaR is the Conditional-Value-at-Risk which quantifies the expected loss exceeding the Value-at-Risk. For a random variable X , $\text{CVaR}_\alpha(X) := \mathbb{E}[X | X \geq \text{VaR}_\alpha(X)]$ for the confidence level $\alpha \in (0, 1)$.
- (12) $\mathcal{N}(\mu, \sigma^2)$ a Gaussian distribution with mean μ and variance σ^2 with distribution $f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$.
- (13) $\mathcal{LN}(\mu, \sigma^2)$ a log-normal distribution with parameters μ and σ^2 , whose probability density function (PDF) is $f(x) = \frac{1}{x\sigma\sqrt{2\pi}\sigma} e^{-\frac{(\ln x - \mu)^2}{\sigma^2}}$.
- (14) $\mathcal{L}(\mu, c)$ a Lévy distribution with location parameter μ and scale parameter c with distribution $\mathcal{L}(x, \mu, c) = \sqrt{\frac{c}{2\pi}} \frac{e^{-\frac{c}{2(x-\mu)}}}{(x-\mu)^{3/2}}$, for $x \geq \mu$.
- (15) $\Theta(y)$ the indicator function defined as

$$\Theta(y) = \begin{cases} 1, & \text{if } y \geq 0 \\ 0, & \text{otherwise,} \end{cases} \quad (2.1)$$

where y can be either a variable or an expression like $y = x - x_{th}$ with a constant x_{th} . In this sense, $\Theta(y)$ can be equivalently represented as $\Theta(x - x_{th})$ or $\Theta(x \geq x_{th})$ for $y = x - x_{th}$.

In the case of asset allocation under uncertainty:

- (16) $w = \{w_i\}_{i=1}^n$ an allocation weight over n assets. If no leverage is allowed in the allocation (no short position can be taken), then $w_i \geq 0$ for all $i \in [1, n]$ and $\sum_{i=1}^n w_i = 1$.
- (17) The return of a portfolio with a specific allocation $\sum_{i=1}^n w_i X_i$.
- (18) $\lambda \in \mathbb{R}$ a Lagrange parameter describing the level of risk aversion of an investor.

3. QUANTUM ENHANCEMENTS OF MONTE CARLO INTEGRATION

3.1. Monte Carlo integration. The task in MCI is to estimate the value of an integral over a measured space Ω , often a subspace of \mathbb{R}^n . Assuming we are interested in integrals over the real line, the expectation of a function $g(x)$ of a continuous random variable X with the PDF $f_X(x)$ is defined as

$$\mathbb{E}[g(X)] = \int_{-\infty}^{\infty} g(x) f_X(x) dx. \quad (3.1)$$

We refer to $g(x)$ as the “function applied”. On any digital computer, the support of the distribution must be discretised and truncated, and the functions may only take discrete values.³ This means

³This applies to both the classical and quantum cases, but because resource management is more involved in QC, more careful consideration must be taken, and this is discussed in detail in this work.

that both classical and quantum digital computers can only approximately prepare the distribution and compute the function applied. Therefore, MCI on a digital computer estimates $\mathbb{E}[g(X)]_S$, where the subscript S means that the expectation of the function applied is computed for a digitised distribution, and the difference between $\mathbb{E}[g(X)]$ and $\mathbb{E}[g(X)]_S$ is the systematic error introduced by the digitisation, which is discussed in detail in this work. Classically, we sample the random variable \mathcal{S} times, calculate the function $g(\cdot)$ for each sample, and obtain an estimate of the integral by averaging these values. Denoting the estimated value with $\widehat{\mathbb{E}[g(X)]}_S$, we then define the estimation error (which includes statistical and systematic errors) as

$$\epsilon_E(\widehat{g(X)}) = \left| \mathbb{E}[g(X)] - \widehat{\mathbb{E}[g(X)]}_S \right|, \quad (3.2)$$

and the root mean-squared error (RMSE) can be defined as

$$\text{RMSE}(\widehat{g(X)}) = \sqrt{\mathbb{E}\left[\left(g(X) - \widehat{\mathbb{E}[g(X)]}_S\right)^2\right]}. \quad (3.3)$$

Classically, the central limit theorem can be invoked to obtain the asymptotic decay of the RMSE as $\mathcal{O}(1/\sqrt{\mathcal{S}})$, where \mathcal{S} is the number of samples. The quantum counterpart of MCI, i.e. QMCI, provides a quadratic speed up to this convergence, meaning the RMSE decays as $\mathcal{O}(1/\mathcal{S})$.

QMCI is a three step process as described in [19], and consists of preparing a quantum state that has the values of the probability distribution encoded in its amplitudes, applying the observable function $g(\cdot)$ to each state coherently and then measuring the amplitude of an ancilla qubit, providing an estimate of the integral (3.1). Mathematically, the starting point is a circuit, P , acting on n qubits, that prepares a quantum state, $|p\rangle$, encoding the probability distribution $f_X(x)$. That is, $|p\rangle = P|0\rangle_n$ such that

$$|p\rangle = \sum_x \sqrt{f_X(x)} |x\rangle. \quad (3.4)$$

The coherent application of the observable function is provided via a circuit R such that

$$R|p\rangle|0\rangle = \sum_x \sqrt{f_X(x)} |x\rangle \left(\sqrt{1-g(x)} |0\rangle + \sqrt{g(x)} |1\rangle \right). \quad (3.5)$$

In Quantinuum's QMCI engine [19], the circuit R is implemented using the Fourier QMCI approach [8]. Finally, the probability of measuring 1 on the final qubit is $\sum_x g(x)f_X(x)$. Notice that there will be no speedup if we directly measure the ancilla of the state $R|p\rangle|0\rangle$, and the number of samples required will be the same as for classical Monte Carlo methods. QAE is the algorithm used to measure this probability, and it is the core subroutine that provides a quadratic improvement for the rate of convergence of QMCI [19] over CMCI.

3.2. Quantum amplitude estimation. QAE, originally introduced in [6], is the key quantum algorithm used for QMCI. Variations of this algorithm have been proposed over time and we present in this Section a brief and standard presentation of it, for consistency. If we consider an unitary operator \mathcal{A} acting on a register of $(n+1)$ qubits $|0\rangle_n|0\rangle \equiv |0\rangle_{n+1}$ such that the following state is prepared

$$\mathcal{A}|0\rangle_{n+1} = \sqrt{1-a}|\psi_0\rangle_n|0\rangle + \sqrt{a}|\psi_1\rangle_n|1\rangle, \quad (3.6)$$

where the $|\psi_0\rangle_n$ and $|\psi_1\rangle_n$ states are normalised, then QAE can estimate the parameter $a = \sin^2 \theta_a$, $\theta_a \in (0, \pi/2]$ with the convergence of the root mean-squared error (RMSE) of the estimate scaling as $\mathcal{O}(1/\mathcal{S})$, where \mathcal{S} is the number of quantum samples i.e., the total number of uses or calls to the oracle \mathcal{A} . This provides a quadratic advantage as compared to classical Monte Carlo sampling

– the source of quadratic advantage in QMCI as previously discussed. The algorithm makes use of the Grover operator, first introduced in [6]

$$Q = \mathcal{A} \left(\mathbf{1} - 2 |0\rangle_{n+1} \langle 0|_{n+1} \right) \mathcal{A}^\dagger \left(\mathbf{1} - 2 |\psi_0\rangle_n |0\rangle \langle \psi|_n \langle 0| \right), \quad (3.7)$$

where $\mathbf{1}$ denotes the identity operator. The canonical method for QAE relies on quantum phase estimation (QPE) (which relies on the inverse quantum Fourier transform) to estimate the parameter a . This requires significant QC resources – a substantial number of qubits and controlled-not (CNOT) gates. For near-term intermediate-scale quantum (NISQ) devices this is problematic, and several approaches have been developed to overcome this issue (see [9] for further discussion).

One such approach is the maximum-likelihood QAE (MLE-QAE) of [48], refined by [49]. There QPE is no longer required, and is replaced by a classical maximum-likelihood-estimation (MLE) subroutine (see also [50] for another alternative without QPE). One prepares

$$Q^{m_k}(\cos \theta_a |\psi_0\rangle_n |0\rangle + \sin \theta_a |\psi_1\rangle_n |1\rangle) = \cos((2m_k+1)\theta_a) |\psi_0\rangle_n |0\rangle + \sin((2m_k+1)\theta_a) |\psi_1\rangle_n |1\rangle, \quad (3.8)$$

for different $m_k = \{m_0, m_1, \dots, m_{M-1}\}$, for N_k measurements (shots) per m_k . The set $\{(m_k, N_k)\}_k$ is usually referred to as the schedule of QAE, and there are several ways of choosing this schedule; for example one schedule introduced in [48] that has near-optimal quadratic convergence is the so-called exponentially increasing sequence (EIS) $\{(m_k, N_{\text{shots}}) \mid m_k = 2^k, k \in \mathbb{N}\}$, for some fixed number of shots N_{shots} , and this is the schedule that is used throughout this paper. Then letting h_k be the number of times the $|\psi_1\rangle_n |1\rangle$ state is measured – with corresponding probability $\sin^2((2m_k+1)\theta_a)$ – the likelihood function is

$$L_k(h_k, \theta_a) = [\sin^2((2m_k+1)\theta_a)]^{h_k} [\cos^2((2m_k+1)\theta_a)]^{N_k-h_k}. \quad (3.9)$$

The combination of the likelihood functions $L_k(h_k; \theta_a)$ provides a single likelihood function

$$L(h, \theta_a) = \prod_{k=0}^{M-1} L_k(h_k; \theta_a), \quad (3.10)$$

with $h = (h_0, h_1, \dots, h_{M-1})$. The MLE estimate of θ_a is then obtained as

$$\hat{\theta}_a = \arg \max_{\theta_a} \log L(h; \theta_a). \quad (3.11)$$

A slight modification to this approach is to instead set a uniform prior distribution $p(\theta_a)$ across a uniformly-spaced grid of n θ_a values, and then use Bayesian inference with the following update rule to the posterior distribution

$$p(\theta_a) \leftarrow p(\theta_a) L_k(h_k | \theta_a), \quad (3.12)$$

(followed by renormalising) to provide an estimate

$$\hat{\theta}_a = \sum_{i=1}^n \theta_a^i p^i. \quad (3.13)$$

It is such an approach that we make use of in the rest of this paper.

Two additional approaches that do not make use of QPE are also worth mentioning for the purposes of this paper. The approach of iterative QAE of [51] gives a robust theoretical guarantee in terms of convergence. In addition, Ref. [19] introduced the linear-combination-of-unitary-operators QAE (LCU-QAE); specifically, [19, Section 5] found evidence that LCU-QAE performs competitively when considering convergence, while also having robust statistical properties of the estimator – being close to asymptotically unbiased and exhibiting near-Gaussian kurtosis and skewness. QAE still remains an active topic of research and new methods have recently been proposed, for example see [52] for a connection between QAE and classical signal processing.

3.2.1. *Noise-aware quantum amplitude estimation.* During the NISQ era, the prominent characteristic of quantum computation is its susceptibility to noise and errors. A key measure quantifying to what extent noise affects operations, and thus how badly noise can affect a given computation, is the two-qubit gate infidelity, $1 - \varphi_2$, where φ_2 is the two-qubit average gate fidelity, defining the average degree of accuracy between the ideal operation of a two-qubit gate and its actual implementation on real quantum hardware. The infidelity thus quantifies the degree of *deviation* between the ideal operation and its actual implementation.

Current NISQ devices have high error rates (i.e., large gate infidelities). To address the challenge of noise affecting computations, quantum error correction protocols have been proposed. However, implementing fully-fledged quantum error-correction (QEC) codes requires a considerable overhead in qubit resources and computational complexity, something not currently feasible on NISQ-era devices due to their limited qubit count and high error rates.

Error mitigation techniques thus have a role in improving the reliability of quantum computations without necessitating full QEC. Error mitigation strategies aim at reducing the impact of noise and errors on the outcomes of quantum algorithms run on noisy devices. Unlike QEC, which demands significant overhead and resources, error mitigation techniques offer intermediate solutions that are more feasible for current NISQ-era devices. A well known example is zero-noise extrapolation [53], which relies on post-processing the outcomes of a noisy computation in order to mitigate the impact of the noise on the final results.

Reference [54] proposes a means of error mitigation that can be applied to any QAE algorithm that uses repeated shots of the same circuit to infer the amplitude, such as the MLE-QAE introduced in the previous section (we will now only consider MLE-QAE in this example). The approach is termed noise-aware QAE (NA-QAE), and involves directly translating the noise at the physical level to the resultant estimation uncertainty at the application level, such that the QAE algorithm is “noise aware”.

In this approach the noise introduced by applying the Grover operator Q in QAE is modelled as Gaussian noise affecting the measured qubit, parameterised by two constants k_σ and k_μ , which encode the properties of the noise, such that after m , applications of the Grover operator the following state is prepared

$$Q^{m_k} (\cos \theta_a |\psi_0\rangle_n |0\rangle + \sin \theta_a |\psi_1\rangle_n |1\rangle) = \cos((2m_k + 1)\theta_a + \theta_a^\epsilon) |\psi_0\rangle_n |0\rangle + \sin((2m_k + 1)\theta_a + \theta_a^\epsilon) |\psi_1\rangle_n |1\rangle, \quad (3.14)$$

where θ_a corresponds to the desired rotation corresponding to the unitary enacted by Q and $\theta_\epsilon \sim \mathcal{N}(k_\mu m_k, k_\sigma m_k)$ is an additional erroneous component to the rotation arising from the noise. The probability of measuring the circuit in the $|1\rangle$ state is then modified to $e^{-2k_\sigma m_k} \sin^2((2m_k + 1)\theta_a + \theta_\epsilon)$ and the likelihood modified accordingly.

If an assumption is made that $k_\mu = 0$ (by previous calibration of the hardware for example), then the error introduced by the noise only arises from the “variance” component proportional to k_σ . This component can be suppressed by increasing the number of shots run. Thus NA-QAE proposes a methodology for the factor increase in the number of shots required to obtain the same estimation worst-case variance for the estimate of the amplitude as one would have achieved in the noiseless case when running N_k shots

$$N_k^{\text{na-qae}} = (4k_\sigma m_k + 1)N_k. \quad (3.15)$$

Of course, implementing this update rule requires knowledge of the parameter k_σ , and this may be possible to calculate from device characteristics alone, under certain assumptions. If the assumption $k_\mu = 0$ is made, then the Gaussian noise model is equivalent to depolarising noise affecting a single qubit. In this case one can equate $k_\sigma = -\ln(\tilde{p}_{\text{coh}})/2$, where \tilde{p}_{coh} is the probability that the state remains coherent after each application of Q under a depolarising noise model. Then

(ignoring 1-qubit gates and other sources of error such as memory errors and state preparation and measurement (SPAM) errors) it can be shown that the coherence probability per application of Q can be approximately expressed as [55]

$$\tilde{p}_{\text{coh}} = [1 - 2(1 - \varphi_2)]^{n_2}, \quad (3.16)$$

where n_2 is the total number of two-qubit gates per application of Q . This then gives a method to estimate k_σ from the device characteristics. However, in the approach of MLE-QAE, all that is required is that a prior distribution is placed on k_σ in order to construct a joint prior distribution $p(\theta_a, k_\sigma)$, and then as the algorithm progresses by running batches of circuits for different m_k values, k_σ can be continuously inferred and updated. In this way, the algorithm is adaptive because, as the noise profile is updated based on the data for a given batch of circuits, the corresponding factor increase in the number of shots to run for the next batch of circuits is then inferred, and so on.

Because the performance of QAE is key to the performance of QMCI, techniques such as NA-QAE will be very useful in the NISQ era when it comes to being able to run real-world problems on quantum hardware. Later on in Section 7, we will demonstrate the first application of NA-QAE for performing QMCI for a problem of real-world interest, based on a simplified SBO problem for portfolio optimisation for asset allocation under uncertainty.

3.3. Quantinuum’s quantum Monte Carlo integration engine. In this work we use the end-to-end engine developed and originally introduced in Ref. [19] to investigate practical applications of QMCI. This section provides a high level overview of the engine and we refer to Ref. [19] for further detail. The engine was designed to provide a modular tool to perform QMCI calculations. Its modularity comes from its division into multiple modules, represented in Ref. [19, Fig. 1], which can be easily updated or replaced, independently of the other modules. There are two notable advantages to this approach: (1) a new bespoke method for e.g., state preparation can be studied and rapidly prototyped and benchmarked against other methods, and (2) given continuous improvements in the field for QAE methods, noise mitigation or state preparation, we can continuously revisit previously investigated use-cases and see how new methods impact their solution. These advantages are further interesting as the QMCI engine is endowed with a resource quantification mode which accounts for all resources (including qubit and gate counts) required to produce a certain calculation up to a given precision. Hence, the modularity allows us to determine the best method to approach a particular QMCI calculation. The engine provides a library of state-preparation methods, functions applied and QAE methods to facilitate the investigation of applications for which QMCI is either a subroutine (such as in a hybrid classical-quantum optimisation problem) or the expected output (as in the pricing of a financial instrument).

3.3.1. State preparation with parameterised quantum circuits. A critical component of the QMCI engine is its library of state-preparation methods that encode probability distributions in the amplitudes of qubit states. As more applications will make use of QMCI over time, the range of probability distributions of interest will also increase. For this reason, the library of methods to load probability distributions into quantum states should be as general as possible, both in terms of the types of distributions and the quantum routines used. There are several approaches to perform state preparation, we refer to Refs. [56–63] and references therein for a broad summary of the state of the art.

As regards the QMCI engine, Ref. [19] describes various distribution-loading techniques, including efficient circuits to load Gaussian and log-normal distributions obtained by training PQCs. Ref. [19] considers two types of ansatz for PQC training: (i) Hardware-efficient (HWE) ansatz and (ii) Tensor network (TN) methods with brickwall ansatz, and we use both methods in this work.

In the next Section, we proceed to a full audit of the errors introduced by the distribution loading process. In particular, we demonstrate that it is key to understand all state-preparation errors that

arise. The state-preparation (or equivalently distribution-loading) step of QMCI is still the subject of active research in the quantum community, as it is currently the bottleneck for extending to higher-dimensional problems or unstructured distributions. Recently, there have been numerous attempts to tackle this problem using more quantum-native approaches such as solving partial differential equations [19], determining the ground state of supersymmetric Hamiltonians [64, 65], or using the techniques of linear combination of unitaries [63]. We expect that any improvement as regards state-preparation methods will greatly expand the applicability of QMCI.

4. SYSTEMATIC ERRORS IN QUANTUM MONTE CARLO INTEGRATION

In this section, we define the fundamental systematic errors that arise when we want to estimate the value of an integral using QMCI. Our methodology here is to first define each error in terms of its source at a high level, and then define a measure that quantifies it in the context of QMCI. Different errors with their associated sources are depicted in Fig. 1. It is important to note that some of these errors occur both in quantum and classical MCI. Also, the errors are not necessarily independent of each other; indeed, the majority of the errors *are* dependent, and while it may be possible to define errors such that they are independent, our measures are designed in order to aid practical use, leading to intuitive measures that are interpretable and thus can provide further insights into results of QMCI. In each error definition, we decompose the quantity that is *actually* being estimated into *intermediate* quantities that help us to intuitively understand and interpret the errors. The *intermediate* quantities are helpful when we want to write down and calculate the general error definitions for a specific function applied. In Sections 5 and Appendix C, we start from an *intermediate* quantity to both calculate the errors for a specific function applied and write an error tailored for the specific application, demonstrating both the importance of the *intermediate* quantities and the adaptability of our approach to practical requirements of different use cases.

4.1. Truncation error. The first error occurs when we truncate the limits of the estimated integral. For example, one or both integral limits may not be finite (as in (3.1)), and truncation of the integral limits to a finite range leads to this error. If we restrict the limits of the integral in (3.1) to the range $[a, b]$ (and if $f_X(x) \neq 0$ for any $x \notin [a, b]$), then this error is defined as

$$\epsilon_{TE}(g(X)) = \left| \int_{-\infty}^{\infty} g(x)f_X(x) dx - \int_a^b g(x)f_X(x) dx \right|. \quad (4.1)$$

Now, the intermediate quantity that we say we want to estimate is the following integral

$$\mathbb{E}[g(X)]_T := \int_a^b g(x)f_X(x) dx. \quad (4.2)$$

The truncation error is represented by the green filled areas at the tails of the distribution in Fig. 1.

4.2. Discretisation error. In MCI, we obtain an estimate of the integral by sampling the random variable X . However, the sample resolution might be limited by the finite number of bits used, leading to a discretisation error. In other words, this error arises when we can only sample the continuous random variable discretely due to the finite number of bits/qubits that encode the random variable. On a classical computer, we have access to (effectively) an unlimited number of bits and can easily suppress this error to reach a desired accuracy in the estimate. However, the same cannot be said for a quantum computer, and this error plays a crucial role in QMCI due to the limited number of qubits used to represent the random variable.

For the integral in (3.1), we define the discretisation error as

$$\epsilon_{DE,n}(g(X)) = \left| \int_a^b g(x)f_X(x) dx - \sum_{i=0}^{N-1} g(x_i)f_X(x_i)\Delta \right|, \quad (4.3)$$

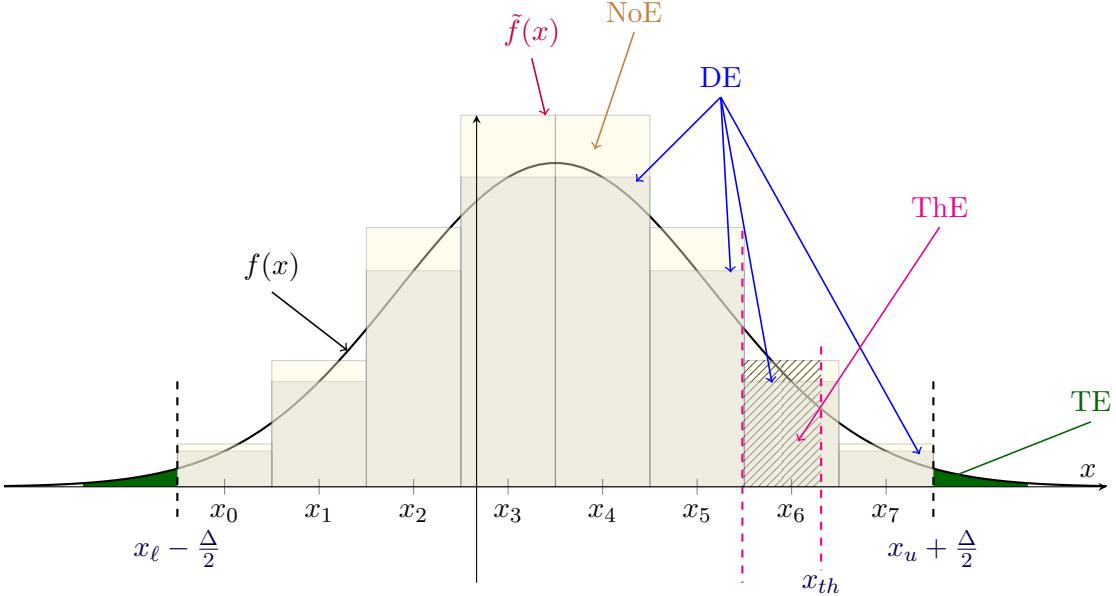


Figure 1. Illustration of the systematic errors in QMCI for a probability density $f_X(x)$ over the domain $x \in [-\infty, \infty]$, which includes four components: truncation error (TE), discretisation error (DE), normalisation error (NoE) and thresholding error (ThE). TE arises when $f(x)$ is prepared in the limited range $[x_\ell - \Delta/2, x_u + \Delta/2]$, which corresponds to the green area. DE arises from the discretisation of $f(x)$ over $N = 2^n$ supports using n qubits over the range, which is the difference between the area covered by $f(x)$ and the area covered by the grey rectangles under the $f(x)$ curve. As illustrated by the yellow area, NoE arises from the renormalisation of the truncated and discretised function $\tilde{f}(x)$, where $\tilde{f}(x)$ represents the renormalised $f(x)$. ThE is not an intrinsic error associated with the representation of the probability distribution; but rather a further “downstream” error that will be incurred if a threshold is taken at a value that doesn’t exactly align with the histogram interval edge points.

where $x_0 := a + \frac{\Delta}{2}$, $x_i := x_0 + i \times \Delta$ (such that $x_{N-1} := b - \Delta/2$), $\Delta := x_{i+1} - x_i = \frac{b-a}{N}$, and $N = 2^n$ for n qubits. With these definitions, the discrete sum

$$\mathbb{E}[g(X)]_{D,n} := \sum_{i=0}^{N-1} g(x_i) f_X(x_i) \Delta, \quad (4.4)$$

is the next intermediate quantity, which is the mid-point Riemann sum approximation of the integral in (4.2).⁴ The subscript D here refers to “discretisation”. In the QMCI engine and in the rest of this work, we denote $x_\ell = x_0$ and $x_u = x_{N-1}$. Discretisation of the distribution is represented by the gray rectangular areas in Fig. 1.

4.3. Normalisation error. This error occurs because we need to load the truncated and discretised distribution into a quantum state. The mass of a probability distribution function (on its

⁴The mid-point Riemann sum is one of the most common techniques to approximate a definite integral by a finite sum using rectangular regions due to its simplicity and accuracy. The choice of mid-point Riemann sum for the discretisation is unrelated to the QMCI engine, which is agnostic to this choice, and a different choice may be used if it is more suitable for a given function applied.

domain) or the sum of all the points of a probability mass function are equal to 1. However, this is altered after truncation and discretisation. From Eq. (4.4), we can thus define

$$\tilde{f}_X(x_i) = \frac{f_X(x_i)\Delta}{\sum_{i=0}^{N-1} f_X(x_i)\Delta}, \quad (4.5)$$

which is illustrated by the yellow rectangular areas in Fig. 1, and $\sum_{i=0}^{N-1} f_X(x_i)\Delta$ is not necessarily equal to 1. Since the probability distribution loaded by a quantum state is always normalised (i.e. the squared amplitudes for a quantum state sum to 1), we have to use the normalised distribution $\tilde{f}_X(x_i)$. Then, the normalisation error is defined as

$$\epsilon_{\text{NoE},n}(g(X)) = \left| \sum_{i=0}^{N-1} g(x_i)f_X(x_i)\Delta - \sum_{i=0}^{N-1} g(x_i)\tilde{f}_X(x_i) \right|, \quad (4.6)$$

$$= \left| \sum_{i=0}^{N-1} g(x_i) (f_X(x_i)\Delta - \tilde{f}_X(x_i)) \right|. \quad (4.7)$$

This means that our next intermediate quantity is

$$\mathbb{E}[g(X)]_{\text{No}} := \sum_{i=0}^{N-1} g(x_i)\tilde{f}_X(x_i). \quad (4.8)$$

where the subscript “No” refers to “normalisation”.

4.4. State-preparation error. The state preparation error introduced in this section is distinct from the aforementioned errors (which are all unavoidable when working on a digital computer). This error arises from the difference between a quantum state that could, in principle, be prepared, and the quantum state that is actually being prepared. Unlike in classical computing, where the values of probability mass must take some discrete value, amplitude encoding for a quantum state allows in principle a continuous range of values constrained to the overall normalisation.

We want to load the normalised probabilities $\tilde{f}_X(x_i)$ into a quantum state as $\sum_{i=0}^{N-1} \sqrt{\tilde{f}_X(x_i)}|x_i\rangle$, where $|x_i\rangle$ are the computational basis states and $N = 2^n$ for n qubits.⁵ However, depending on the state-preparation method, this state cannot be loaded exactly; rather it will be loaded approximately. Denoting the prepared state as

$$|\psi\rangle = \sum_{i=0}^{N-1} \sqrt{p(x_i)}|x_i\rangle, \quad (4.9)$$

we then define the state-preparation error as

$$\epsilon_{SE}(g(X)) = \left| \sum_{i=0}^{N-1} g(x_i)\tilde{f}_X(x_i) - \sum_{i=0}^{N-1} g(x_i)p(x_i) \right|, \quad (4.10)$$

$$= \left| \sum_{i=0}^{N-1} g(x_i) (\tilde{f}_X(x_i) - p(x_i)) \right|. \quad (4.11)$$

Finally, we define the quantity that we *actually* estimate in QMCI as

$$\mathbb{E}[g(X)]_S := \sum_{i=0}^{N-1} g(x_i)p(x_i), \quad (4.12)$$

⁵Note that *any* normalised state can in principle be prepared - but this may require an arbitrarily deep circuit.

where S stands for “state”, meaning that the expectation of the function applied is computed for the discrete distribution loaded by the quantum state. Evaluating all errors along the way

$$\mathbb{E}[g(X)] \rightarrow \mathbb{E}[g(X)]_T \rightarrow \mathbb{E}[g(X)]_{D,n} \rightarrow \mathbb{E}[g(X)]_{No} \rightarrow \mathbb{E}[g(X)]_S, \quad (4.13)$$

leads to an upper bound on the total systematic error, as the sum of all aforementioned errors:

$$|\mathbb{E}[g(X)] - \mathbb{E}[g(X)]_S| \leq \epsilon_{SE}(g(X)) + \epsilon_{NE,n}(g(X)) + \epsilon_{DE,n}(g(X)) + \epsilon_{TE}(g(X)). \quad (4.14)$$

Adding the errors in this manner provides a loose upper bound on the discrepancy between $\mathbb{E}[g(X)]$ and $\mathbb{E}[g(X)]_S$, and understanding how each of these influences the final computation is of practical relevance for applications.

4.5. Operational errors. Until now, all errors studied apply for any arbitrary function applied $g(\cdot)$. We introduce here the notion of operational errors associated with how the imperfect probability distribution propagates when functions are applied thereto. This is not something that can be done completely independently of the actual functions that are being applied, and thus this needs to be part of the analysis for a particular calculation.

The QMCI engine provides operations to be applied to the random variables represented in a quantum circuit. We call such operations *circuit enhancement operations*. When we apply certain enhancement operations, these operations might introduce additional errors. In the QMCI engine, there are many enhancement operations (most of which are binary), and we here consider the thresholding operation, which we use for VaR and CVaR calculations in this paper. Let us say we want to estimate $\mathbb{E}[X \Theta(X \geq V_{Th})]$. In the QMCI engine, we achieve this by defining a threshold with an inclusive upper bound of V_{Th} . However, if $V_{Th} \in (x_i, x_{i+1})$ (open interval), then the engine uses x_i as the bound of the threshold. This gives rise to the thresholding error defined as

$$\epsilon_{ThE}(X \Theta(X \geq V_{Th})) = |\mathbb{E}[X \Theta(X \geq V_{Th})] - \mathbb{E}[X \Theta(X \geq x_i)]|. \quad (4.15)$$

As mentioned previously, we use the thresholding operation for the VaR and CVaR estimations in Section 5, and we explain the accuracy of the estimates based on the interplay between the discretisation and thresholding errors. Therefore, in Section 5, we use an error that combines discretisation and thresholding errors into a single measure, denoted as $\epsilon_{DThE,n}$. Having introduced all the systematic errors, in the next section, we apply our systematic error analysis to different use cases revealing insights into the performance of QMCI and its bottlenecks.

5. ESTIMATING (CONDITIONAL-)VALUE-AT-RISK USING THE QUANTUM MONTE CARLO INTEGRATION ENGINE

In this section, we use the QMCI engine to estimate VaR_α and CVaR_α , and we benchmark the results against both the analytical values and classical estimations. Using the QMCI engine’s in-built state-preparation library we load a 5-qubit Gaussian distribution, and then use this circuit to estimate VaR_α and CVaR_α . Performing QMCI using this circuit in general yields accurate estimates of the mean and second moment [19]. However, we demonstrate that VaR_α can only be estimated accurately for small α values or for specific configurations (“sweet spots”) where systematic errors are minimised. Similarly, CVaR_α estimates are only accurate at these specific “sweet spots”. Consequently, we conclude that encoding a Gaussian distribution using only 5 qubits is insufficient for running SBO where α varies. Based on our systematic error analysis, we demonstrate that the primary source of error in both VaR_α and CVaR_α estimations is the thresholding error, which is related to the discretisation error. Then, we determine the minimum number of qubits required to suppress these errors to a sufficient degree. In other words, we show that, given a sufficient number of qubits (and assuming accurate state-preparation techniques when

using a larger number of qubits), QMCI provides accurate estimations and exhibits the expected quadratic advantage in terms of RMSE scaling as compared to CMCI.

5.1. Value-at-Risk estimations. Risk management constitutes a fundamental element within the framework of volatile economic environments. It encompasses the identification and analytical assessment of risk indicators, aimed at appraising and mitigating their detrimental effects. This process is important in ensuring the stability and resilience of financial and economic systems against uncertainties inherent in market dynamics.

The significance of VaR and CVaR as risk measures cannot be overstated, forming the foundation of the Basel III regulatory agreement [66] (requiring credit granting institutions such as banks to perform stress tests using VaR-related measures). The computation of these risk measures, therefore, is of paramount importance, and practitioners utilise Monte Carlo simulations in order to determine the VaR and CVaR of a given portfolio [67]. This process typically involves constructing a model that represents the portfolio's assets, and then calculating the aggregate value of the portfolio across N distinct scenarios based on variations in the model's input parameters. VaR_α is attractive because it is a conceptually simple and intuitive quantification of risk. More formally, we may represent the gains (loss if negative) of an investment portfolio over a fixed time as a random variable X . Given a probability level $0 < \alpha < 1$, VaR_α of the random variable X is given by

$$\text{VaR}_\alpha(X) = \inf\{\gamma : \mathbb{P}(X \leq \gamma) \geq \alpha\}. \quad (5.1)$$

VaR is equivalent to the percentile for a probability distribution. We see that when the loss distribution is continuous, $\text{VaR}_\alpha(X)$ is the loss such that

$$\mathbb{P}(X \leq \text{VaR}_\alpha(X)) = \mathbb{E}[\Theta(\text{VaR}_\alpha(X) - X)] = \alpha, \quad (5.2)$$

where $\Theta(\cdot)$ is the indicator function as defined in (2.1). VaR_α is interpreted as the threshold value below which the probability of the portfolio or the investment losses exceeding the value is α or less.

Algorithm 1: The bisection algorithm for estimating $\text{VaR}_\alpha(X)$

```

1 Start with two values  $a$  and  $b$  such that  $g(a) > \alpha$  and  $g(b) < \alpha$ ;
2 while  $|b - a| \geq \epsilon$  do
3   Estimate  $g(c)$ , where  $c = \frac{a+b}{2}$  is the midpoint of the interval  $(a, b)$ ;
4   if  $g(c) = \alpha$  then
5     | Stop and return  $\text{VaR}_\alpha(X) = c$ ;
6   else if  $g(c) < \alpha$  then
7     | Set  $b = c$ ;
8   else
9     | Set  $a = c$ ;
10  end
11 end
12 Return  $\text{VaR}_\alpha(X) = \frac{a+b}{2}$ ;

```

Despite the widespread use of VaR_α , it lacks many important features that are crucial for risk management. Specifically, VaR_α does not look beyond the α -percentile; that is, VaR_α does not account for losses exceeding it, and hence it can provide an inadequate picture of risks by failing to capture so-called “tail risks”. Another challenge with VaR_α is that its optimisation becomes challenging when derived from scenario-based calculations (as is the case for the studies discussed in this work). Specifically, it has been shown [68,69] that VaR_α , as a function of portfolio allocations, can possess several local extrema; this significantly increases the difficulty in identifying the most

advantageous portfolio composition, or for accurately computing the VaR_α for a given portfolio mix.

Instead, CVaR has become the modern standard risk measure for a variety of practical applications, since CVaR_α is recognized as a suitable model of risk in volatile economic circumstances. It is widely used for modeling and optimising hedge and credit risks [70]. Furthermore, in terms of regulation, both in the insurance domain with the Solvency II Directive [71] and in the financial domain with Basel III Rules, the role, importance, and applicability of CVaR_α cannot be overstated.

While VaR_α gives a threshold level of loss with a specified confidence level, CVaR_α goes a step further by quantifying the expected loss exceeding VaR_α . Practitioners are interested in CVaR_α because it is easily expressible as an optimisation problem [35, 72, 73]. It is also continuous in α and convex (concave if seen from the insurance perspective) in the variables. The definition of CVaR_α , as an expectation, is given by

$$\begin{aligned}\text{CVaR}_\alpha(X) &= \mathbb{E}[X|X \geq \text{VaR}_\alpha(X)], \\ &= \frac{1}{1-\alpha} \int_{-\infty}^{+\infty} \Theta(x - \text{VaR}_\alpha(X)) x f(x) dx.\end{aligned}\tag{5.3}$$

This definition can be extended to the discrete case [72], where CVaR_α is the weighted average of VaR_α and the expected losses strictly exceeding VaR_α .

While the preferred risk measure is CVaR_α , a reader might assume that VaR_α is not relevant. However, this is not the case since an interesting example of portfolio optimisation under uncertainty is one which calculates VaR_α and optimizes CVaR_α simultaneously [73]. Interestingly, this SBO is one of the better candidates for a quantum solution, where one might hope to obtain maximal benefits by exploiting a quadratic speedup using QMCI.

5.2. Estimation of VaR_α and CVaR_α using Monte Carlo integration. We now describe how to estimate VaR_α and CVaR_α using CMCI. Estimation of $\text{VaR}_\alpha(X)$ can be expressed as a root-finding problem i.e., $g(\gamma) = \mathbb{E}[\Theta(\gamma - X)]$, with $\gamma = \text{VaR}_\alpha(X)$. We can use a simple bisection method to find the root of $g(\gamma) = \alpha$, as outlined in Algorithm 1. The expectation value in the definition of the function $g(\gamma)$ is estimated using MCI (CMCI or QMCI). Suppose that the random variable X follows a distribution $f_X(x)$. Classically, we sample \mathcal{S} random numbers from $f_X(x)$, denoted by the sorted observations $\{\tilde{X}_1, \dots, \tilde{X}_{\mathcal{S}}\}$ such that $\tilde{X}_1 \leq \dots \leq \tilde{X}_{\mathcal{S}}$. Then, an estimate of $g(\gamma)$ can be obtained

$$\widehat{g(\gamma)} := \frac{I_\alpha}{\mathcal{S}},\tag{5.4}$$

where $I_\alpha = \lfloor \alpha \mathcal{S} \rfloor$ denotes the index of the I_α -th observation in \tilde{X} . Next, we turn to the estimation of $\text{CVaR}_\alpha(X)$. Following from (5.3), we can estimate $\text{CVaR}_\alpha(X)$ as

$$\widehat{\text{CVaR}}_\alpha(X) := \frac{\tilde{X}_{I_\alpha+1} + \dots + \tilde{X}_{\mathcal{S}}}{\mathcal{S} - I_\alpha}.\tag{5.5}$$

parameter	value / choice
X	$\mathcal{N}(\mu, \sigma^2)$
μ	0.10
σ	0.05

TABLE 1. The parameters of the Gaussian distribution used for the VaR_α and CVaR_α estimations.

5.3. Numerical Experiments. We use a Gaussian distribution to model the underlying random variable X (due to its known analytical solutions) to benchmark the performance of QMCI and

CMCI, not only against one another, but also with respect to the closed-form solutions to the problems. This approach allows us to make precise comparisons and assess the reliability of using QMCI in practical scenarios, ensuring that any observed discrepancies or improvements in the results obtained using QMCI can be accurately attributed to the method itself, rather than to characteristics of the distribution. These numerical analyses reveal that QMCI can provide similar estimates in terms of RMSE as CMCI for certain α values, however the numerical results also demonstrate that for particular α values QMCI gives large errors that lead to inaccurate estimates. In addition, based on CVaR $_{\alpha}$ estimation, we also showcase the expected quadratic advantage of QMCI over CMCI for those particular configurations where errors are sufficiently suppressed. This section presents the results and describes how they were obtained, leaving the analysis and explanation of the observed results for Section 5.4.

Before presenting the results, we describe the distribution and its parameters, and provide analytical expressions for the quantities estimated using CMCI and QMCI. For the analyses in this section, we use a one-dimensional Gaussian random variable $X \sim \mathcal{N}(\mu, \sigma^2)$ with the parameters specified in Table 1. For the bisection method used to determine VaR $_{\alpha}$ (as described in Algorithm 1), $g(\gamma)$ can be evaluated analytically as follows:

$$g(\gamma) = \mathbb{E}[\Theta(\gamma - X)] = \int_{-\infty}^{\gamma} dx \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma} = \frac{1}{2} \left(1 + \text{erf}\left(\frac{\gamma - \mu}{\sqrt{2}\sigma}\right) \right). \quad (5.6)$$

Given the value of VaR $_{\alpha}$, the CVaR $_{\alpha}(X)$ in (5.3) can similarly be analytically calculated for a Gaussian distribution. We omit this calculation for brevity.

	Analytical value	CMCI	QMCI
VaR $_{0.50}$	0.100	0.101	0.112
VaR $_{\sim 0.69}$	0.125	0.126	0.132
VaR $_{\sim 0.83}$	0.148	0.149	0.148
VaR $_{\sim 0.84}$	0.150	0.150	0.165

TABLE 2. VaR $_{\alpha}$ estimations at the last iteration of the bisection method shown in Fig. 2 for QMCI and CMCI with 10,000 samples from the Gaussian distribution (with the parameters in Table 1). For two α values, the VaR $_{\alpha}$ estimates obtained using QMCI (marked with red) are far away from their analytical values. For the other two α values, QMCI and CMCI give similar VaR $_{\alpha}$ estimates. These behaviours are explained by the error analysis described in Section 5.4.

5.3.1. *Estimation of VaR $_{\alpha}$ using the bisection method.* We run Algorithm 1 with the stopping criterion $\epsilon = 1 \times 10^{-4}$ and use QMCI or CMCI as a sub-routine to estimate $g(c)$. Note that the stopping criterion is independent of the MCI sub-routine, and we compare the estimates at the last step where Algorithm 1 hits the stopping criterion. In our analysis, we set the initial values of a and b to $a = \mu + 3\sigma$ and $b = \mu - 3\sigma$, satisfying $g(a) > 0$ and $g(b) < 0$. As in the rest of the paper, we use an implementation of the MLE-QAE algorithm for QMCI and load the Gaussian distribution using a 5-qubit circuit (i.e., creating 32 support points for the discretisation). For the QMCI estimation of $\mathbb{E}[\Theta(c - X)]$ in Algorithm 1, we use the thresholding functionality of the QMCI engine.

Figure 2 presents a comparative analysis of the estimation errors for both QMCI and CMCI using 10000 samples (in each step of each run with different α). The absolute difference between the analytical value (α) and the estimated value ($\widehat{g(\gamma)}$) is plotted over multiple steps of the loop of Algorithm 1 for four different α values: 0.5, 0.83, 0.69, and 0.84. Each subplot illustrates how the

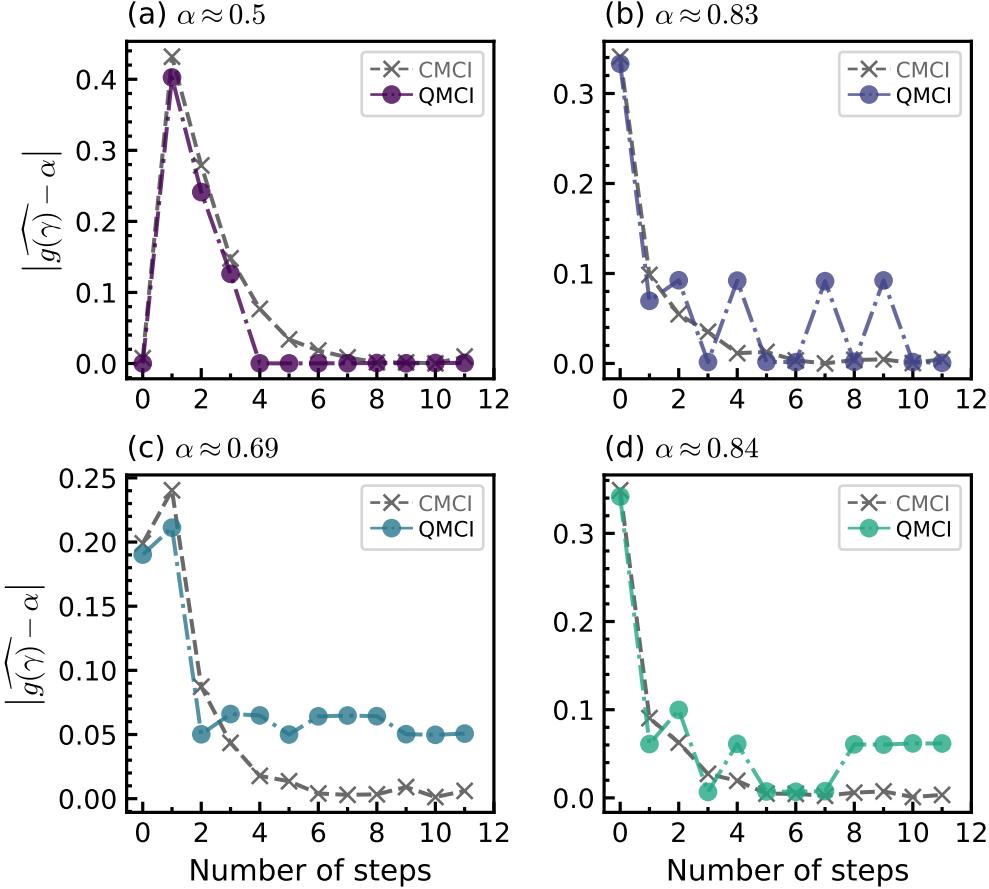


Figure 2. Difference between the analytical value of α and the estimate $\widehat{g(\gamma)}$ obtained from QMCI or CMCI at each iteration of the loop of Algorithm 1 presented for four α values. In each iteration of the loop, $\widehat{g(\gamma)}$ are obtained using 10,000 samples for both QMCI and CMCI. For (a)–(b), QMCI shows similar trends to CMCI, and both converge to similar values and have small errors at the last step, where Algorithm 1 reaches its stopping criterion. For (c)–(d), QMCI does not get close to the analytical value of α and has large errors at the last step. The VaR_α estimations at the last iteration and the corresponding analytical values are given in Table 2.

estimation error evolves as the number of steps increases, offering insights into the performance of both integration methods.

In Fig. 2 (a)–(b), where $\alpha \approx 0.5$ and $\alpha \approx 0.83$, respectively, QMCI shows similar trends to CMCI, and both converge to similar values and have small errors at the last step. However, in Fig. 2 (c)–(d), with $\alpha \approx 0.69$ and $\alpha \approx 0.84$, respectively, QMCI does not get close to the analytical value of α and has large errors at the last step. The VaR_α estimations at the last iteration and the corresponding analytical values are given in Table 2. For two α values (corresponding to (c)–(d) in Fig. 2), the VaR_α estimates obtained using QMCI (marked with red) are not close to the analytical values. For the other two α values, QMCI and CMCI give similar VaR_α estimates.

Overall, the results indicate that while QMCI has the potential to provide accurate estimations for these quantities, this is highly dependent on the specific conditions – specifically the value of α . These behaviours are explained by the error analysis in Section 5.4.

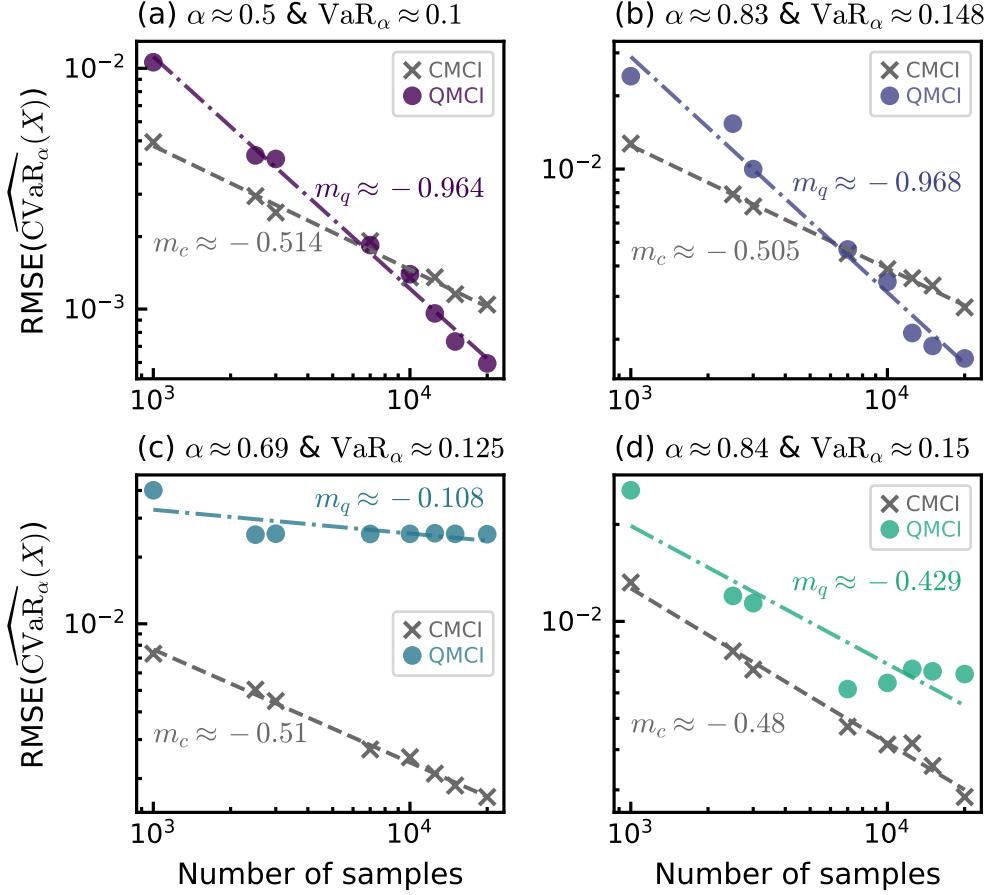


Figure 3. Comparisons of the RMSE scaling (as a function of number of samples \mathcal{S}) in CVaR_α estimations using QMCI and CMCI for four α values. RMSE estimations are obtained by averaging over 250 independent runs. For (a)–(b), QMCI clearly performs better than CMCI, meaning $m_q \sim -1$ and $m_c \sim -0.5$, thus QMCI gives a better RMSE than CMCI for a sufficiently large number of samples. For (c), QMCI scales much worse than CMCI, while QMCI scaling is similar to CMCI for (d).

5.3.2. Scaling of the root mean-squared error for CVaR_α estimation. In Section 5.3.1 we showed that (albeit only for certain α values) QMCI can produce accurate VaR_α estimates. However we did not analyse the convergence properties. In this section, based on CVaR_α estimations, we investigate whether QMCI exhibits a quadratic advantage in RMSE scaling over CMCI, and we estimate CVaR_α using QMCI and CMCI across different sample sizes and estimate their RMSE from 250 independent runs of CVaR_α estimations. We show that QMCI has a clear quadratic advantage for the same α values for which QMCI gave accurate VaR_α estimations. We explain these observations based on the error analysis in Section 5.4.

Figure 3 presents comparisons of the RMSE scaling (as a function of number of samples) in CVaR_α estimations using QMCI and CMCI for four α values. Representing the number of samples with \mathcal{S} , we hereby denote the RMSE scaling as $\mathcal{O}(\mathcal{S}^{m_q})$ and $\mathcal{O}(\mathcal{S}^{m_c})$ for QMCI and CMCI, respectively. In Fig. 3, we fit the RMSE estimations to obtain estimates of m_q and m_c . In Fig. 3 (a)–(b), QMCI clearly outperforms CMCI, meaning the the RMSE scales as $m_q \sim -1$ for QMCI, while $m_c \sim -0.5$ for CMCI. Thus, in Fig. 3 (a)–(b), QMCI has better RMSE scaling than CMCI for a

	RMSE for CMCI	RMSE for QMCI
CVaR _{0.50}	0.0010	0.0006
CVaR _{~0.69}	0.0016	0.0254
CVaR _{~0.83}	0.0027	0.0017
CVaR _{~0.84}	0.0028	0.0069

TABLE 3. RMSE of the CVaR $_{\alpha}$ estimations using QMCI and CMCI with 20000 samples (for 250 independent runs with the Gaussian distribution in Table 1). For $\alpha \sim 0.69$, the RMSE of QMCI (marked with red) is about an order of magnitude larger than CMCI. For $\alpha \sim 0.84$, the RMSE of QMCI (marked with blue) is larger than CMCI (but approximately the same order of magnitude). For the other two α values, QMCI has a smaller RMSE than CMCI.

sufficiently large number of samples. In Fig. 3 (c), the RMSE for QMCI is almost constant as a function of the number of samples, with much worse scaling than for CMCI (which has $m_c \sim -0.5$ for all cases). Finally, the QMCI RMSE scaling is similar to that of CMCI in Fig. 3 (d). The RMSE values for $S = 20000$ are given in Table 3.

In summary, Fig. 3 reveals that QMCI achieves a clear quadratic speedup over CMCI for those α values where good VaR $_{\alpha}$ estimates are obtained in the previous section. Conversely, for the other two α values whereby QMCI does not provide accurate VaR $_{\alpha}$ estimates, the RMSE scaling fails to show any advantage, and is worse than for CMCI. The underlying causes of these behaviors are explained in Section 5.4.

5.4. Systematic error analyses for VaR $_{\alpha}$ and CVaR $_{\alpha}$ estimation. In Section 5.3, we performed CMCI and QMCI computations to estimate the VaR and CVaR for four different α values, and demonstrated that QMCI provides both accurate estimates and an advantage in RMSE scaling for only two of these values. In this section, we analyse the systematic errors present in QMCI in order to understand the reasons behind this. Specifically, we introduce the concept of “scaled error”, which is the ratio of a particular systematic error to the analytical value. This analysis is critical in order to explain the observed variations in QMCI performance.

We begin by defining the systematic errors for CVaR $_{\alpha}$ estimation, noting that similar definitions for $\widehat{g(\gamma)}$ are omitted for brevity. We then present the behavior of these errors in Fig. 4 as a function of VaR $_{\alpha}$ and the VaR $_{\alpha}$ values corresponding to the four α values analysed previously are marked with vertical lines. Table 4 lists the scaled errors for the four α cases, providing a quantitative comparison. Finally, we examine the maximum of the scaled errors as a function of the number of qubits, to determine the qubit count required to suppress these errors below an empirically determined upper bound.

Our findings show that for the two α values yielding good estimates, the scaled errors are of the order of 10^{-3} or smaller. For the other two α values, the scaled errors exceed 10^{-3} . Furthermore, we demonstrate that for a Gaussian distribution truncated to $[-6\sigma, 6\sigma]$ and loaded using a circuit with greater than or equal to 15 qubits, then the maximal error ratios are all less than 10^{-3} for CVaR $_{\alpha}$ estimation and for VaR $_{\alpha}$ within the range $(-\infty, 4\sigma]$. This indicates that increasing the number of qubits can effectively reduce systematic errors, thereby enhancing the accuracy of QMCI estimates for a broader range of α values. However, this assumes the existence of state-preparation methods that can accurately and efficiently load such distributions.

5.4.1. Systematic errors for CVaR $_{\alpha}$ estimation. We now provide concrete expressions for the truncation error and the combined discretisation and thresholding errors in CVaR $_{\alpha}$ estimation. This

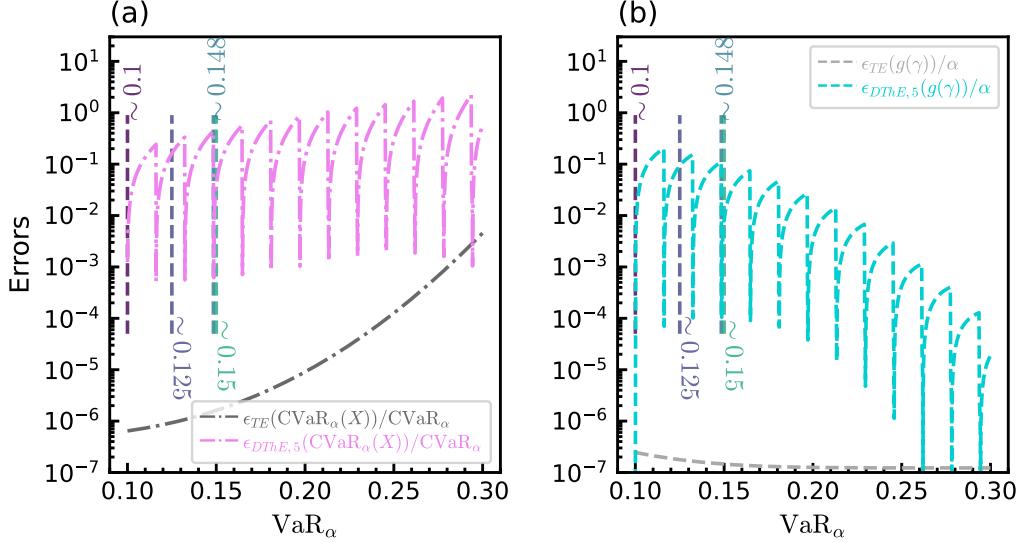


Figure 4. The ratio of $\epsilon_{DThE,5}$ (the combined discretisation and thresholding error) and the truncation errors to the analytical values as a function of VaR_α for (a) $CVaR_\alpha$ and (b) $g(\gamma)$ estimations with the 5-qubit Gaussian distribution (with the parameters given in Table 1). $CVaR_\alpha(X)$ and $g(\gamma)$ represent the function applied, and $CVaR$ and α (i.e., without the random variable X) represent the analytical values. The four VaR_α values marked with the vertical dashed lines correspond to the α values considered in Tables 2–4. The exact values of the scaled errors for the marked VaR_α values are given in Table 4.

	$\epsilon_{DThE,5}(g(\gamma))/\alpha$	$\epsilon_{DThE,5}(CVaR_\alpha(X))/CVaR_\alpha$
$VaR_{0.50} = 0.1$	2.73×10^{-8}	0.001
$VaR_{\sim 0.69} = 0.125$	0.093	0.162
$VaR_{\sim 0.83} \sim 0.148$	0.001	0.003
$VaR_{\sim 0.84} = 0.15$	0.008	0.039

TABLE 4. The ratios of $\epsilon_{DThE,5}$ (the combined discretisation and thresholding error) to their analytical values for $CVaR_\alpha$ and $g(\gamma)$ estimations with the 5-qubit Gaussian distribution (with the parameters given in Table 1). For $\alpha = 0.5$ and $\alpha \sim 0.83$, the error ratios for the $CVaR_\alpha$ and $g(\gamma)$ estimations are of the order of (or smaller than) 10^{-3} , and these are the same α values that give accurate results in Tables 2–3 and show better RMSE scaling in Fig. 3. For $\alpha \sim 0.69$ and $\alpha \sim 0.84$, the error ratios for the $g(\gamma)$ estimations (marked with red) are, respectively about two orders and an order of magnitude larger than 10^{-3} , and these are the α values for which QMCI give worse estimates than CMCI in Table 2. For $\alpha \sim 0.83$, the error ratio for $CVaR_\alpha$ is an order of magnitude larger than 10^{-3} , and the RMSE values of CMCI and QMCI for this α value in Table 3 are comparable to each other. Finally, the error ratio for $CVaR_\alpha$ for $\alpha \sim 0.69$ is two orders of magnitude larger than 10^{-3} , and the RMSE values of QMCI for this α value in Table 3 are an order of magnitude larger than the RMSE values of CMCI.

exemplifies how we consider concrete error expressions tailored for a specific function applied. Additionally, Figs. 4 - 5 show the “scaled errors”, which we prefer for meaningful comparisons of the errors in different regimes. However, this is only possible when the analytical value or a ground truth for the estimated quantity is known.

For a given (inclusive) lower threshold value $\text{VaR}_\alpha = \gamma$, CVaR_α is given by

$$\text{CVaR}_\alpha(X) = \mathbb{E}[X|X \geq \gamma] = \frac{1}{\mathbb{P}(X \geq \gamma)} \int_{\gamma}^{\infty} xf(x)dx, \quad (5.7)$$

and we write the truncated conditional expectation as

$$\mathbb{E}[X|X \geq \gamma]_T = \frac{1}{\mathbb{P}(X \geq \gamma)} \int_{\gamma}^{x_u + \Delta/2} xf(x)dx, \quad (5.8)$$

where we assume that $\mathbb{P}(X \geq \gamma)$ is known (the same for all following definitions). The truncation error then becomes

$$\epsilon_{TE}(\text{CVaR}_\alpha(X)) := \epsilon_{TE}(\mathbb{E}[X|X \geq \gamma]) = |\mathbb{E}[X|X \geq \gamma] - \mathbb{E}[X|X \geq \gamma]_T|. \quad (5.9)$$

Next, we write the discretised conditional expectation as

$$\mathbb{E}[X|X \geq \gamma]_{DTh,n} = \frac{1}{\mathbb{P}(X \geq \gamma)} \sum_{i=I_\alpha}^{N-1} x_i f(x_i) \Delta, \quad (5.10)$$

where I_α is the index of the support point corresponding to $\gamma + \frac{\Delta}{2}$, and $N = 2^n$ is the number of points in the support for n qubits. If $\gamma + \frac{\Delta}{2}$ is exactly equal to a support point, $\mathbb{E}[X|X \geq \gamma]_{DTh,n} = \mathbb{E}[X|X \geq \gamma]_{D,n}$. Otherwise, $\mathbb{E}[X|X \geq \gamma]_{DTh,n}$ also includes the thresholding error. Finally, we represent the combined discretisation and thresholding errors as

$$\epsilon_{DThE,n}(\mathbb{E}[X|X \geq \gamma]) = |\mathbb{E}[X|X \geq \gamma]_T - \mathbb{E}[X|X \geq \gamma]_{DTh,n}|. \quad (5.11)$$

Similarly, starting from (5.2), we can write the intermediate quantities and errors for $g(\gamma)$.

5.4.2. Numerical analysis of the systematic errors in VaR_α and CVaR_α estimations. Fig. 4 shows the ratio of the combined discretisation and thresholding error ($\epsilon_{DThE,5}$) and truncation errors to the analytical values as a function of VaR_α for (a) CVaR_α and (b) $g(\gamma)$ estimations. The functions $\text{CVaR}(X)$ and $g(\gamma)$ represent the estimated values, with their analytical values represented by CVaR and α (i.e., without the random variable X). Fig. 4 shows that the truncation error ratio is almost always smaller than the combined discretisation and thresholding error. We observe that $\epsilon_{DThE,5}$ exhibits “troughs” at certain VaR_α values. This is explained by the increase of the thresholding error when the threshold γ is shifted from a point that is exactly between two support points to another mid-point. $\epsilon_{DThE,5}$ reaches its maximum just before γ equals the next mid-point, and the size of the error is represented in Fig. 1. For the four aforementioned values of α , we report the values of the different ratios error to the analytical values in Table 4.

Table 4 shows that, for $\alpha \approx 0.69$ and $\alpha \approx 0.84$, $\epsilon_{DThE,5}(g(\gamma))/\alpha$ values are significantly larger than 10^{-3} , correlating with the poorer QMCI performance compared to CMCI for the estimation of $g(\gamma)$ for these α values in Table 2. In contrast, for $\alpha = 0.5$ and $\alpha \approx 0.83$, $\epsilon_{DThE,5}(g(\gamma))/\alpha$ values are of the order of or smaller than 10^{-3} , and QMCI provided accurate estimations of $g(\gamma)$ for these α values in Table 2. Similarly, RMSE scaling for CVaR_α estimation in Fig. 3 demonstrates a quadratic speed up for $\alpha = 0.5$ and $\alpha \approx 0.84$, and Table 4 shows that the $\epsilon_{DThE,5}(\text{CVaR}_\alpha(X))/\text{CVaR}_\alpha$ values for these α values are again of the order of 10^{-3} . For $\alpha \sim 0.84$, $\epsilon_{DThE,5}(\text{CVaR}_\alpha(X))/\text{CVaR}_\alpha$ value is an order of magnitude larger than 10^{-3} , and the RMSE values of CMCI and QMCI for

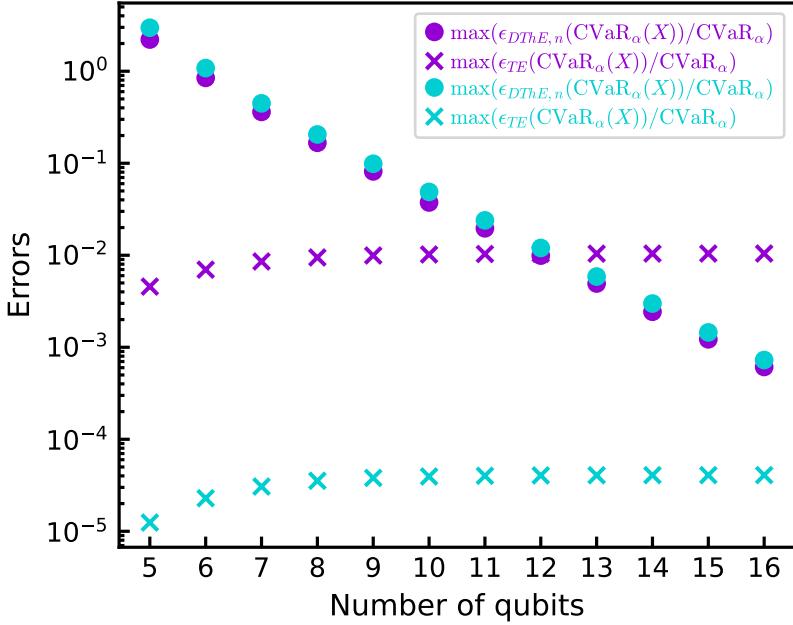


Figure 5. Maximum of the ratio of $\epsilon_{DThE,5}$ (the combined discretisation and thresholding errors) to analytical values for $CVaR_\alpha$ estimations for VaR_α in the range $(-\infty, 4\sigma]$ with the 5-qubit Gaussian distribution (with the parameters given in Table 1). Different colours represent different truncation ranges. Purple (turquoise) represents the Gaussian distribution with range $[-5\sigma, 5\sigma]$ ($[-6\sigma, 6\sigma]$).

this α value in Table 3 are of the same order, with a similar RMSE scaling in Fig. 3. Finally, $\epsilon_{DThE,5}(CVaR_\alpha(X))/CVaR_\alpha$ for $\alpha \sim 0.69$ is two orders of magnitude larger than 10^{-3} , and the RMSE values of QMCI for this α value in Table 3 is an order of magnitude larger than the RMSE value of CMCI. To summarise, we observe that if the scaled errors are below 10^{-3} , QMCI provides accurate estimates and exhibits quadratic advantage.

5.4.3. Maximum of the scaled errors for $CVaR_\alpha$ estimation. Given the empirical upper bound (of 10^{-3}) for the scaled error, we now analyse the scaled errors for $CVaR_\alpha$ estimation to determine the number of qubits required to keep systematic errors *sufficiently suppressed* in order that QMCI can provide accurate estimations. In particular, we show in Fig. 5 the maximum of the ratio of $\epsilon_{DThE,5}$ to analytical values for $CVaR_\alpha$ estimations for VaR_α in the range $(-\infty, 4\sigma]$ for the Gaussian distribution. We analyse two different truncation ranges $[-5\sigma, 5\sigma]$ and $[-6\sigma, 6\sigma]$. Fig. 5 shows that the maximum of ratio truncation error for 5σ truncation is never $\lesssim 10^{-3}$. However, the 6σ truncation – despite having a slightly larger $\epsilon_{DThE,5}$ error ratio – achieves *sufficiently small* truncation error ratios. In both cases the $\epsilon_{DThE,5}$ error ratios reach *sufficiently small* values (i.e. $\lesssim 10^{-3}$) when the number of qubits is greater than or equal to 15.

Our error analyses show that QMCI can provide accurate VaR_α and $CVaR_\alpha$ estimates when all the scaled errors are kept small which happens when a sufficient number of qubits are used to load the distribution. To our knowledge, the determination of this number in the context of specific computations has not been previously explored to any significant degree.

6. MEAN-CVAR SIMULATION-BASED PORTFOLIO OPTIMISATION OVER STABLE DISTRIBUTIONS

In 1952, Roy formulated a safety-first criterion [74], where an investor minimises the probability of wealth falling below some minimal level. The problem formulated by Roy is equivalent to optimising for a quantile, and hence equivalent to replacing the Variance by the CVaR – see for example [35]. We choose in this Section to focus on the problem of portfolio optimisation formulated by minimising risk (the CVaR_α specifically) while requiring a minimum expected return – see also Ref. [75]. An “equivalent” formulation amounts to maximising the return by restricting large risks. This can be done by swapping the CVaR_α and the expected return in the problem formulation and minimising the negative expected return with a penalised CVaR_α constraint.

Consider a portfolio of n assets with a random return that minimises negative returns (penalising risk) together with budget constraints and bounds on each exposure. Given returns of n assets, where w_i is the weight of the i -th asset in the portfolio, and X_i represents the return of asset i , where $i = 1, \dots, n$, then $\{X_1, \dots, X_n\} \sim \Omega_X$ and $w = (w_1, \dots, w_n)$ are the decision variables. The expected return of the portfolio over all assets is then $\mathbb{E}[X = \sum_{i=1}^n w_i X_i]$. Following insights from Ref. [76, Theorems 3 and 5], we model the risky portfolio optimisation problem as follows:

$$\begin{aligned} & \underset{w \in \mathbb{R}^n}{\text{minimize}} \quad f(w, X) = -\mathbb{E}[X] + \lambda \text{CVaR}_\alpha(X) \\ & \text{subject to} \quad w_{\min} \leq w_i \leq w_{\max}, \quad \forall i = 1, \dots, n, \\ & \quad \sum_{i=1}^n w_i = 1; \end{aligned} \tag{6.1}$$

where w_{\min} and w_{\max} are the minimum and maximum allowable weight for each asset, respectively. The constraint $\sum_{i=1}^n w_i = 1$ implies no leverage. Furthermore, we assume no short positions, that is $w_i > 0$ for all $i = 1, \dots, n$. The problem formulated in (6.1) can be solved using existing solvers such as the Sequential Least-Squares Programming (SLSQP) algorithm [77, 78] that is a well-known method for solving constrained nonlinear optimization in an iterative manner, in which the objective function $f(w, X)$ can be evaluated using CMCI or QMCI. More explicitly, we give the schematic pseudo-code to solve the problem in Algorithm 2.

Algorithm 2: Iterative CVaR_α Portfolio optimisation

- 1 **Input:** Joint distribution of asset returns $\{X_i, i = 1, \dots, n\} \sim \Omega_X$, risk aversion parameter λ , confidence level α ;
 - 2 **Output:** Optimized portfolio weights $w^* \in [0, 1]^n$;
 - 3 Initialize $w^{(0)}$ randomly subject to $\sum_i w_i^{(0)} = 1$;
 - 4 Define objective function $f(w, X) = -\mathbb{E}[w_i^\top X_i] + \lambda \text{CVaR}_\alpha(w_i^\top X_i)$;
 - 5 $k \leftarrow 0$;
 - 6 **repeat**
 - 7 Simulate portfolio returns $X^{(k)} = \sum_{i=1}^n w_i^{(k)} X_i$ using CMCI or QMCI;
 - 8 Calculate CVaR_α for $X^{(k)}$ as $\text{CVaR}_\alpha(X^{(k)})$ using CMCI or QMCI;
 - 9 Update objective function $f^{(k)}(w, X) = -\mathbb{E}[X^{(k)}] + \lambda \text{CVaR}_\alpha(X^{(k)})$;
 - 10 Solve $w^{(k+1)} = \arg \min_w f^{(k)}(w, X)$ subject to $\sum_i w_i = 1$ and $w_{\min} \leq w_i \leq w_{\max}$;
 - 11 $k \leftarrow k + 1$;
 - 12 **until** convergence criterion is met;
 - 13 **return** $w^* = w^{(k)}$;
-

parameter	value / choice
X_i	$\mathcal{N}(\mu_i, \sigma_i^2)$
μ_1	0.10
μ_2	0.10
σ_1	0.05
σ_2	0.10
α	0.95
w_{\min}	0.10
w_{\max}	0.90
λ	0.10
number of samples	10000

TABLE 5. Parameter values for the Mean-CVaR optimisation.

6.1. Numerical Experiments. The Mean-CVaR optimisation formulated in (6.1) involves vector optimisation over uncertainty and the calculation of the expected return and CVaR in terms of multiple assets depend on the weight factor, thus the optimal solution cannot be obtained analytically. The numerical experiments discussed in this section therefore focus on how using QMCI as a subroutine affects the behavior and efficiency of the optimisation algorithm when compared to using CMCI as a subroutine. The numerical experiments will provide a playground to observe the difference in the convergence to the final optimised objective value when using QMCI and CMCI.

We consider the problem where the returns of the i -th asset $X_i, i = 1, \dots, n$ follows some specified distribution. For instance, $X_i \sim \mathcal{L}(\mu_i, c_i)$ or $X_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$, for each asset i , with $i = 1, \dots, n$. For simplicity, we choose a portfolio consisting of $n = 2$ different assets, where the returns X_1 and X_2 are independent and Gaussian distributed with $X_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$, $i=1,2$. The objective function (6.1) is then expanded as follows

$$f(w, X) = -\mathbb{E}[w_1 X_1 + w_2 X_2] + \lambda \text{CVaR}_\alpha(w_1 X_1 + w_2 X_2). \quad (6.2)$$

We refer to Table 5 for the values of all parameters chosen for the numerical experiments. The computation of CVaR_α using both CMCI and QMCI follow the same procedure introduced in Section 5. In the numerical experiments, all weights w_i are initialised as $w_i^{(0)} = \frac{1}{n}$ for Algorithm 2 used to solve the optimisation problem (6.2).

Figure 6 displays the values of the objective function $f(w, X)$ in the iterative procedure of Algorithm 2 for a single run. Within the numerical experiments, the expectation of the weighted returns is estimated either with CMCI or QMCI and $\text{CVaR}_{0.95}$ is estimated using CMCI in order to avoid the propagation of systematic errors that arise when calculating the expected return and the CVaR in the objective function of (6.1) using QMCI, as discussed in Section 5. We observe that the objective values estimated using either QMCI or CMCI converge to stable values with an increasing the number of iterations and the curve follow a similar trajectory up to an offset which seems constant. This indicates that QMCI can serve as an effective subroutine for function estimation within classical optimisation algorithms.

Figure 7 shows the value of the objective function $f(w, X)$ for 100 independent runs of Algorithm 2 over different number of samples for CMCI and QMCI, respectively. We we can see that the average values of the final objective $\bar{f}(w^*, X)$ obtained by Algorithm 2 using QMCI and CMCI converge to consistent stable values as the number of samples increases.

We have performed a Mean-CVaR optimisation of asset allocation for two independently distributed Gaussian distributions. As the CVaR estimation using QMCI is sensitive to the number of samples as well as the error induced when running QMCI, the CVaR was estimated using CMCI

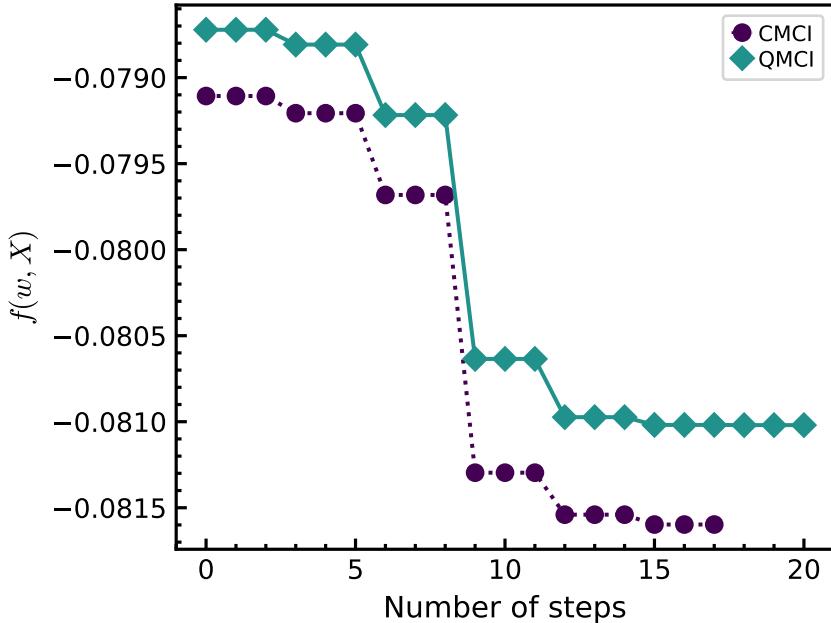


Figure 6. The estimated objective $f(w, X)$ versus the number of iterations in Algorithm 2 using both QMCI and CMCI. The final optimised objective for CMCI is $f(w, X) = -0.0807$ with optimised weights $w^* = [0.7894, 0.2106]$, while for QMCI $f(w^*, X) = -0.0811$ with $w^* = [0.7245, 0.2755]$.

during the optimisation procedure, while the expected return is estimated using QMCI. For a future work, we plan to investigate the resources required to evaluate the objective function fully quantumly.

To conclude this Section, we point out a notable difference between the classical and the quantum approach. When using classical Monte Carlo methods for Mean-CVaR optimisation, the returns of different portfolio assets are sampled from a given distribution. The same sample is then used to evaluate the objective function (both the expected return and the CVaR) with different weight vectors (i.e., decision variables) in the classical optimisation algorithm at each optimisation iteration. This notably ensures that the variation of the objective function estimated at different iterations solely arises from the updates of the weights. However with QMCI, we need to re-evaluate the expected return and CVaR at each iterative step, and thus the variation of the objectives includes the systematic errors that arise from estimating the objective using QMCI, alongside the variation induced by updating the weight vector. This implies that the classical algorithm with QMCI as a subroutine is only practically realisable when the systematic errors on the objective induced by QMCI are negligible compared to the variation on the objective caused by updating the weight vector. Therefore, this requires us to consider quantum circuits with larger numbers of qubits to sufficiently reduce systematic errors. Determining the precise hardware resources required will be the object of future study.

7. ASSET ALLOCATION UNDER UNCERTAINTY AND MEAN-VARIANCE OPTIMISATION

The objective of this Section is to introduce the problem of determining an optimal asset allocation strategy when market conditions are uncertain; whether this is parameter uncertainty or

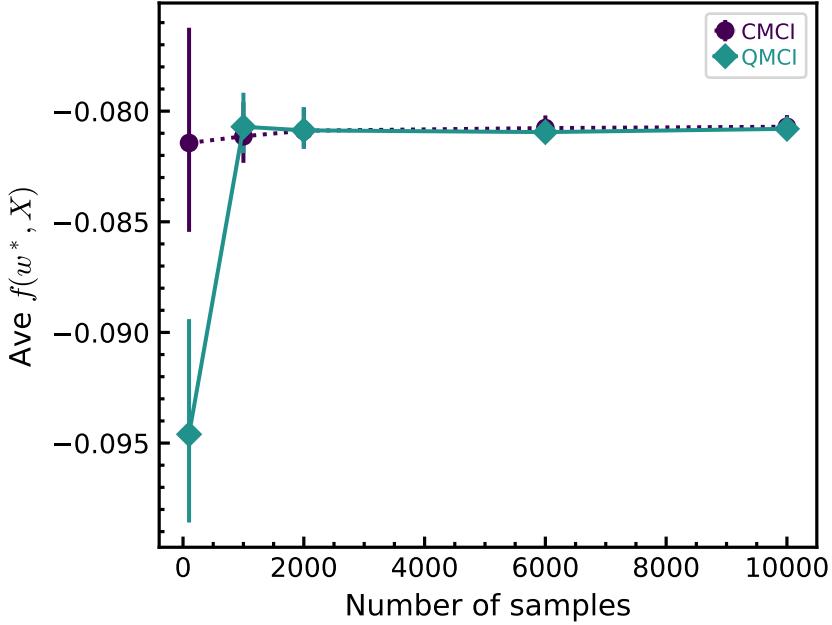


Figure 7. The average value of the optimised objective $f(w^*, X)$ as a function of the number of samples, in which each point of $\bar{f}(w^*, X)$ is obtained by averaging over 100 independent runs.

distributional uncertainty. Indeed, uncertainties on parameters and distributions can be modelled as random variables, random processes, or random fields [79], which leads to an SP problem as stated in (1.1). Thus, QMCI can be a subroutine to aid in solving (1.1). In the context of mathematical finance, Monte Carlo simulations generate multiple scenarios of future market conditions, and the optimisation problem aims to find the allocation strategy that maximises the expected utility under these scenarios.

7.1. Framework and definitions. A popular formulation of investment optimisation is choosing a point on the Capital Allocation Line (See e.g. Ref. [35, page 55]). This problem is equivalent to choosing an allocation to a risk-free asset with positive return and the market risky portfolio. As a particular case, the reader might want to think of a simplified example where there is only one risky asset. Using this formulation, it is possible to optimise the “terminal utility”, $U(V_{T+1})$, a function of the value V_{T+1} of the portfolio at time $T + 1$ (without loss of generality), by choosing the proportion ω_T of the current wealth V_T , at time T , to invest in a risky asset and the remaining in a risk-free asset ($\tilde{\omega}_T = 1 - \omega_T$). We will further assume that no leverage is allowed and hence that $\tilde{\omega}_T \geq 0$ and $\omega_T \geq 0$. The objective function may then be given as

$$\hat{\omega}_T = \arg \max_{\omega_T \in [0,1]} \mathbb{E}_T [U(V_{T+1})], \quad (7.1)$$

where \mathbb{E}_T denotes the expectation at time T , based on all available information up to that time. The terminal utility $U(V_{T+1})$ is a function of the chosen allocation ω_T and the respective returns of the risky and risk-free assets. The wealth equation is

$$\begin{cases} V_{T+1} = V_T \times \left(\omega_T e^{r_{T+1} + R_T^{(f)}} + \tilde{\omega}_T e^{R_T^{(f)}} \right), \\ \tilde{\omega}_T + \omega_T = 1, \end{cases} \quad (7.2)$$

where $r_{T+1} = X_{T+1} - R_T^{(f)}$ represents the “risk premium”, the excess return of the risky asset over the risk-free rate, $R_T^{(f)}$ (which is the deterministic risk-free rate at time T) and X_{T+1} (which is the return of the risky asset at the time $T + 1$). In practice, we usually consider logreturns instead of raw returns, which can be expressed as $X_{T+1} = \log\left(\frac{V_{T+1}}{V_T}\right)$.

Equation (7.1) can be expressed as an integral over all possible future returns, weighted by the predictive distribution of these returns

$$\hat{\omega}_T = \arg \max_{\omega_T \in [0,1]} \int_{\Omega} U \left(V_T \times (\omega_T e^{r_{T+1} + R_T^{(f)}} + (1 - \omega_T) e^{R_T^{(f)}}) \right) p(r_{T+1}|y_T, \Theta) dr_{T+1}, \quad (7.3)$$

where Θ represents the set of parameters that describe the predictive distribution p of the excess returns, and y_T encapsulates all past data up to time T . One more detail is that of the excess market return r_{t+1} which may be modeled as a linear process

$$r_{t+1} = \mu_{t+1} + e_{t+1}, \quad t = 1, \dots, T, \quad (7.4)$$

where μ_{t+1} is the expected return at time $t + 1$ – which may be constant or time-varying based on predictors up to time $t + 1$ – and e_{t+1} is a random error term with mean 0 and variance σ^2 . This equation deliberately does not specify a distribution for the error term, allowing for a flexible model that can accommodate different forms of distribution uncertainty.

7.2. Application to the Mean-Var simulation-based portfolio optimisation problem. We now consider a simple quadratic utility function which defines the Mean-Var optimisation problem, of form

$$U = \mathbb{E}[V] - \lambda \sigma_V^2, \quad (7.5)$$

where $\lambda > 0$ is a factor that represents the risk aversion (in this simple model it is assumed to be a constant) and $\sigma_V = \sqrt{\text{Var}[V]}$ i.e., the volatility of the wealth V of a portfolio of interest. This utility function has a certain theoretical appeal, though we acknowledge that the formulation of the investment problem is rarely setup like this in practice. Firstly, (as argued in [34] and [35]) this would limit the optimisation problem to the time horizon corresponding to the risk-free asset that is under consideration. Secondly, the optimal allocation corresponds to a hypothetical utility function that is not realistic [80], and hence impossible to estimate. Thirdly, the assumption that variations (positive and negative) both contribute to risk is not realistic (typically people desire positive returns and aim to avoid negative returns (losses); variance penalises both). However, if an investor were to be rational and uniformly risk averse, this formulation is a reasonable description of reality. From (7.5) the utility to maximise is thus given by

$$\begin{aligned} U(V_{T+1}) &= \mathbb{E}[V_{T+1}] - \lambda \sigma_{V_{T+1}}^2, \\ &= \mathbb{E}[V_{T+1}] - \lambda \left(\mathbb{E}[V_{T+1}^2] - \mathbb{E}[V_{T+1}]^2 \right). \end{aligned} \quad (7.6)$$

7.2.1. First numerical example: Mean-Var portfolio optimisation with a Gaussian asset. To simulate a realistic example for running numerical experiments, we consider the simplified setting presented in Table 6, where the risk return r_{T+1} is modelled by a Gaussian and we consider a situation with a moderate level of risk aversion. We can expand the objective function given in (7.6) by inserting the wealth equation given in (7.2), so that the objective function then has an explicit dependence on $\{r_{T+1}, \omega_T, R_T^{(f)}, \lambda\}$, and if we re-parameterise the equation based on the fact that the exponential of the risk return will be log-normally distributed i.e.,

$$e^{r_{T+1}} \equiv \varrho_{r_{T+1}} \sim \mathcal{LN}(\mu, \sigma^2), \quad (7.7)$$

parameter	value / choice
unit of T	1 year
V_T	100\$
$R_T^{(f)}$	6%
λ	0.5
r_{T+1}	$\mathcal{N}(\mu, \sigma^2)$
σ	0.2
μ	$12\% - R_T^{(f)}$

TABLE 6. Choice of parameters for the numerical example of Mean-Var SBO for portfolio optimisation of Section 7.2. The values used and the general approach is similar to [35].

we find that the utility to optimise can be expressed as

$$U(\varrho_{T+1}, \omega_T, R_T^{(f)}, \lambda) = V_T \left(\omega_T e^{R_T^{(f)}} \int_{\mathbb{R}} \varrho_{t+1} p(\varrho_{t+1}) d\varrho_{t+1} + \tilde{\omega}_T e^{R_T^{(f)}} \right) \\ - \lambda V_T^2 \omega_T^2 e^{2R_T^{(f)}} \left(\int_{\mathbb{R}} \varrho_{t+1}^2 p(\varrho_{t+1}) d\varrho_{t+1} - \left(\int_{\mathbb{R}} \varrho_{t+1} p(\varrho_{t+1}) d\varrho_{t+1} \right)^2 \right). \quad (7.8)$$

The integrals in the expression are equivalent to the expectation value and variance of ϱ_{T+1} , and these can be calculated using QMCI. The full derivation of this expression is detailed in Appendix A. We summarise in Table 6 the values chosen to obtain the subsequent numerical results.

Closed-form solution

The closed-form solution for the optimisation problem is derived in Appendix A. From (A.11), the optimal value for the 5-bit discretised version of the problem with the parameters defined in Table 6 is $\hat{\omega}_T^{\text{disc}} = 0.0164$.

Numerical Experiments

We now discuss the numerical experiments that were carried out using QMCI as a subroutine to estimate the utility when performing the SBO described in this section. In particular, we explore how the effects of noise impacts the optimisation when performed on the QMCI engine.⁶ Note that because the purpose of this numerical experiment is to study the performance of QMCI optimisation when performed in both noiseless and noisy regimes, then in contrast to the previous sections we do not focus on comparisons of the performance of QMCI with CMCI. We therefore do not carry out a detailed analysis of errors as was done in previous sections.

The application of the technique of NA-QAE for error mitigation discussed in Section 3.2.1 for mitigating the noise when running QMCI is investigated, under the assumption that the Gaussian noise parameter k_μ is negligible – equivalently assuming a depolarising noise model. The numerical experiments were carried out using both Qulac’s noiseless state-vector simulator [81] and a noisy simulator that emulates the result of computations performed on Quantinuum’s trapped-ion quantum computer, H1.⁷ In general, this study represents the first time that the error-mitigation

⁶The use case described in this section is the most promising of the two use cases discussed in this paper for a study involving noise, as no enhancement operations are necessary in the circuit, which means the circuits are relatively low depth and do not contain a large number of qubits.

⁷It is known from previous calibration runs on the noisy H1 simulator that k_μ is negligible for the case of shallow circuits with a small number of qubits, and thus NA-QAE methods should be applicable based on our assumptions.

Samples	2000	3000	4000	5000	6000
m_k value (largest circuit)	16	32	32	32	64
two-qubit gates (largest circuit)	3490	6914	6914	6914	13762
Expected number of depolarising errors (largest circuit, $1 - \varphi_2 = 8.830 \times 10^{-4}$)	4	8	8	8	16
Expected number of depolarising errors (largest circuit, $1 - \varphi_2 = 1.766 \times 10^{-5}$)	0	0	0	0	0

TABLE 7. Information about the largest circuits that would be run for given number of samples assuming the optimal allocation. All relevant parameters are defined in Section 3.2.1.

protocol of NA-QAE has been directly applied to a problem of real-world interest that makes use of QMCI as a subroutine. The notation for this section follows that of Section 3.2.

When considering the presence of noise, based on H1 device characteristics as specified in July 2024 [82], the maximum gate count of circuits (in terms of two-qubit gates) that can be reliably run can be estimated. H1 has a two-qubit gate average infidelity of $1 - \varphi_2 = 8.830 \times 10^{-4}$. Assuming a depolarising noise model, this corresponds to a depolarising error probability of $p_{\text{dep}} = 1 - \frac{4}{3}(8.830 \times 10^{-4}) = 0.999$; ignoring all other sub-dominant sources of noise compared to the two-qubit gate error this means that approximately 850 two-qubit gate operations can be performed, on average, before there is an error that causes complete decoherence. However, in a future scenario where the infidelity is decreased by approximately 1.7 orders of magnitude (i.e. the infidelity is instead taken as being $1 - \varphi_2 \sim 1.766 \times 10^{-5}$), then $\sim 42,000$ two-qubit gate operations, on average, can be performed before there is such an error. A 5-qubit circuit which prepares a log-normal distribution for ϱ_{T+1} with the parameters given in Table 6 contains ~ 66 two-qubit gates when compiled to run on H1. Each Grover iteration then adds an additional ~ 214 gates. Thus, realistically for the current infidelity, only a few Grover iterates can be performed before errors start to get out of hand.

Information regarding the largest circuits that would be run for given numbers of samples in the noiseless case (assuming the optimal allocation value for ω_T) is given in Table 7 for both the current infidelity and the future scenario. We can see that by 3000 and 6000 samples, i.e. where the largest circuits to run are $m_k = 32$ and $m_k = 64$, respectively, noise effects would be expected to be significant at the current infidelity. However, in the future scenario, all required computations should be able to be performed coherently.⁸ Thus, in order to be able to meaningfully run NA-QAE (i.e. without noise becoming overwhelming), the level of noise in the simulation was artificially reduced in order to correspond to the future scenario. The number of QMCI samples that were probed are then $\mathcal{S} \in \{2000, 3000, 4000, 5000, 6000\}$.

In order to model the exponential of the risk-return $\varrho_{r_{T+1}}$ given in (7.7), a 5-qubit log-normal distribution was prepared that was loaded directly in the QMCI engine using the in-built state-preparation library. The QMCI engine was then used to build efficient, low depth circuits that estimate both the expectation value and variance of the random variable $\varrho_{r_{T+1}}$ using QMCI.

Within the QMCI engine, an implementation of the MLE-QAE algorithm, optimised based on empirical studies carried out previously to determine the optimal number of shots to run for all m_k values in the EIS, alongside efficient use of all provided samples, was used for the studies. An uniform prior was placed on both θ_a and k_σ , where in the latter case the range was from k_σ to $2k_\sigma$, based on the estimate $k_\sigma = 0.004$ corresponding to $\tilde{p}_{\text{coh}} = 0.992$ derived from (3.16).

⁸Especially considering that for NA-QAE, fewer deep circuits are expected to be run than in the noiseless case i.e., more shots of lower depth circuits will be run.

For each total number of samples probed, 100 runs of SBO in the noiseless regime and two runs on the noisy simulator (where one run used NA-QAE to perform error mitigation and the other did not implement any error-mitigation methods), were performed.⁹

For each run of SBO, the local minimum was determined using the bounded version of Brent's algorithm, which scans the range $\hat{\omega}_T \in [0, 1]$ and at each scan point performs QMCI in order to calculate the value of the objective function given in (7.6) (by running the expectation value and variance QMCI circuits described previously). We note that Brent's algorithm is not particularly well-suited to optimisation problems in the presence of noise, however given that the aim of this study is to directly compare the performance of the optimisation in both the noiseless and noisy regimes, then we choose the same optimiser in both cases – with Brent's algorithm chosen for its relative simplicity.

For a given run of QMCI, when calculating the objective function, one consideration that is important is – for a fixed number of samples – how best to divide these samples between the expectation value and variance calculations (and thus also to the individual second moment and expectation value calculations that make up the variance calculation), in order to minimise the total error on the final estimate. In order to determine the optimal division of samples, an additional classical minimisation at each scan point, based on the SLSQP algorithm, was performed in order to minimise the theoretical expected asymptotic upper bound on the RMSE of the estimate of the utility for the given number of samples.¹⁰

The results are given in Fig. 8, which demonstrates the optimal allocation as a function of the number of samples for both runs on the noisy emulator (with and without NA-QAE) alongside the average optimal allocation, and the maximum and minimum optimal allocations, observed in the noiseless case. The analytical value for the minimum is also shown.

Firstly, we note that, unfortunately, due to the fact that only single runs on the noisy emulator were able to be performed, it is difficult to say anything conclusive with regards to the performance of the noisy runs, particularly given the high variance exhibited in the noiseless case. In regards to the comparison of running on the noisy simulator with and without NA-QAE, it may be argued that, on average, the runs performed without applying NA-QAE are slightly further from the true minimum, but this is not always the case (for example the values for 5000 and 6000 samples are closer); indeed this can easily be explained by low statistics (a sample size of one). As such, it is difficult to tell if the error mitigation has improved the results. For example, we can see that in both cases the majority of values (four out of five in both cases) are consistent with values that were also observed in the noiseless case.

In general the issue for this study is that both optimisation and estimation are performed, which makes rigorous comparison difficult. In order to explore the performance in more detail, we now focus on just the estimation part of the routine. We calculated the utility function evaluated at the true optimal 100 times when run on the noiseless and noisy simulators, in the latter case both with and without using NA-QAE. Figure 9 compares the convergence properties of the RMSE of the estimates as a function of the number of samples. Here we can see that in the noisy case NA-QAE generally improves the convergence, bringing the performance closer to the noiseless case, with four out of five data points having a lower RMSE than when the circuit is run on the noisy emulator without applying error mitigation. This demonstrates the power of the NA-QAE method for this simplified example. As to why there is a discrepancy for 2000 uses, it is likely that the level of

⁹We were only able to perform single runs for each number of samples on the noisy simulator due to the very large computational time required to run the calculations (as it was not possible to parallelise computations when running on the simulator).

¹⁰This calculation assumes unbiased estimators – clearly not the case here for a finite number of samples, but correct in the asymptotic case.

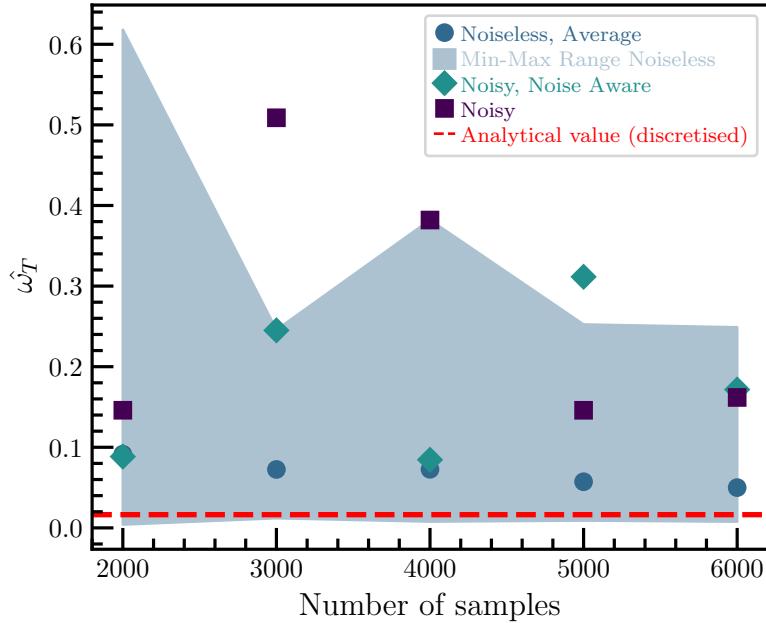


Figure 8. Results of the full SBO for determining the optimal allocation as a function of the number of samples, where the utility function is evaluated at each scan point using the QMCI engine and the respective number of samples. (blue) gives the results using a noiseless simulator, which are averaged across 100 independent runs, and the shaded region represents the range of values observed from those 100 runs. (green) and (purple) give the results using a noisy simulator of Quantinuum’s H1 device with a single run in each case, the former with NA-QAE methods applied for error mitigation, and the latter without any error mitigation applied. (red) gives the analytical value for the minimum.

noise is so low based on this reduced noise setting (the deepest circuits run have $m_k = 16$) that the NA-QAE method does not work effectively – or this could also be a statistical fluctuation.

Despite being unable to make any conclusive remarks as regards the benefits of applying NA-QAE to the full SBO problem, it is still worth making the point that this is the first time that NA-QAE has been applied to a relevant real-world problem that would make use of QMCI as a subroutine. It is reassuring that the results we obtain are, in general, consistent with results one could also expect to obtain in the noiseless case (even considering the reduced level of noise explored in this study).

Whilst comparisons between the noiseless performance of QMCI and CMCI is not the focus of this section (this is rather the primary focus of the studies in previous sections), if we ignore the noisy runs and just focus on the noiseless case, we can also see that even here the optimisation (on average) is still some way off from converging to the true minimum as would be expected for CMCI, even at $S = 6000$; this is worth discussing briefly and is likely based on the following:

- The evaluation of the cost function relies on using QMCI to estimate the expectation value and the variance (which is itself composed of separate evaluations of the expectation value and second moment).
- Because the total number of samples is fixed, the samples specified in the plot are shared between each of these evaluations based on minimising the theoretical total error on the

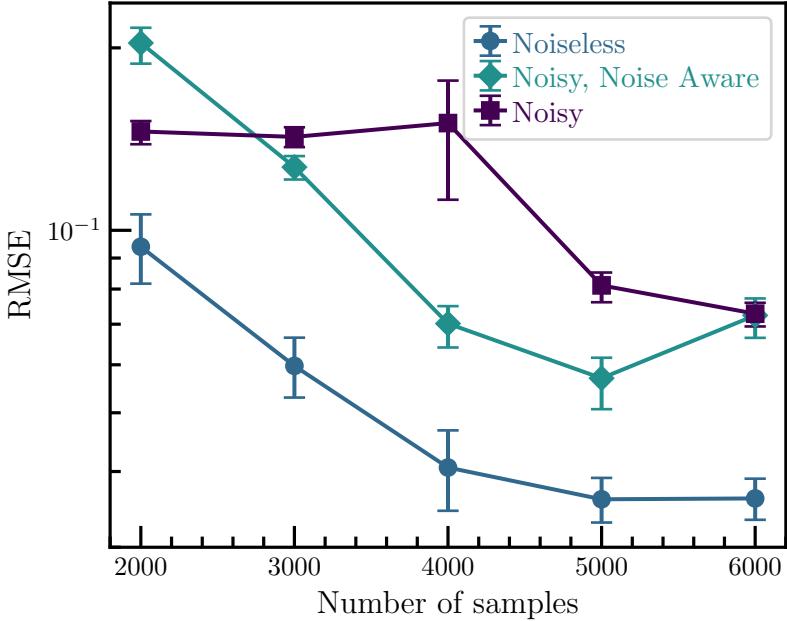


Figure 9. RMSE as a function of the number of samples when the utility function is calculated using the QMCI engine and evaluated at the true analytical minimum $\hat{\omega}_T^{\text{disc}}$. (blue) gives the results using a noiseless simulator. (green) and (purple) give the results using a noisy simulator of Quantinuum’s H1 device, the former with NA-QAE methods applied for error mitigation, and the latter without any error mitigation applied. All points are averaged across 100 independent runs.

utility as discussed, i.e. each integral does not make use of \mathcal{S} samples but rather some fraction of \mathcal{S} .

- Performing QMCI relies on performing QAE, and previous studies have shown that for smaller numbers of samples, i.e. $\mathcal{S} < 1000$, the convergence properties of the MLE-QAE algorithm differ from the asymptotic behaviour of QAE. The minimisation of the theoretical total error on the utility mentioned above assumes the asymptotic behaviour of the convergence, and this is therefore a source of systematic error.
- Each of the integral evaluations in the utility individually contribute their own sources of error. These have different magnitudes depending on the the number of samples assigned to the evaluation (based on running QAE) alongside the specific systematic errors (as described in Section 4) and convergence properties of the particular function being applied to the random variable (expectation value or second moment in this case).
- Because the theoretical error on the second moment calculation, arising from the specific systematic errors and convergence properties of the function applied, is known to be significantly larger than for expectation values [19], then assuming asymptotic QAE convergence, the majority of samples will be assigned to that calculation. This therefore leads to additional errors on the expectation value calculations that only have a small number of samples assigned, given that the QAE convergences will not necessarily behave as expected.
- The total error on the utility is thus compounded when these evaluations are combined. As the cost function is then evaluated at each scan point, errors propagate throughout the entire minimisation process, likely resulting in large deviations for the converged optimum.

Given these points, it is likely that a very large number of samples (and a larger number of qubits) would be required, such that these sources of error are sufficiently suppressed, to be able to achieve the relevant accuracy to provide quadratic advantage for this particular problem at this scale, even in the noiseless case. We thus conclude that this is likely a problem more suited to running in the future, when the capabilities of quantum computers have been developed further, i.e. when deeper circuits with a larger number of qubits can be run.

8. CONCLUSIONS

This work can be characterised in the context of hybrid classical-quantum computation where an open question is: how much classical computation should be offloaded to the quantum device? For the applications studied, the hybridisation was investigated for the evaluation of the objective function of an optimisation problem, where all or part of the objective function can a priori be estimated quantumly. Specifically, this paper has thoroughly explored the limitations and bottlenecks associated with using quantum Monte Carlo integration (QMCI) as a subroutine for simulation-based optimisation (SBO) in quantitative finance applications, focusing mainly on the estimation of quantities like Value-at-Risk (VaR) and Conditional-Value-at-Risk (CVaR). We identified two key challenges: (i) systematic errors arising from limited resources and (ii) the impact of noise in quantum hardware. Through a detailed analyses of these challenges, we demonstrated the critical role that systematic errors play in determining whether QMCI can outperform classical Monte Carlo integration (CMCI), and show that noise-aware quantum amplitude estimation (NA-QAE) can be used to mitigate device noise when estimating the cost function of a SBO. We also explored the possibility of using QMCI and CMCI together to estimate different terms in a composite cost function.

We identified and provided a detailed description of all the systematic errors. Then, we analysed the systematic errors in VaR and CVaR estimations to show that QMCI not only provides accurate estimations but also offers a quadratic speed-up over classical methods, but only when the systematic errors are sufficiently small. We showed that the systematic errors primarily stem from the limited number of qubits used to load probability distributions and the precision of the distribution-loading. With these analyses, we then determined an empirical upper bound for the systematic errors to achieve quantum speed-up. Then, by assuming we can load the distribution with a small state-preparation error and using the empirical upper bound on the error magnitude, we determine the minimum number of qubits required for quantum speed up in CVaR estimations. This highlights the main bottleneck for the speed-up: the ability to load distributions accurately using a large number of qubits, which remains an open problem in quantum computing.

Additionally, we show that the main limitation for VaR and CVaR estimations is the thresholding error, and the size of systematic errors depends on the specific function applied in QMCI. For instance, the same quantum state used for distribution loading can yield accurate estimations for certain functions, such as mean calculations. This insight led us to propose the use of QMCI and CMCI in tandem, where QMCI estimates terms in a composite cost function that have small associated systematic errors, while CMCI handles other terms. This hybrid approach demonstrates that quantum and classical methods can be effectively integrated to leverage the strengths of each.

Lastly, we examined the effects of noise in quantum hardware on SBO applications. While NA-QAE appears to improve results for individual cost function estimations by bringing them closer to the noiseless case, the overall impact on the full SBO problem remains inconclusive. This is primarily due to the complexity of the optimisation process, which requires more extensive testing on quantum hardware. Nevertheless, for estimation tasks, NA-QAE shows promise as an effective error mitigation technique.

In summary, while QMCI has the potential to offer significant computational advantages over classical methods for financial applications, there remain several open challenges that must be

addressed for these advantages to be fully realised. Chief among these challenges is the need for efficient and accurate methods to load probability distributions using a large number of qubits, as well as managing the impact of noise in current quantum hardware. Despite these challenges, the hybrid QMCI-CMCI approach and NA-QAE represent promising strategies for overcoming some of these limitations, laying the groundwork for further exploration in the intersection of quantum computing and quantitative finance.

It is known that underlying assets being modelled by Gaussian and log-normal distributions is unrealistic, as discussed in Appendix B. Thus a possible and important extension to this work is analyse more realistic distributions, such as heavy-tailed distributions – see Appendix C for our initial analyses of the systematic errors associated with loading Lévy distributions on quantum computers. Finally, as another promising extension of our work, the framework discussed in this work can be further generalised and we leave this as an open problem for the community. For instance, one might consider a more complex problem such as “trade-off optimisation” with different risk measures – see Appendix D.

ACKNOWLEDGEMENTS

We thank M. Spranger for his work that helped form the background to this paper, and for all of the fruitful discussions we had along the way. GK would like to acknowledge that this work has been partially supported by project MIS 5154714 of the National Recovery and Resilience Plan Greece 2.0 funded by the European Union under the NextGenerationEU Program. GK further acknowledges support of the Czech Science Foundation (2307947S).

Disclaimer. This paper was prepared for information purposes and is not a product of HSBC Bank Plc. or its affiliates. Neither HSBC Bank Plc. nor any of its affiliates make any explicit or implied representation or warranty and none of them accept any liability in connection with this paper, including, but not limited to, the completeness, accuracy, reliability of information contained herein and the potential legal, compliance, tax or accounting effects thereof. Copyright HSBC Group 2024.

REFERENCES

- [1] D. J. Egger, C. Gambella, J. Marecek, S. McFaddin, M. Mevissen, R. Raymond, A. Simonetto, S. Woerner, and E. Yndurain, “Quantum computing for finance: State-of-the-art and future prospects,” *IEEE Transactions on Quantum Engineering*, vol. 1, pp. 1–24, 2020. [Online]. Available: <https://ieeexplore.ieee.org/document/9222275>
- [2] D. Herman, C. Googin, X. Liu, A. Galda, I. Safro, Y. Sun, M. Pistoia, and Y. Alexeev, “A survey of quantum computing for finance,” *arXiv preprint arXiv:2201.02773*, 2022. [Online]. Available: <https://doi.org/10.48550/arXiv.2201.02773>
- [3] L. K. Grover, “A fast quantum mechanical algorithm for database search,” in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, ser. STOC ’96. New York, NY, USA: Association for Computing Machinery, 1996, p. 212–219. [Online]. Available: <https://doi.org/10.1145/237814.237866>
- [4] P. Shor, “Algorithms for quantum computation: discrete logarithms and factoring,” in *Proceedings 35th Annual Symposium on Foundations of Computer Science*, ser. SFCS-94. IEEE Comput. Soc. Press, 1994. [Online]. Available: <http://dx.doi.org/10.1109/SFCS.1994.365700>
- [5] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman, “Exponential algorithmic speedup by a quantum walk,” in *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, ser. STOC ’03. New York, NY, USA: Association for Computing Machinery, 2003, p. 59–68. [Online]. Available: <https://doi.org/10.1145/780542.780552>
- [6] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, “Quantum amplitude amplification and estimation,” *Quantum Computation and Information*, p. 53–74, 2002. [Online]. Available: <http://dx.doi.org/10.1090/conm/305/05215>
- [7] A. Montanaro, “Quantum speedup of Monte Carlo methods,” *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 471, no. 2181, p. 20150301, 2015. [Online]. Available: <https://doi.org/10.1098/rspa.2015.0301>

- [8] S. Herbert, "Quantum Monte Carlo integration: The full advantage in minimal circuit depth," *Quantum*, vol. 6, p. 823, Sep 2022. [Online]. Available: <http://dx.doi.org/10.22331/q-2022-09-29-823>
- [9] P. Intallura, G. Korpas, S. Chakraborty, V. Kungurtsev, and J. Marecek, "A survey of quantum alternatives to randomized algorithms: Monte Carlo integration and beyond," *arXiv preprint arXiv:2303.04945*, 2023. [Online]. Available: <https://doi.org/10.48550/arXiv.2303.04945>
- [10] D. Herman, C. Googin, X. Liu, Y. Sun, A. Galda, I. Safro, M. Pistoia, and Y. Alexeev, "Quantum computing for finance," *Nature Reviews Physics*, vol. 5, no. 8, p. 450–465, Jul 2023. [Online]. Available: <http://dx.doi.org/10.1038/s42254-023-00603-1>
- [11] A. Abbas, A. Ambainis, B. Augustino, A. Bärtschi, H. Buhrman, C. Coffrin, G. Cortiana, V. Dunjko, D. J. Egger, B. G. Elmegreen *et al.*, "Quantum optimization: Potential, challenges, and the path forward," *arXiv preprint arXiv:2312.02279*, 2023. [Online]. Available: <https://doi.org/10.48550/arXiv.2312.02279>
- [12] R. Shaydulin, C. Li, S. Chakrabarti, M. DeCross, D. Herman, N. Kumar, J. Larson, D. Lykov, P. Minssen, Y. Sun, Y. Alexeev, J. M. Dreiling, J. P. Gaebler, T. M. Gatterman, J. A. Gerber, K. Gilmore, D. Gresh, N. Hewitt, C. V. Horst, S. Hu, J. Johansen, M. Matheny, T. Mengle, M. Mills, S. A. Moses, B. Neyenhuis, P. Siegfried, R. Yalovetzky, and M. Pistoia, "Evidence of scaling advantage for the quantum approximate optimization algorithm on a classically intractable problem," *Science Advances*, vol. 10, no. 22, May 2024. [Online]. Available: <http://dx.doi.org/10.1126/sciadv.adm6761>
- [13] S. Woerner and D. J. Egger, "Quantum risk analysis," *npj Quantum Information*, vol. 5, no. 1, p. 15, 2019.
- [14] D. J. Egger, R. G. Gutierrez, J. Mestre, and S. Woerner, "Credit risk analysis using quantum computers," *IEEE Transactions on Computers*, vol. 70, no. 12, pp. 2136–2145, Dec 2021. [Online]. Available: <https://doi.org/10.1109/TC.2020.3038063>
- [15] N. Stamatopoulos, G. Mazzola, S. Woerner, and W. J. Zeng, "Towards quantum advantage in financial market risk using quantum gradient algorithms," *Quantum*, vol. 6, p. 770, 2022. [Online]. Available: <http://dx.doi.org/10.22331/q-2022-07-20-770>
- [16] T. Matsakos and S. Nield, "Quantum Monte Carlo simulations for financial risk analytics: scenario generation for equity, rate, and credit risk factors," p. 1306, Apr 2024. [Online]. Available: <http://dx.doi.org/10.22331/q-2024-04-04-1306>
- [17] N. Stamatopoulos, D. J. Egger, Y. Sun, C. Zoufal, R. Iten, N. Shen, and S. Woerner, "Option pricing using quantum computers," *Quantum*, vol. 4, p. 291, 2020. [Online]. Available: <http://dx.doi.org/10.22331/q-2020-07-06-291>
- [18] S. Chakrabarti, R. Krishnakumar, G. Mazzola, N. Stamatopoulos, S. Woerner, and W. J. Zeng, "A threshold for quantum advantage in derivative pricing," *Quantum*, vol. 5, p. 463, 2021. [Online]. Available: <https://doi.org/10.22331/q-2021-06-01-463>
- [19] I. Y. Akhalwaya, A. Connolly, R. Guichard, S. Herbert, C. Kargi, A. Krajenbrink, M. Lubasch, C. M. Keever, J. Sorci, M. Spranger *et al.*, "A modular engine for quantum Monte Carlo integration," *arXiv preprint arXiv:2308.06081*, 2023. [Online]. Available: <https://doi.org/10.48550/arXiv.2308.06081>
- [20] M. Giron, G. Korpas, W. Parvaiz, P. Malik, and J. Aspman, "Approaching collateral optimization for NISQ and quantum-inspired computing," *IEEE Transactions on Quantum Engineering*, p. 1–18, 2023. [Online]. Available: <http://dx.doi.org/10.1109/TQE.2023.3314839>
- [21] C. Ryan-Anderson, J. G. Bohnet, K. Lee, D. Gresh, A. Hankin, J. P. Gaebler, D. Francois, A. Chernoguzov, D. Lucchetti, N. C. Brown, T. M. Gatterman, S. K. Halit, K. Gilmore, J. A. Gerber, B. Neyenhuis, D. Hayes, and R. P. Stutz, "Realization of real-time fault-tolerant quantum error correction," *Phys. Rev. X*, vol. 11, p. 041058, Dec 2021. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevX.11.041058>
- [22] D. Bluvstein, S. J. Evered, A. A. Geim, S. H. Li, H. Zhou, T. Manovitz, S. Ebadi, M. Cain, M. Kalinowski, D. Hangleiter, J. P. B. Ataides, N. Maskara, I. Cong, X. Gao, P. S. Rodriguez, T. Karolyshyn, G. Semeghini, M. J. Gullans, M. Greiner, V. Vuletić, and M. D. Lukin, "Logical quantum processor based on reconfigurable atom arrays," *Nature*, Dec 2023. [Online]. Available: <http://dx.doi.org/10.1038/s41586-023-06927-3>
- [23] D. Ruiz, J. Guillaud, A. Leverrier, M. Mirrahimi, and C. Vuillot, "LDPC-cat codes for low-overhead quantum computing in 2D," *arXiv preprint arXiv:2401.09541*, 2024. [Online]. Available: <https://doi.org/10.48550/arXiv.2401.09541>
- [24] M. da Silva, C. Ryan-Anderson, J. Bello-Rivas, A. Chernoguzov, J. Dreiling, C. Foltz, J. Gaebler, T. Gatterman, D. Hayes, N. Hewitt *et al.*, "Demonstration of logical qubits and repeated error correction with better-than-physical error rates," *arXiv:2404.02280*, 2024. [Online]. Available: <https://arxiv.org/abs/2404.02280>
- [25] J. Preskill, "Quantum computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, 2018. [Online]. Available: <http://dx.doi.org/10.22331/q-2018-08-06-79>
- [26] "Quantum technology use cases as fuel for value in finance — mckinsey.com," <https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/tech-forward/quantum-technology-use-cases-as-fuel-for-value-in-finance>, [Accessed 31-05-2024].

- [27] J. A. Montanez-Barrera and K. Michelsen, "Towards a universal QAOA protocol: Evidence of quantum advantage in solving combinatorial optimization problems," 2024. [Online]. Available: <https://arxiv.org/abs/2405.09169>
- [28] J. Y. Campbell, A. W. Lo, A. C. MacKinlay, and R. F. Whitelaw, "The econometrics of financial markets," *Macroeconomic Dynamics*, vol. 2, no. 4, pp. 559–562, 1998. [Online]. Available: <http://dx.doi.org/10.1017/S1365100598009092>
- [29] P. Glasserman, *Monte Carlo Methods in Financial Engineering*. Springer New York, 2003. [Online]. Available: <http://dx.doi.org/10.1007/978-0-387-21617-1>
- [30] M. B. Giles and L. Szpruch, "Multilevel Monte Carlo methods for applications in finance," *High-Performance Computing in Finance*, pp. 197–247, 2018. [Online]. Available: https://www.worldscientific.com/doi/abs/10.1142/9789814436434_0001
- [31] S. Sahni, "Computationally related problems," *SIAM Journal on Computing*, vol. 3, no. 4, p. 262–279, Dec 1974. [Online]. Available: <http://dx.doi.org/10.1137/0203021>
- [32] P. M. Pardalos and S. A. Vavasis, "Quadratic programming with one negative eigenvalue is np-hard," *Journal of Global Optimization*, vol. 1, no. 1, p. 15–22, 1991. [Online]. Available: <http://dx.doi.org/10.1007/BF00120662>
- [33] A. Ben-Tal and A. Nemirovski, "Robust convex optimization," *Mathematics of Operations Research*, vol. 23, no. 4, pp. 769–805, 1998. [Online]. Available: <http://www.jstor.org/stable/3690632>
- [34] P. J. S. De Brouwer, "Maslowian portfolio theory: An alternative formulation of the behavioural portfolio theory," *Journal of Asset Management*, vol. 9, no. 6, pp. 359–365, Feb 2009. [Online]. Available: <http://dx.doi.org/10.1057/jam.2008.35>
- [35] ———, *Maslowian Portfolio Theory, a Coherent Approach to Strategic Asset Allocation*. Brussels: VUBPress, 2012.
- [36] ———, "Proposal for a practical implementation of Maslowian portfolio theory," *Problemy Zarządzania*, vol. 14, no. 4 (63) t.1, pp. 39–56, 2016. [Online]. Available: <https://bibliotekanauki.pl/articles/525428>
- [37] A. Nemirovski and A. Shapiro, "Convex approximations of chance constrained programs," *SIAM Journal on Optimization*, vol. 17, no. 4, p. 969–996, Jan 2007. [Online]. Available: <http://dx.doi.org/10.1137/050622328>
- [38] A. Shapiro, D. Dentcheva, and A. Ruszczyński, *Lectures on stochastic programming*, 3rd ed., ser. MOS-SIAM Series on Optimization. New York, NY: Society for Industrial & Applied Mathematics, Dec 2021. [Online]. Available: <https://doi.org/10.1137/1.9781611976595>
- [39] Q.-M. Shao, H. Yu, and J. Yu, "Do stock returns follow a finite variance distribution?" *Annals of Economics and Finance*, vol. 2, p. 467, 2001. [Online]. Available: https://ink.library.smu.edu.sg/soe_research/526
- [40] A. M. Law and M. G. McComas, "Simulation-based optimization," in *Proceedings of the Winter Simulation Conference*, vol. 1. IEEE, 2002, pp. 41–44. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/1172866>
- [41] A. Gosavi, *Simulation-Based Optimization: Parametric Optimization Techniques and Reinforcement Learning*. Springer US, 2015. [Online]. Available: <http://dx.doi.org/10.1007/978-1-4899-7491-4>
- [42] T. Homem-de Mello and G. Bayraksan, "Monte Carlo sampling-based methods for stochastic optimization," *Surveys in Operations Research and Management Science*, vol. 19, no. 1, pp. 56–85, 2014. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1876735414000038>
- [43] J. Gacon, C. Zoufal, and S. Woerner, "Quantum-enhanced simulation-based optimization," in *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE, 2020, pp. 47–55. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9259950>
- [44] A. Gosavi, *Simulation-Based Optimization*. Springer US, 2003. [Online]. Available: <http://dx.doi.org/10.1007/978-1-4757-3766-0>
- [45] J. Adachi and A. Gupta, "Simulation-based parametric optimization for long-term asset allocation using behavioral utilities," *Applied Mathematical Modelling*, vol. 29, no. 4, p. 309–320, Apr. 2005. [Online]. Available: <http://dx.doi.org/10.1016/j.apm.2004.07.011>
- [46] C. Bao, G. Lee, and Z. Zhu, "A simulation-based portfolio optimization approach with least squares learning," in *Proceedings of the World Congress on Engineering*, vol. 2, jul 2014. [Online]. Available: https://www.iaeng.org/publication/WCE2014/WCE2014_pp905-910.pdf
- [47] S. Chakrabarti, R. Krishnakumar, G. Mazzola, N. Stamatopoulos, S. Woerner, and W. J. Zeng, "A threshold for quantum advantage in derivative pricing," *Quantum*, vol. 5, p. 463, Jun 2021. [Online]. Available: <https://doi.org/10.22331/q-2021-06-01-463>
- [48] Y. Suzuki, S. Uno, R. Raymond, T. Tanaka, T. Onodera, and N. Yamamoto, "Amplitude estimation without phase estimation," *Quantum Information Processing*, vol. 19, pp. 1–17, 2020. [Online]. Available: <http://dx.doi.org/10.1007/s11128-019-2565-2>
- [49] A. Callison and D. E. Browne, "Improved maximum-likelihood quantum amplitude estimation," *arXiv preprint arXiv:2209.03321*, 2022. [Online]. Available: <https://doi.org/10.48550/arXiv.2209.03321>

- [50] S. Aaronson and P. Rall, “Quantum approximate counting, simplified,” in *Symposium on simplicity in algorithms*. SIAM, 2020, pp. 24–32. [Online]. Available: <http://dx.doi.org/10.1137/1.9781611976014.5>
- [51] D. Grinko, J. Gacon, C. Zoufal, and S. Woerner, “Iterative quantum amplitude estimation,” *npj Quantum Information*, vol. 7, no. 1, Mar 2021. [Online]. Available: <http://dx.doi.org/10.1038/s41534-021-00379-1>
- [52] F. Labib, B. D. Clader, N. Stamatopoulos, and W. J. Zeng, “Quantum amplitude estimation from classical signal processing,” *arXiv preprint arXiv:2405.14697*, 2024. [Online]. Available: <https://doi.org/10.48550/arXiv.2405.14697>
- [53] T. Giurgica-Tiron, Y. Hindy, R. LaRose, A. Mari, and W. J. Zeng, “Digital zero noise extrapolation for quantum error mitigation,” in *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE, Oct 2020. [Online]. Available: <http://dx.doi.org/10.1109/QCE49297.2020.00045>
- [54] S. Herbert, I. Willians, R. Guichard, and D. Ng, “Noise-aware quantum amplitude estimation,” *arXiv preprint arXiv:2109.04840*, 2021. [Online]. Available: <https://doi.org/10.48550/arXiv.2109.04840>
- [55] Y. R. Sanders, J. J. Wallman, and B. C. Sanders, “Bounding quantum gate error rate based on reported average fidelity,” *New Journal of Physics*, vol. 18, no. 1, p. 012002, Dec 2015. [Online]. Available: <https://dx.doi.org/10.1088/1367-2630/18/1/012002>
- [56] M. Plesch and Č. Brukner, “Quantum-state preparation with universal gate decompositions,” *Phys. Rev. A*, vol. 83, no. 3, p. 032302, 2011. [Online]. Available: <http://dx.doi.org/10.1103/PhysRevA.83.032302>
- [57] Y. R. Sanders, G. H. Low, A. Scherer, and D. W. Berry, “Black-box quantum state preparation without arithmetic,” *Phys. Rev. Lett.*, vol. 122, no. 2, Jan 2019. [Online]. Available: <https://dx.doi.org/10.1103/PhysRevLett.122.020502>
- [58] C. Zoufal, A. Lucchi, and S. Woerner, “Quantum generative adversarial networks for learning and loading random distributions,” *npj Quantum Information*, vol. 5, no. 1, p. 103, 2019. [Online]. Available: <https://doi.org/10.1038%2Fs41534-019-0223-2>
- [59] X.-M. Zhang, T. Li, and X. Yuan, “Quantum state preparation with optimal circuit depth: Implementations and applications,” *Phys. Rev. Lett.*, vol. 129, no. 23, p. 230504, 2022. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.129.230504>
- [60] J. Bausch, “Fast black-box quantum state preparation,” *Quantum*, vol. 6, p. 773, 2022. [Online]. Available: <https://doi.org/10.22331/q-2022-08-04-773>
- [61] A. G. Rattew and B. Koczor, “Preparing arbitrary continuous functions in quantum registers with logarithmic complexity,” 2022. [Online]. Available: <https://arxiv.org/abs/2205.00519>
- [62] A. Wodecki, J. Marecek, V. Kungurtsev, P. Eichler, G. Korpas, and P. Intallura, “Spectral methods for quantum optimal control: Artificial boundary conditions,” *arXiv preprint arXiv:2403.14436*, 2024. [Online]. Available: <https://arxiv.org/abs/2403.14436>
- [63] M. Rosenkranz, E. Brunner, G. Marin-Sanchez, N. Fitzpatrick, S. Dilkes, Y. Tang, Y. Kikuchi, and M. Benedetti, “Quantum state preparation for multivariate functions,” *arXiv preprint arXiv:2405.21058*, 2024. [Online]. Available: <https://arxiv.org/abs/2405.21058>
- [64] G. Mazzola, “Sampling, rates, and reaction currents through reverse stochastic quantization on quantum computers,” *Phys. Rev. A*, vol. 104, no. 2, p. 022431, 2021. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.104.022431>
- [65] M. G. Jackson, “Quantum computing state preparation from supersymmetric quantum mechanics,” *in preparation*, 2024.
- [66] B. C. on Banking Supervision, “Basel iii: A global regulatory framework for more resilient banks and banking systems.” *Document, Bank for International Settlements, Basel, Switzerland*, 2010. [Online]. Available: https://www.bis.org/publ/bcbs189_dec2010.htm
- [67] L. J. Hong, Z. Hu, and G. Liu, “Monte Carlo methods for value-at-risk and conditional value-at-risk: A review,” *ACM Trans. Model. Comput. Simul.*, vol. 24, no. 4, Nov 2014. [Online]. Available: <https://doi.org/10.1145/2661631>
- [68] H. Mausser and D. Rosen, “Beyond VaR: from measuring risk to managing risk,” in *Proceedings of the IEEE/IAFE 1999 Conference on Computational Intelligence for Financial Engineering (CIFER) (IEEE Cat. No.99TH8408)*, 1999, pp. 163–178. [Online]. Available: <https://doi.org/10.1109/CIFER.1999.771115>
- [69] R. McKay and T. E. Keefer, “VaR is a dangerous technique,” *Corporate Finance Searching for Systems Integration Supplement*, vol. 9, p. 30, 1996. [Online]. Available: <https://ralphmckay.wordpress.com/1996/08/03/>
- [70] J. Kisiala, “Conditional value-at-risk: Theory and applications,” *arXiv preprint arXiv:1511.00140*, 2015. [Online]. Available: <https://doi.org/10.48550/arXiv.1511.00140>
- [71] “DIRECTIVE 2009/138/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on the taking-up and pursuit of the business of insurance and reinsurance (Solvency II),” *Official Journal of the European Union*, 2009. [Online]. Available: <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32009L0138>

- [72] R. T. Rockafellar and S. Uryasev, "Conditional value-at-risk for general loss distributions," *Journal of banking & finance*, vol. 26, no. 7, pp. 1443–1471, 2002. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0378426602002716>
- [73] R. T. Rockafellar, S. Uryasev *et al.*, "Optimization of conditional value-at-risk," *Journal of risk*, vol. 2, pp. 21–42, 2000. [Online]. Available: <https://www.risk.net/journal-risk/2161159/optimization-conditional-value-risk>
- [74] A. D. Roy, "Safety first and the holding of assets," *Econometrica*, vol. 20, no. 3, pp. 431–449, 1952. [Online]. Available: <http://www.jstor.org/stable/1907413>
- [75] R. T. Rockafellar and S. Uryasev, "Optimization of conditional value-at-risk," *Journal of Risk*, vol. 3, pp. 21–41, 2000. [Online]. Available: <https://api.semanticscholar.org/CorpusID:854622>
- [76] P. Krokmal, t. Uryasev, and J. Palmquist, "Portfolio optimization with conditional value-at-risk objective and constraints," *The Journal of Risk*, vol. 4, no. 2, p. 43–68, Mar 2001. [Online]. Available: <http://dx.doi.org/10.21314/JOR.2002.057>
- [77] P. T. Boggs and J. W. Tolle, "Sequential quadratic programming," *Acta numerica*, vol. 4, pp. 1–51, 1995. [Online]. Available: <https://doi.org/10.1017/S0962492900002518>
- [78] J. Nocedal and S. J. Wright, "Quadratic programming," *Numerical optimization*, pp. 448–492, 2006. [Online]. Available: <https://doi.org/10.1007/978-0-387-40065-5>
- [79] J. Zhang, "Modern Monte Carlo methods for efficient uncertainty quantification and propagation: A survey," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 13, no. 5, p. e1539, 2021. [Online]. Available: <https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wics.1539>
- [80] P. J. S. De Brouwer and F. Van den Spiegel, "The fallacy of large numbers revisited: the construction of a utility function that leads to the acceptance of two games while one is rejected," *Journal of Asset Management*, vol. 1, no. 3, pp. 257–266, Jan 2001. [Online]. Available: <https://doi.org/10.1057/palgrave.jam.2240020>
- [81] Y. Suzuki, Y. Kawase, Y. Masumura, Y. Hiraga, M. Nakadai, J. Chen, K. M. Nakanishi, K. Mitarai, R. Imai, S. Tamiya, T. Yamamoto, T. Yan, T. Kawakubo, Y. O. Nakagawa, Y. Ibe, Y. Zhang, H. Yamashita, H. Yoshimura, A. Hayashi, and K. Fujii, "Qulacs: a fast and versatile quantum circuit simulator for research purpose," *Quantum*, vol. 5, p. 559, Oct 2021. [Online]. Available: <http://dx.doi.org/10.22331/q-2021-10-06-559>
- [82] "Quantinuum spec sheet parameters," <https://github.com/CQCL/quantinuum-hardware-specifications/blob/main/notebooks/Spec%20sheet%20parameters.csv>, accessed: 2024-07-29.
- [83] J. Lehoczky, *Distributions, Statistical: Approximations*. Elsevier, 2001, p. 3784–3787. [Online]. Available: <http://dx.doi.org/10.1016/B0-08-043076-7/00406-X>
- [84] T. Fukunaga and K. Umeno, "Universal lévy's stable law of stock market and its characterization," *arXiv preprint arXiv:1709.06279*, 2018. [Online]. Available: <https://arxiv.org/abs/1709.06279>
- [85] D. Krezolek, "The application of alpha-stable distributions in portfolio selection problem - the case of metal market," *Studia Ekonomiczne*, no. 247, p. 57–68, 2015. [Online]. Available: <https://cejsh.icm.edu.pl/cejsh/element/bwmeta1.element.cejsh-8b05cdd9-e12e-4714-a722-138b755b1a50>
- [86] J.-P. Bouchaud and M. Potters, *Theory of Financial Risk and Derivative Pricing: From Statistical Physics to Risk Management*, 2nd ed. Cambridge University Press, 2003. [Online]. Available: <https://doi.org/10.1017/CBO9780511753893>
- [87] L. Devroye, *Non-uniform random variate generation*, 1986th ed. New York, NY: Springer, Apr 1986. [Online]. Available: <https://doi.org/10.1007/978-1-4613-8643-8>
- [88] F. Brück, "Generative neural networks for characteristic functions," *arXiv preprint arXiv:2401.04778*, 2024. [Online]. Available: <https://doi.org/10.48550/arXiv.2401.04778>
- [89] N. L. Johnson, S. Kotz, and N. Balakrishnan, *Continuous univariate distributions, volume 2*, ser. Wiley Series in Probability and Statistics. John wiley & sons, 1995, vol. 289. [Online]. Available: <https://books.google.co.uk/books?id=BTANEAAQBAJ>
- [90] S. Nadarajah, "A generalized normal distribution," *Journal of Applied Statistics*, vol. 32, no. 7, p. 685–694, Sep 2005. [Online]. Available: <http://dx.doi.org/10.1080/02664760500079464>
- [91] J. P. Nolan, *Modeling with Stable Distributions*. Springer International Publishing, 2020, p. 25–52. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-52915-4_2
- [92] J. Elmerraji, "Optimal cvar portfolio construction under power law stochastic walks," *SSRN Electronic Journal*, 2021. [Online]. Available: <http://dx.doi.org/10.2139/ssrn.3959932>
- [93] X.-L. Gong and X. Xiong, "Multi-objective portfolio optimization under tempered stable Lévy distribution with Copula dependence," *Finance Research Letters*, vol. 38, p. 101506, Jan 2021. [Online]. Available: <http://dx.doi.org/10.1016/j.frl.2020.101506>
- [94] A. Papapantoleon, "An introduction to Lévy processes with applications in finance," *arXiv preprint arXiv:0804.0482*, 2008. [Online]. Available: <https://doi.org/10.48550/arXiv.0804.048>
- [95] A. Chekhlov, S. Uryasev, and M. Zabarankin, *Portfolio optimization with drawdown constraints*. World Scientific, Jan 2004, p. 209–228. [Online]. Available: http://dx.doi.org/10.1142/9789812562586_0013

- [96] A. Chekhlov, S. P. Uryasev, and M. Zabarankin, “Portfolio optimization with drawdown constraints,” *SSRN Electronic Journal*, 2000. [Online]. Available: <http://dx.doi.org/10.2139/ssrn.223323>

APPENDIX A. ANALYTICAL DERIVATION OF THE UTILITY FUNCTION AND MINIMUM FOR THE MEAN-VAR PORTFOLIO PROBLEM

We derive in this Section the analytical expression of the utility function to be minimised and its analytical minimum for the Mean-Var portfolio problem discussed in Section 7.2.

A.1. Simplifying the utility function. Starting from (7.6) and substituting (7.2), we have

$$\begin{aligned} U(r_{T+1}, \omega_T, R_T^{(f)}, \lambda) &= \mathbb{E} \left[V_T \times \left(\omega_T e^{r_{T+1} + R_T^{(f)}} + \tilde{\omega}_T e^{R_T^{(f)}} \right) \right] - \lambda \left(\mathbb{E} \left[V_T^2 \times \left(\omega_T e^{r_{T+1} + R_T^{(f)}} + \tilde{\omega}_T e^{R_T^{(f)}} \right)^2 \right] \right. \\ &\quad \left. - \mathbb{E} \left[V_T \times \left(\omega_T e^{r_{T+1} + R_T^{(f)}} + \tilde{\omega}_T e^{R_T^{(f)}} \right) \right]^2 \right). \end{aligned} \quad (\text{A.1})$$

Given that we can take $R_T^{(f)}$ as a constant for a single time period, this can be written as the following integral to be optimised

$$\begin{aligned} -U(r_{T+1}, \omega_T, R_T^{(f)}, \lambda) &= - \left[V_T \omega_T e^{R_T^{(f)}} \int_{\mathbb{R}} e^{r_{T+1}} p(r_{T+1}) dr_{T+1} + V_T \tilde{\omega}_T e^{R_T^{(f)}} \right] \\ &\quad + \lambda V_T^2 \omega_T^2 e^{2R_T^{(f)}} \left(\int_{\mathbb{R}} e^{2r_{T+1}} p(r_{T+1}) dr_{T+1} - \left(\int_{\mathbb{R}} e^{r_{T+1}} p(r_{T+1}) dr_{T+1} \right)^2 \right). \end{aligned} \quad (\text{A.2})$$

To evaluate this utility function we thus just need to estimate the following integrals

$$\mathcal{I}_1 = \int_{\mathbb{R}} e^{r_{T+1}} p(r_{T+1}) dr_{T+1}, \quad \mathcal{I}_2 = \int_{\mathbb{R}} e^{2r_{T+1}} p(r_{T+1}) dr_{T+1}, \quad (\text{A.3})$$

using MCI. Alternatively, if we take the variable of integration (i.e., the random variable) as $\varrho_{t+1} = e^{r_{T+1}}$ instead, then we can write the utility to calculate using MCI as

$$\begin{aligned} -U(\varrho_{T+1}, \omega_T, R_T^{(f)}, \lambda) &= -V_T \omega_T e^{R_T^{(f)}} \int_{\mathbb{R}} \varrho_{t+1} p(\varrho_{t+1}) d\varrho_{t+1} + V_T \tilde{\omega}_T e^{R_T^{(f)}} \\ &\quad + \lambda V_T^2 \omega_T^2 e^{2R_T^{(f)}} \left(\int_{\mathbb{R}} (\varrho_{t+1})^2 p(\varrho_{t+1}) d\varrho_{t+1} - \left(\int_{\mathbb{R}} \varrho_{t+1} p(\varrho_{t+1}) d\varrho_{t+1} \right)^2 \right), \end{aligned} \quad (\text{A.4})$$

which amounts to calculating the mean and variance of ϱ_{t+1} using MCI. To evaluate this utility function we thus just need to estimate the following integrals

$$\mathcal{I}_3 = \int_{\mathbb{R}} \varrho_{t+1} p(\varrho_{t+1}) d\varrho_{t+1}, \quad \mathcal{I}_4 = \int_{\mathbb{R}} \varrho_{t+1}^2 p(\varrho_{t+1}) d\varrho_{t+1}, \quad (\text{A.5})$$

using MCI. Due to the optimisation being found to be more stable based on the approach of modelling the random variable as ϱ_{t+1} ,¹¹ this is what is thus adopted.

¹¹It is more stable because the pre-factors in front of the various integrals to compute are generally smaller, so that fluctuations in the output of the estimation of the integrals are “amplified” less in the final calculation. In practice this amounts to a ‘smoother’ optimisation trajectory curve.

A.2. Optimising the utility function for a Gaussian random variable. To find the analytical value for the optimum we start from the expression obtained in (A.2). If we take returns as being Gaussian distributed $r_{T+1} \sim \mathcal{N}(\mu, \sigma^2)$ then this expression simplifies to

$$\begin{aligned} -U(r_{T+1}, \omega_T, R_T^{(f)}, \lambda) = & - \left[V_T \omega_T e^{R_T^{(f)}} e^{\mu + \frac{\sigma^2}{2}} + V_T \tilde{\omega}_T e^{R_T^{(f)}} \right] \\ & + \lambda \left[V_T^2 \omega_T^2 e^{2R_T^{(f)}} e^{2(\mu + \sigma^2)} + 2V_T^2 \omega_T \tilde{\omega}_T e^{2R_T^{(f)}} e^{\mu + \frac{\sigma^2}{2}} + V_T^2 \tilde{\omega}_T^2 e^{2R_T^{(f)}} - \left(V_T \omega_T e^{R_T^{(f)}} e^{\mu + \frac{\sigma^2}{2}} + V_T \tilde{\omega}_T e^{R_T^{(f)}} \right)^2 \right], \end{aligned} \quad (\text{A.6})$$

Then differentiating we have

$$\begin{aligned} \frac{\partial(-U)}{\partial \omega_T} = & -V_T e^{R_T^{(f)}} e^{\mu + \frac{\sigma^2}{2}} + V_T e^{R_T^{(f)}} \\ & + \lambda \left[2e^{2R_T^{(f)}} e^{2(\mu + \sigma^2)} V_T^2 \omega_T + 2e^{2R_T^{(f)}} e^{\mu + \frac{\sigma^2}{2}} V_T^2 (1 - \omega_T) \right. \\ & - 2e^{2R_T^{(f)}} V_T^2 (1 - \omega_T) - 2e^{R_T^{(f)}} e^{\mu + \frac{\sigma^2}{2}} V_T^2 \omega_T \\ & \left. - 2 \left(-e^{R_T^{(f)}} V_T + e^{R_T^{(f)}} e^{\mu + \frac{\sigma^2}{2}} V_T \right) \left(e^{R_T^{(f)}} V_T (1 - \omega_T) + e^{R_T^{(f)}} e^{\mu + \frac{\sigma^2}{2}} V_T \omega_T \right) \right], \end{aligned} \quad (\text{A.7})$$

Setting this equal to zero and solving gives the minimum as

$$\hat{\omega}_T = \frac{e^{-R_T^{(f)} - 2\mu - \sigma^2} (e^{\mu + \frac{\sigma^2}{2}} - 1)}{2V_T \lambda (e^{\sigma^2} - 1)}. \quad (\text{A.8})$$

We know however that when we are actually evaluating the minimum using the QMCI engine, as we only have a limited number of qubits available – which we denote as n_q – we are not actually calculating the integrals based on a pseudo-continuous distribution (as one could effectively do classically using as many bits as required for precision) but are instead calculating using a discrete approximation to the distribution. Thus if we would like to evaluate the performance of the optimisation using the QMCI engine, we should compare to a discretised version of the problem.

A.3. Optimising the utility function using a discrete random variable. If we model this discrete distribution over $N = 2^{n_q}$ points as a series of weighted delta functions treated as a continuous PDF such that

$$p(r_{T+1}) \approx \sum_{i=0}^{N-1} \omega^i(\mu, \sigma^2) \delta(r_{T+1} - r_{T+1}^i), \quad (\text{A.9})$$

then we thus have

$$\begin{aligned} \mathcal{I}_1^{\text{disc}} &= \int_{\mathbb{R}} e^{r_{T+1}} \sum_{i=0}^{N-1} \omega^i(\mu, \sigma^2) \delta(r_{T+1} - r_{T+1}^i) dr_{T+1} = \sum_{i=0}^{N-1} \omega^i(\mu, \sigma^2) e^{r_{T+1}^i}, \\ \mathcal{I}_2^{\text{disc}} &= \int_{\mathbb{R}} e^{2r_{T+1}} \sum_{i=0}^{N-1} \omega^i(\mu, \sigma^2) \delta(r_{T+1} - r_{T+1}^i) dr_{T+1} = \sum_{i=0}^{N-1} \omega^i(\mu, \sigma^2) e^{2r_{T+1}^i}, \end{aligned} \quad (\text{A.10})$$

and then following the same methodology as earlier, we get the discrete approximation to the minimum as

$$\hat{\omega}_T^{\text{disc}} = -\frac{e^{-R_T^{(f)}} \left[\sum_{i=0}^{N-1} \omega^i(\mu, \sigma^2) e^{r_{T+1}^i} - 1 \right]}{2V_T \lambda \left[\left(\sum_{i=0}^{N-1} \omega^i(\mu, \sigma^2) e^{r_{T+1}^i} \right)^2 - \sum_{i=0}^{N-1} \omega^i(\mu, \sigma^2) e^{2r_{T+1}^i} \right]}. \quad (\text{A.11})$$

APPENDIX B. STABLE DISTRIBUTIONS AND EXTENDED INTERESTING DISTRIBUTIONS

A number of works in the quantum finance literature have focused on examples involving Gaussian random variables. As an effort to bring quantum finance closer to more realistic problems, we discuss here stable distributions and other interesting distributions which could be investigated in future studies.

B.1. Stable distributions. In formulation (6.1) for the Mean-CVaR optimisation problem, the return of the portfolio R is a random variable. In Markowitz theory, it is traditionally assumed that the returns follow a Gaussian or log-normal distribution. In practice, returns have fat tails and are not distributed as such [83, 84], hence one needs to investigate more general distributions. An interesting family of distributions are the stable distributions which aim to verify the following definition

Definition B.1 (Stability of a random variable). *Consider a random variable X and a set $\{X_1, \dots, X_n\}$ where each X_i are i.i.d. copies of X . Then, X is said to be stable iff for each $n > 1$, there exists scaling factor $c_n > 0$ and a shift factor d_n such that*

$$\sum_{i=1}^n X_i = c_n X + d_n. \quad (\text{B.1})$$

Returns of financial assets inherently have the property of stability since the return of asset i at time $[t_j, t_{j+T}]$, is given as

$$R_i(t_j) = \frac{V(t_{j+T}) - V(t_j)}{V(t_j)} \approx \log V(t_{j+T}) - \log V(t_j),$$

that is continuous compounded return. Therefore

$$R_i([t_1 \rightarrow t_k]) := \sum_{\ell=1}^k R_i(t_\ell) \approx \log V(t_{k+1}) - \log V(t_1),$$

where $R_i([t_1 \rightarrow t_k])$ represents the total return from t_1 to t_{k+1} , and is equivalent to the sum of the individual returns over each time period from 1 to k .

Among the stable distributions, α -stable distributions offer the most realistic models [85] in mathematical finance. An α -stable distribution is a family of distributions depending on four parameters, specified by stability index $\alpha \in (0, 2]$, skewness β , scale γ , and location δ . Gaussian distributions form a special case where $\alpha = 2, \beta = 0$ [86]. Interestingly, empirical evidence suggests that α -stable distributions are comparatively more risk adverse. The difficulty with α -stable distributions lie in the fact that direct sampling from the characteristic function is not possible. Fourier inversion formula and Lévy's characterisation theorem are required in order to invert these distributions and obtain the density or distribution function. There exist classical methods to try to overcome this problem such as Devoye's method [87] or Brück's generative neural network [88]. It remains an open problem to extend these methods to the quantum setting.

B.2. Other interesting distributions. We now list some other distributions which have not to our knowledge been studied in the quantum literature and would be of interest to use for quantum algorithms that apply to more realistic use cases.

- **Shifted log-normal distribution.** The shifted log-normal distribution extends the Gaussian distribution via an additional parameter controlling skewness [89]. The PDF is given by

$$f(x) = \frac{1}{(x - \theta)\sigma\sqrt{2\pi}} \exp\left(-\frac{(\ln(x - \theta) - \mu)^2}{2\sigma^2}\right), \quad (\text{B.2})$$

for $x > \theta$.

- **Generalised Gaussian distribution.** The generalised Gaussian distribution extends the Gaussian distribution via an additional parameter controlling kurtosis [90].

The shifted log-normal and generalised Gaussian distributions are of particular interest in financial modeling due to their ability to model and represent non-trivial, non-Gaussian characteristics that are frequently observed in financial data. The shifted log-normal distribution skewness parameter θ allows for the modeling of asymmetric distributions with a lower bound. This is particularly relevant in financial modelling applications where certain variables, such as asset prices or returns, are constrained by a minimum value (such as a fundamental price or a regulatory requirement). By incorporating this shift, the distribution can more accurately reflect the empirical properties of the underlying data, leading to enhanced model realism and improved risk assessment.

On the other hand, the generalised Gaussian distribution extends the traditional Gaussian framework by including a parameter to control the kurtosis of the distribution. This additional flexibility enables the distribution to adapt to the leptokurtic nature of financial returns (characterised by heavy tails and sharp peaks). This is crucial since the ability to capture and model excess kurtosis is crucial for accurate risk estimation, particularly in the context of extreme events and tail-risk management. By employing the generalised Gaussian distribution, one may obtain a more precise quantification of the probabilities associated with rare, high-impact events, facilitating robust stress testing and capital allocation decisions.

APPENDIX C. ERROR ANALYSIS FOR THE LÉVY DISTRIBUTION

It is an established fact that, at least over small time-periods, the distributions of returns of financial assets are heavy-tailed compared to the Gaussian distribution [86]. This implies that the Gaussian distribution significantly under-estimates extreme returns. Since financial returns are, in essence, convolutions of returns on smaller time scales, it is sensible to consider Lévy distributions [91–93]¹² over smaller time-periods as they are stable under addition. They are able to embody characteristics of market data such as heavy tails and asymmetry which are crucial for modeling financial risks and returns over different time scales. The PDF of the Lévy distribution, denoted by $\mathcal{L}(\mu, c)$, is defined for $x \geq \mu$ where μ is the location parameter and c is the scale parameter. Its PDF is

$$\mathcal{L}(x, \mu, c) = \sqrt{\frac{c}{2\pi}} \frac{e^{-\frac{c}{2(x-\mu)}}}{(x - \mu)^{3/2}},$$

highlighting its heavy-tailed nature which results in infinite variance and, under certain conditions, infinite expectation.

As x approaches positive or negative infinity, the functional form of the Lévy distribution approximates $\frac{\mu A_\pm^\mu}{|x|^{\mu+1}}$, where $0 < \mu < 2$, with μ being the stability index or characteristic exponent, and A_\pm represents the tail amplitudes which determine the decay scale of the distribution's tails. Given

¹²and more generally stable distributions, see Appendix B

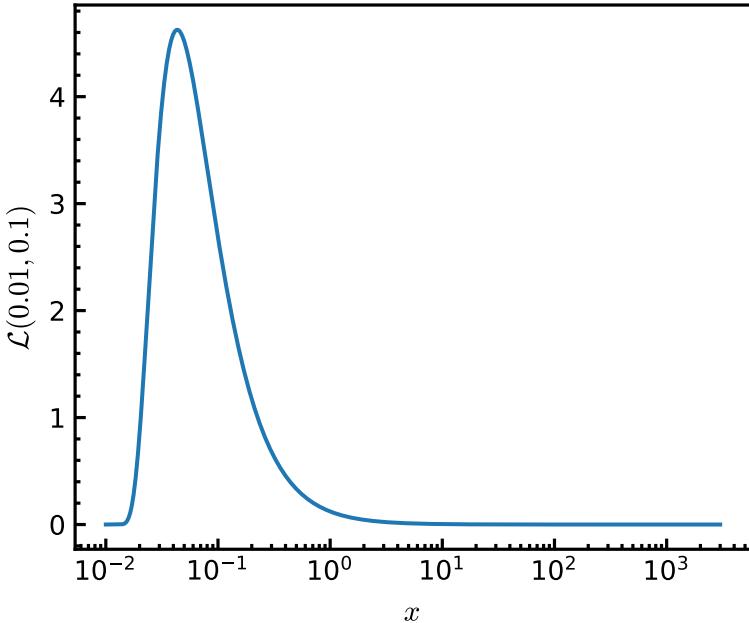


Figure 10. Trained Lévy distribution with $\mu = 0.01$ and $c = 0.1$ in the range $[0.01, 3000]$.

the inherent asymmetry, quantified by the asymmetry parameter $\beta = \frac{A_+^\mu - A_-^\mu}{A_+^\mu + A_-^\mu}$, Lévy distributions are particularly suited for modeling financial instruments or portfolios where risk assessment is critical [94]. For instance, in risk management, the heavy-tailed feature of the Lévy distribution implies that risk measures like VaR and CVaR can reach particularly high values, reflecting significant potential losses in adverse market conditions. Due to these properties, Lévy distributions are often truncated in practical applications to manage the implications of their infinite support, focusing instead on a finite range of practical interest, which simplifies both the theoretical analysis and numerical computations involved in financial risk assessments.

We subsequently study a Lévy distribution with a truncated support range $[\mu, x_u)$ and as in Section 3.3.1, the circuit representing this distribution is trained using a hardware-efficient ansatz. The trained distribution is given in Fig. 12. The aim of this Section is to investigate the impact in terms of quantum resources of implementing a distribution that is more suitable for real-life applications.

C.1. How to choose the upper bound x_u . For practical reasons (the limited number of qubits available for quantum hardware and the lack of efficient state-preparation techniques for a large number of qubits), we are restricted to using circuits with a small number of qubits. This poses an extra challenge for the Lévy distribution due to its heavy tail. To understand this, we carry out an analysis of the systematic errors, and thus we start by defining these errors.

C.1.1. Error definitions. In this Section, we provide expressions for the systematic errors when the function applied is the constant 1, which is equivalent to integrating the PDF (without a function applied). For this, we first write the truncated cumulative probability:

$$P_T = \int_{x_\ell - \Delta/2}^{x_u + \Delta/2} \mathcal{L}(\mu, c) dx, \quad (\text{C.1})$$

46

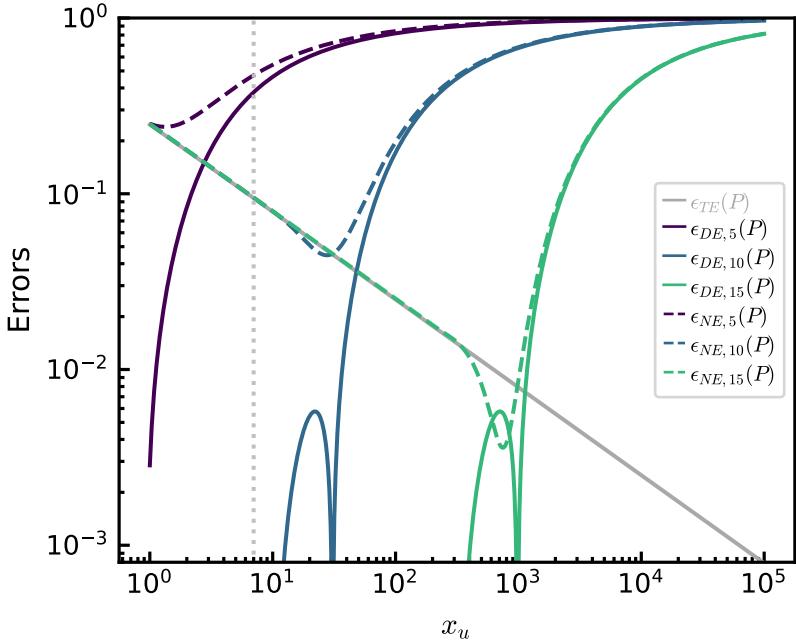


Figure 11. Truncation, discretisation, and normalisation errors for the Lévy distribution $\mathcal{L}(0.01, 0.1)$ as a function of x_u for 5-, 10-, and 15-qubit circuits. The truncation error (as expected) gets smaller with increasing x_u . The discretisation and normalisation errors get smaller as the number of qubits is increased, but increase with x_u for a fixed number of qubits. The vertical dashed silver line represents the x_u choice used for the trained circuit represented in Fig. 12.

and the truncation error:

$$\epsilon_{TE}(P) = 1 - P_T. \quad (\text{C.2})$$

The discretised probability is given by

$$P_{D,n} = \sum_{x=x_\ell}^{x_u} \mathcal{L}(x, \mu, c) \Delta, \quad (\text{C.3})$$

where 2^n is the number of terms in the summation and n is the number of qubits used to load the distribution. Finally, we write the discretisation error as

$$\epsilon_{DE,n}(P) = |P_T - P_D|, \quad (\text{C.4})$$

and the normalisation error as

$$\epsilon_{NE,n}(P) = 1 - P_{D,n}. \quad (\text{C.5})$$

C.1.2. The trade-off between truncation and discretisation errors. In Fig. 11, we present the truncation, discretisation, and normalisation errors for the Lévy distribution $\mathcal{L}(0.01, 0.1)$ as a function of x_u for 5-, 10-, and 15-qubit circuits. Figure 11 shows that we need to choose a large x_u value ($\sim 10^3$) to have $P_T \sim 0.99$ which is equal to $1 - \epsilon_{TE}(P)$. However, as we can also see in Figs. 10–11, the majority of the probability density is still in the smaller range $[\mu, 1]$. Therefore, choosing a large x_u value will lead to considerable discretisation error when we have a limited number of support points. Fig. 11 also shows the discretisation and normalisation errors when we use 2^5 , 2^{10} , and 2^{15} support points for $\mathcal{L}(0.01, 0.1)$. We can see that, in general, there is a trade-off between the

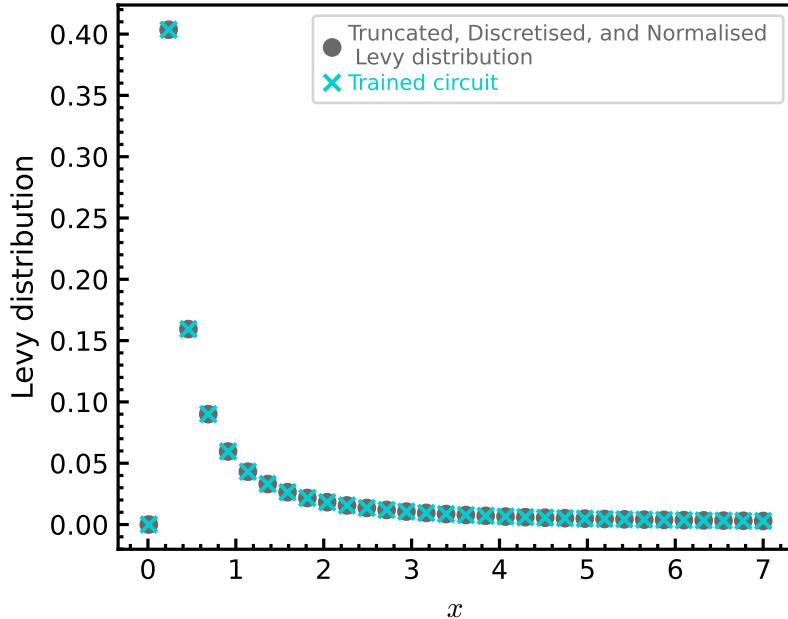


Figure 12. $\mathcal{L}(0.01, 0.1)$ distribution prepared by the trained circuit. The distribution obtained from trained circuit has a small state-preparation error and is qualitatively a good approximation ($L_\infty \sim 10^{-7}$) of the training target i.e. the truncated, discretised, and normalised Lévy distribution.

truncation and discretisation errors. For the 5-qubit case, choosing $x_u = 7$ gives us $\epsilon_T \sim 0.1$, which is shown by the vertical dashed line, and this is what we chose to train for Section C.2.

Finally, in Fig. 11, we can see that a larger number of support points lead to lower discretisation errors, as expected. Interestingly, it is not monotonically increasing, and this means one can go up to $P_T \sim 0.95$ for a small discretisation error at certain ‘‘sweet spots’’. However, training or loading a distribution with a small state-preparation error for circuits with a large number of qubits is a challenging and interesting question that is left for future study.

C.2. 5-qubit trained circuit for the Lévy distribution. Finally, for demonstration purposes, we trained a 5-qubit circuit to load the truncated, discretised, and normalised $\mathcal{L}(0.01, 0.1)$ distribution. We use the HWE ansatz. Fig. 12 shows the distribution loaded by such a circuit and the corresponding truncated, discretised, and normalised $\mathcal{L}(0.01, 0.1)$ distribution. Though the trained circuit has a small state-preparation error and is a good approximation of the training target, it is not a good approximation for the continuous distribution shown in Fig. 10 (see the silver dashed-line in Fig. 11 for the sizes of the systematic errors).

APPENDIX D. PORTFOLIO OPTIMISATION WITH CONDITIONAL-DRAWDOWN-AT-RISK

The ‘‘trade-off optimisation’’ problem, framed within a SBO portfolio-optimisation framework, seeks to balance the return expectations against risk considerations, quantified through the Conditional-Drawdown-at-Risk (CDaR) measure [95]. The problem can be formulated as

$$\underset{w \in \mathcal{W}}{\text{maximize}} \quad \mathbb{E}[X] - \lambda \text{CDaR}(w). \quad (\text{D.1})$$

Specifically, CDaR is defined as

$$\text{CDaR}(w, \beta) = \frac{1}{1 - \beta} \int_{D(w, r, t) \geq \alpha(w)} D(w, r, t) p(r(t)) dr(t),$$

where $w \in \mathcal{W} \subset \mathbb{R}^n$ is the vector of portfolio weights, r is the vector of cumulative asset returns (daily), with probability mass function $p(r(t))$. Furthermore, $D(w, r, t) = \max_{\tau < t} (w^T r(\tau) - w^T r(t))$ is the drawdown of the portfolio, α is the portfolio drawdown (DaR) with confidence β and $0 \leq \tau \leq t$.

Here, the objective function is crafted to maximise the expected returns $\mathbb{E}[X]$, adjusted by a risk aversion parameter λ that scales the impact of the CDaR. The CDaR, is a risk measure that evaluates the worst expected loss under normal market conditions over a specified period within a given confidence interval and the risk aversion parameter λ is a scalar that quantifies the decision-maker's risk tolerance, influencing the trade-off between risk and return [96]. Other risk measures, for example Entropic-Value-at-Risk can also be considered for our approach. Nevertheless, we leave this for future work.