Quantum Information Processing (2023) 22:51
https://doi.org/10.1007/s11128-022-03777-2

Quantum computing for financial risk measurement

Sascha Wilkens[1] · **Joe Moorhouse**[2]

Received: 12 February 2022 / Accepted: 25 November 2022 / Published online: 18 January 2023

Abstract

Quantum computing allows a significant speed-up over traditional CPU- and GPU-based algorithms when applied to particular mathematical challenges such as optimisation and simulation. Despite promising advances and extensive research in hard- and software developments, currently available quantum systems are still largely limited in their capability. In line with this, practical applications in quantitative finance are still in their infancy. This paper analyses requirements and concrete approaches for the application to risk management in a financial institution. On the examples of Value-at-Risk for market risk and Potential Future Exposure for counterparty credit risk, the main contribution lies in going beyond textbook illustrations and instead exploring must-have model features and their quantum implementations. While conceptual solutions and small-scale circuits are feasible at this stage, the leap needed for real-life applications is still significant. In order to build a usable risk measurement system, the hardware capacity—measured in number of qubits—would need to increase by several magnitudes from their current value of about 10^2. Quantum noise poses an additional challenge, and research into its control and mitigation would need to advance in order to render risk measurement applications deployable in practice. Overall, given the maturity of established classical simulation-based approaches that allow risk computations in reasonable time and with sufficient accuracy, the business case for a move to quantum solutions is not very strong at this point.

The views expressed in this paper are those of the authors and do not necessarily reflect the views and policies of their respective employers. Special thanks to Stefan Wörner and Daniel Egger of IBM Research and Marco Paini of Rigetti Computing for helpful discussions. The authors are also grateful to Sebastian Cassel as well as three anonymous referees for their valuable inputs and suggestions.

✉ Sascha Wilkens
Wilkens@gmx.de

Joe Moorhouse
Joe.Moorhouse@gmail.com

1 London, UK

2 Oxford, UK

Keywords Quantum computing · Finance · Risk measurement · Market risk · Counterparty credit risk

JEL Classification G17 · G32

1 Introduction

The term 'quantum computing' refers to using quantum mechanics, i.e. the theory to describe physical properties and effects of atoms and subatomic particles, on specially designed devices.[1] Compared to 'classical' computations on CPU and GPU hardware,[2] the use of quantum hardware and specially designed software promises a significant speed advantage, dependent on the problem at hand. For certain types of applications, such as integer factorisation—often used as a basis for encryption—quantum computing is the only known way to obtain solutions in an acceptable timeframe.

The landscape of quantum technologies, research and investments has seen an exponential growth over recent years [55]. Financial institutions such as large banks have been investing substantially into quantum research—mainly on rented hardware with so far limited capabilities—and the exploration of its use cases. High hopes have been expressed for applications such as trading and risk management [11, 20, 22, 44, 69, 90]. The recent trend of using machine learning for financial problems offers a straightforward extension to quantum computing. It is as such vigorously pursued by banks [19]. Suitable quantum applications are especially those where a large number of algorithmic repetitions is required, for example, the brute-force enumeration of possible solutions for a system of equations and, to a lesser extent, in Monte Carlo simulation.

The contribution of this study is an exploration of the feasibility of using quantum computing in actual large-scale risk applications in financial institutions. In a typical market risk application that makes use of simulating future market scenarios—such as a Monte Carlo-based Value-at-Risk (VaR)—a large number of risk factors and their joint evolution need to be reflected. Furthermore, the corresponding changes in the present value (PV) of instruments in a portfolio require revaluations in each potential future state. The situation in a counterparty credit risk context is similar: scenarios across a multitude of risk factors and, additionally, numerous time steps need to be combined with portfolio revaluations.

The analysis shows that the capacity of current state-of-the-art quantum systems, measured through the number of 'qubits', severely limits both the number of risk factors that can be simulated as well as the accuracy of their marginal and joint distributions. In conjunction with typical 'noise' patterns that quantum circuits experience in practice, risk measurement applications with real-life use cases are out of reach for the foreseeable future.[3] Bearing in mind that practical risk applications in financial institutions using classical computing are meanwhile mature and deliver accurate enough

[1] See, for example, the introductory work in [45] and the state-of-the-art technology summary in [32].

[2] CPU: Central Processing Unit; GPU: Graphics Processing Unit.

[3] On the example of market risk, the authors of [37] come to a similar assessment.

results in acceptable amounts of time even for large-scale portfolios, the business case for a move to quantum computing is not apparent at this point.

The paper is organised as follows. Sect. 2 summarises the current state of research into quantum applications in quantitative finance. Sect. 3 defines the framework for practical quantum use cases in risk measurement. It outlines fundamental challenges in terms of size, speed and accuracy requirements and recalls some essentials on quantum computing. The basic building blocks in the form of quantum amplitude estimation as well as techniques for scenario modelling and working with profit-and-loss (P&L) distributions are described. Furthermore, quantum-specific challenges for real-life use cases such as the handling of noise are covered. Sect. 4 presents an application to market risk as typically encountered in financial services. Starting from the underlying risk factor model with its calibration and the VaR calculation, the creation of a suitable quantum circuit is described. This is followed by an analysis of the estimation results. Covering the same aspects, a worked example for measuring counterparty credit risk is discussed in Sect. 5. In Sect. 6, the influence of adding typical quantum noise as present in current devices is illustrated. The Python-based *Qiskit* framework[4] of IBM is used for the actual implementations. Section 7 concludes and provides an outlook for future research.

2 Research landscape in quantitative finance

Quantitative finance is a prime candidate for the use of quantum computing, given that many algorithms rely on simulation or on solving complex optimisation problems which either demand a vast amount of computational power or are even unsolvable with classical approaches.[5] Practical applications of quantum computing in quantitative finance are still comparatively limited.

Pricing of derivatives. Amongst financial instruments, some classes are easier to handle than others. For the purpose of 'pricing', i.e. the determination of the PV, one will typically rely on (1) replication arguments—assembling the same payoff from more basic building blocks whose PVs are known or easier to obtain—and/or (2) assumptions on the future evolution of the value drivers (e.g. stock prices) and obtain the PV through mathematical derivation or simulation. A particular challenge are derivatives, whose values depend on that of one or more underlying variables. Especially the pricing of instruments with nonlinear payoffs such as options is an established challenge.[6] In case no closed-form pricing formulas or efficient PDE-based[7] approaches are available, Monte Carlo simulation is the standard tool. Quantum approaches have been proposed for a variety of applications, for example:

[4] See https://qiskit.org.

[5] Amongst many others, [1, 12, 29] provide an overview of applications in finance, current techniques and prospects.

[6] See [47] as an industry reference on derivatives contracts and their pricing.

[7] PDE: Partial Differential Equation.

- Pricing of standard ('vanilla'), path-dependent (e.g. barrier and Asian) and multi-asset options in a Black-Scholes [9] and local volatility [28] framework [4, 18, 33, 38, 51, 54, 64, 72, 73, 75, 76, 80]
- Pricing of options under a stochastic volatility [16] and jump-diffusion process [95]
- Pricing of American-style options [27, 62]
- Pricing of interest-rate derivatives with a multi-factor model [58, 84]
- Pricing of collateralised debt obligations (CDOs) [83]

Risk measurement. In a financial context, 'risk' usually refers to an adverse event associated with a (financial) loss. Quantum applications that have been proposed so far span, for instance:

- Estimation of risk measures such as VaR (see also Sect. 4), Conditional VaR (CVaR) [91] and the corresponding risk contributions [63]
- Estimation of credit risk (Economic Capital) [30]
- Sensitivity analysis for a (business) risk model at an exchange [15]

Miscellaneous. Quantum algorithms have been applied across a range of other areas in finance as well. For example:

- Portfolio optimisation—construction of 'optimal' portfolios, for instance, in terms of the best trade-off between expected return and risk [6, 26, 46, 56, 70, 87, 88]
- Time series forecasting [31]
- (High-frequency) trading and arbitrage [21, 94]
- Credit scoring and classification [61]
- Handling of transaction settlements (i.e. the exchange of securities and cash between parties) at a clearing house [13]

Notably, techniques for estimating gradients and higher-order derivatives on quantum hardware [57, 78, 81] are highly relevant for many applications in finance such as pricing, hedging and risk management. The same is true for techniques to reflect pseudo-random numbers, as an essential building block in typical Monte Carlo use cases, in a quantum circuit [50, 65]. Crucial for risk applications is the reflection of the dependence structure amongst value drivers, usually expressed by means of copulas (see also Sect. 4). The corresponding quantum implementation [60, 93] is therefore an important component. Furthermore, Quantum Machine Learning (QML) has received considerable attention across the full range of techniques such as supervised and unsupervised learning as well as generative modelling tasks [7, 8, 71, 77]. Specifically for financial applications, methods for market scenario generation, such as the joint evolution of dependent risk factors [53], are a promising route [3, 23, 52, 97].

3 Framework for quantum computing in risk measurement

3.1 Fundamental requirements

Practical use cases in quantitative finance usually exhibit several vital challenges. One of the most important ones is the *scalability* of any application—such as the pricing

and/or risking of instruments—so that calculations with a minimum *accuracy*[8] are technically feasible within acceptable *runtimes*.

For instance, a market risk application in a large financial institution typically encounters the following 'size challenges':

- Portfolios: 1,000 to 10,000
- Instruments/trades:[9] 100,000 to 1,000,000
- Valuation parameters: 10,000 to 10,000,000

For the actual risk quantification, one would rely on:

- Scenarios (for a subset of the valuation parameters): 100 to 1,000,000
- Time horizons (not necessarily equally spaced): 1 to 1,000

In spite of the inherent model risk, accuracy of risk measures is expected to be high in practice, usually not exceeding $\pm 10\%$ around the 'true' value. Risk calculations need to fit into a few hours as a maximum (e.g. overnight batches), but, dependent on the application, outputs can be required in a matter of minutes or even seconds (e.g. for intraday risk and what-if analyses for trading decisions).

These requirements drive, amongst others, storage needs as well as the minimum processing speed of a risk system.

3.2 Essentials on quantum computing

The core idea behind quantum computing is the ability to perform parallel computations at scale. As a building block, a two-dimensional quantum-mechanical system—a *qubit*—is used, which encodes the classical '0' and '1' bits in its basis states, $|0\rangle$ and $|1\rangle$. Through *superposition* the quantum system can be in all of its states at the same time, which can be exploited from a computational angle.

In a practical application, one will usually (a) encode the input data through the initial states of the qubits, (b) bring those qubits into superposition, (c) apply an algorithm (also: *oracle*) across all states (see, for example, Sect. 3.3) and (d) measure one or several qubits. This measurement has a random result whose interpretation is to reflect the solution to the problem at hand.[10]

[8] 'Accuracy', defined as a measurement close to the true value, should ideally be accompanied by 'precision', i.e. the property that repeated measurements—ceteris paribus—lead to the same result.

[9] While the terms 'instrument' and 'trade' are often used interchangeably in practice, more precisely, there are usually many trades referring to the same instrument. The aggregation of trades might lead to an overall 'position' in one instrument. In the following, the study largely refers to instruments.

[10] See, for example, [69, 88], amongst many others. Note that quantum computations per se are entirely deterministic, i.e. given a starting state of the system, the same final state will always be reproduced. When extracting the output in 'classical terms', however, the same final quantum state may be measured as different classical states, following a probability distribution. In rare cases, the probability is one for a particular output, rendering the algorithm de facto deterministic.

3.3 Quantum amplitude estimation as a basis

Quantum Amplitude Estimation (QAE) [14] promises a speed-up in the calculation of expected values compared to 'classical' simulation. Concretely, with an operator $\mathcal{A}$ acting on a system[11],

$$|\Psi\rangle = \mathcal{A}|0\rangle = \sqrt{\beta}|\tilde{\Psi}_1\rangle|1\rangle + \sqrt{1-\beta}|\tilde{\Psi}_0\rangle|0\rangle,$$

the unknown parameter $\beta = |\langle\tilde{\Psi}_1|\tilde{\Psi}_1\rangle| \in [0, 1]$ is estimated, the amplitude of the state $|\tilde{\Psi}_1\rangle$.

Quantum phase estimation leads to an estimator $\tilde{\beta}$ with the property

$$|\beta - \tilde{\beta}| \leq \frac{\pi}{M} + \frac{\pi^2}{M^2} = \mathcal{O}(M^{-1})$$

with a probability of at least $8/\pi^2 \approx 81.1\%$ (see, for example, [29]). M is hereby the number of quantum samples. Estimating β in this way leads to a quadratic speed-up compared to classical Monte Carlo simulation with its convergence rate $\mathcal{O}(M^{-1/2})$.

The original QAE requires large circuits and is computationally expensive. Various improvements have been developed over time [16, 36, 41, 43, 66, 74, 82] and form part of industry-standard implementations by now, for example, Iterative QAE (IQAE).

3.4 Modelling future scenarios

Distributions. In the context of quantitative finance, the evolution of asset prices is regularly modelled with the help of distributions. These can either reflect the prices directly or their changes (returns) relative to fixed (= current) values. Alternatively, one can simulate the valuation parameters that—in conjunction with pricing functions—determine the future value or return distribution of the assets in question. An example are stock prices that feed an option pricing formula.

For instance, assuming a log-normal asset distribution, the asset price S_T at a time horizon T can be expressed through

$$\mathbb{P}(S_T) = \frac{1}{S_0\sigma\sqrt{2\pi T}} e^{-\frac{(\ln S_T - \mu)^2}{2\sigma^2 T}} \tag{1}$$

with $\mu := \left(r - \frac{1}{2}\sigma^2\right) T + \ln S_0$, S_0 as the current price, r as the risk-free interest rate and σ as the asset volatility.

In the quantum setup, one generally represents a random variable X through a quantum state consisting of n qubits [80]. The first step usually consists of a truncation to $[X_{\text{lower}}, X_{\text{upper}}]$ and a discretisation of the distribution. For the latter, let $i = 2^{n-1} j_{n-1} + \cdots + 2j_1 + j_0$ with $j_k \in \{0, 1\}$ and $k = 0, \ldots, n-1$. The affine-

[11] With potentially n qubits ($|0\rangle_n$); to simplify the notation, a single qubit ($|0\rangle$) is used here.

linear transformation

$$\{0, \ldots, 2^n - 1\} \ni i \longrightarrow X_{\text{lower}} + \frac{X_{\text{upper}} - X_{\text{lower}}}{2^n - 1} i \in [X_{\text{lower}}, X_{\text{upper}}]$$

then fulfills the requirement. With $p_i \in [0, 1]$ as the corresponding probabilities, one can use a loading operator $\mathcal{P}_X$ to create a state

$$|0\rangle_n \xrightarrow{\mathcal{P}_X} |\psi\rangle_n = \sum_{i=0}^{2^n-1} \sqrt{p_i}|i\rangle_n, \tag{2}$$

with an n-qubit state $|i\rangle_n = |j_{n-1} \ldots j_0\rangle$. The d-dimensional extension of (2) reads

$$|\psi\rangle_n = \sum_{i_1,\ldots,i_d} \sqrt{p_{i_1,\ldots,i_d}}|i_1\rangle_{n_1} \otimes \cdots \otimes |i_d\rangle_{n_d}, \tag{3}$$

whereby $\otimes$ denotes the tensor product.

This setup allows the reflection of future scenarios of quantities such as asset prices, if required also jointly with other factors.

Stochastic processes. In order to express the evolution of asset prices, their returns or those of their valuation drivers through time, stochastic processes are a convenient way to formalise the dynamics. Notably, a stochastic process (S_t) implies a distribution of its modelled quantities at any future time point $0 < t \leq T$.

An important and flexible class of stochastic processes is the *Ornstein-Uhlenbeck (O-U)* type [85]. It has many applications in quantitative finance and can be used, for example, to model mean-reverting behaviour. A typical application is the modelling of interest rates, whose values tend to be 'constrained' to certain corridors through time. An O-U process is Gaussian and Markovian.[12] The general form reads $dx_t = \theta\,(\mu - x_t)\,dt + \sigma dW_t$. Assuming the Markov property, in a discrete setting, the increments of (S_t) between t to $t + 1$ can be expressed by a distribution that depends only on S_t.

In the quantum setup, in analogy to (3), one can prepare $|\psi\rangle_n$ as

$$|\underbrace{i_1}_{\Delta t}\underbrace{i_2}_{2\Delta_t}\ldots\underbrace{i_N}_{T}\rangle,$$

to reflect the prices or returns over time, based on N discrete time steps of length $\Delta t = T/N$. An extension to reflect the joint evolution of multiple stochastic processes follows the same building blocks.

[12] Generally, stochastic processes with stationary and independent increments have the Markov property.

3.5 Distributional properties: mean and quantiles

As explained, for example, in [91], one can define a function $f : \{0, \ldots, 2^n - 1\} \to [0, 1]$ and with this an operator

$$F : |i\rangle_n|0\rangle \to |i\rangle_n \left(\sqrt{1 - f(i)}|0\rangle + \sqrt{f(i)}|1\rangle\right), \quad i \in \{0, \ldots, 2^n - 1\}$$

that acts on an ancilla qubit. After applying F to $|\psi\rangle_n|0\rangle$ as per (2), one can use the resulting expression,

$$\sum_{i=0}^{2^n-1} \sqrt{1 - f(i)}\sqrt{p_i}|i\rangle_n|0\rangle + \sum_{i=0}^{2^n-1} \sqrt{f(i)}\sqrt{p_i}|i\rangle_n|1\rangle,$$

to approximate various quantities of the distribution. QAE can be applied to obtain the probability of measuring $|1\rangle$ in the last qubit, equal to $\sum_{i=0}^{2^n-1} p_i f(i)$. This expression is also that of the expectation of a (discrete) random variable X, i.e. $\mathbb{E}[f(X)]$.

Estimation of the mean. With

$$f(i) := i/(2^n - 1),$$

one can estimate $\mathbb{E}[\frac{X}{2^n-1}]$ and from that $\mathbb{E}[X]$.

Estimation of distribution quantiles. Quantile estimation is integral to risk measurement. Given the stochastic nature of the evolution of value drivers for financial instruments, their future values are expressed through distributions. By zooming into a particular quantile, one can summarise the risk in a single figure. For a random variable X and a given probability level $\alpha \in [0, 1]$, the quantile measure is defined as $Q_\alpha(X) = \inf\{x \in \mathbb{R} : \mathbb{P}[X \leq x] \geq \alpha\}$. In a quantum setup, let

$$f_I(i) = \begin{cases} 1, & \text{if } i \leq I \\ 0, & \text{otherwise} \end{cases}$$

so that applying the corresponding F_I to $|\psi\rangle_n|0\rangle$ yields a probability of measuring $|1\rangle$ in the last qubit of $\sum_{i=0}^{I} p_i = \mathbb{P}[X \leq I]$. With an interval search over I one can find the smallest I_α that fulfils $\mathbb{P}[X \leq I_\alpha] \geq \alpha$ and that is hence an estimator for $Q_\alpha(X)$.

3.6 Motivation for application in risk measurement

Many risk calculations are based on Monte Carlo simulation in order to obtain a distribution of future 'states of the world', from which measures can be derived—and on which usually risk controls such as limits will be imposed.

Basic Monte Carlo approaches can be improved in various ways to enhance convergence beyond $\mathcal{O}(M^{-1/2})$. A typical example is the use of quasi-Monte Carlo simulation that relies on deterministic, low-discrepancy sequences of random draws.

However, these are difficult to work with in higher dimensions that are required in financial risk calculations with a large numbers of risk factors. Not only is a better convergence rate not guaranteed, but results can even be subject to biases.

A quantum implementation for risk measurement purposes aims at exploiting the quadratic speed-up of Monte Carlo simulation offered through the use of QAE, as described in Sect. 3.3, i.e. a convergence rate $\mathcal{O}(M^{-1})$. Established techniques to reduce further the variance of estimators can be applied if considered necessary and suitable for high-dimensionality problems.

3.7 Quantum-specific challenges

Commercially relevant risk calculations present a particular challenge for quantum computing, especially for the following reasons:

- *Number of risk factors and their dependence structure*. The currently available quantum hardware exhibits only a limited number of qubits, typically not more than around 100 [34]. This can be a severe limiting factor for large-scale practical applications. Financial risk calculations often require a significant number of risk factors (see Sect. 3.1), modelled as correlated random variables. It is in part an algorithmic problem to construct these factors as quantum states in order to minimise the use of qubits as well as gates. Even the accurate reflection of distributions as per (2) and (3) already demands at least five qubits for each dimension (equivalent to a support with $2^5 = 32$ discrete points). Operations for risk factor convolutions, the application of payoff profiles, the determination of distributional properties and QAE itself require a non-negligible number of additional qubits.
  While it is possible to 're-use' qubits in a quantum circuit—essentially by 'undoing' previous operations, exploiting the unitary nature of all quantum gates—this process is not only adding complexity but also subjects the circuit to noise (see below). Furthermore, while the number of available qubits in hardware grows, not all designs allow all operations across all qubits. An example is a 'swap' of any two qubits, which might be limited to certain parts of the system. Quantum simulators can be used to develop and test algorithms, but the size of the problems cannot exceed a certain size, given that the calculations would tend to be prohibitively expensive in terms of computation time on classical machines.
  As a consequence, at least until quantum hardware has reached a much more mature state, 'hybrid' approaches have been suggested: only time-critical and suitable operations—ideally in a highly QPU-optimised format—are passed from the CPU to the QPU and back afterwards, also bearing in mind noise implications [24].

- *'Noise'*. Randomness arising from experiments like Monte Carlo simulation can be addressed by established statistical means such as confidence intervals as a function of the number of runs. An added layer of complexity in a quantum system stems from the fact that operations like gates are subject to 'quantum noise'. This encompasses any undesired sources that change the quantum system and can originate, for example, from hardware imperfections or thermal noise. Any practical quantum algorithm therefore needs to take the impact of noise on the

estimation results into account (see also Sect. 6) and, ideally, mitigate them to the largest extent possible [35, 48].

4 Application to market risk

4.1 General setup

Market risk relates to adverse changes in the PV of financial instruments. The projection horizon is usually short, between one and ten business days. It is quantified by determining the future distribution of changes to a portfolio PV. An extreme outcome of this P&L distribution is the VaR, which serves as a standard market risk measure.[13] It quantifies the extent of possible financial losses over a specific time horizon and for a given level of confidence. For a random variable ΔP representing the P&L and a probability $\alpha \in [0, 1]$, the measure is defined as

$$\mathrm{VaR}_{1-\alpha}(\Delta P) = -\inf\{p \in \mathbb{R} : \mathbb{P}[\Delta P \leq p] \geq \alpha\}.$$

Typical values in practice are $\alpha = 0.01$ and $\alpha = 0.05$.[14]

For the actual risk calculation, it is common to model the joint distribution of the *returns*, i.e. moves, of all relevant risk factors (e.g. stock prices and interest rates for various maturities). In conjunction with a repricing of the instruments in a portfolio, this allows to determine the P&L distribution and, from that, the VaR. One distinguishes three key methods for its calculation:

A. *Historical VaR.* Risk factors are modelled by directly applying historically observed returns to current market values. The dependence structure between risk factors is implicitly determined by their past co-movements.
B. *Parametric VaR.* Risk factors are modelled using probability distributions calibrated to historical returns; simplifying assumptions such as a multivariate normal distribution are made in order to obtain closed-form expressions for the VaR.
C. *Monte Carlo VaR.* Risk factors are again reflected by probability distributions calibrated to historically observed returns. The joint distributions are often fitted to historical data more closely than in the case of a Parametric VaR and, as such, usually do not allow for closed-form expressions.

The Monte Carlo VaR lends itself well to applying the QAE algorithm (as per Sect. 3.3). Practical use cases typically rely on a good representation of the *marginal* risk factor returns for which a normal distribution is an over-simplification.

Figure 1 summarises the components of a Monte Carlo-based market risk calculation and how classical and quantum computer implementations differ. In Sect. 4.2, the risk factor modelling and its calibration are described. Besides the move from normal to non-normal marginal factor distributions, the way to 'decouple' those from the dependence structure with the help of copulas is introduced. Starting from the

[13] See [49] as a standard reference on market risk.

[14] An extension to Expected Shortfall (ES) [59] is straightforward. As a 'tail average', it is defined as $\mathrm{ES}_{1-\alpha}(\Delta P) = \frac{1}{\alpha}\int_0^\alpha \mathrm{VaR}_{1-\gamma}(\Delta P)d\gamma$.

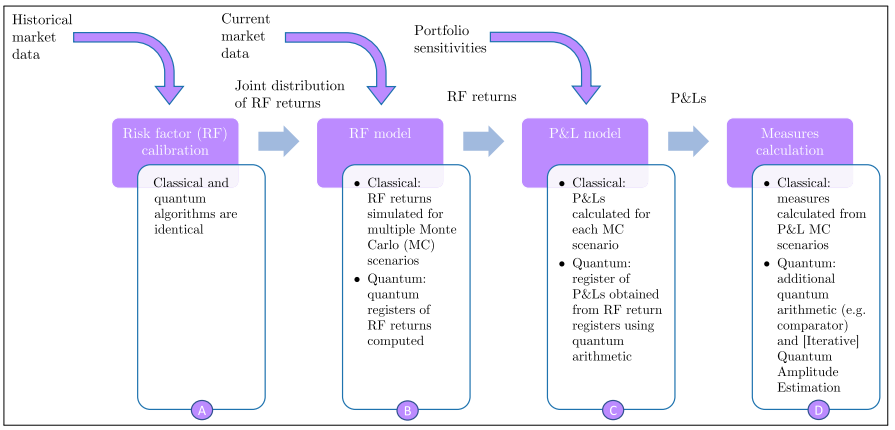


Fig. 1 Components of a market risk calculation—comparing classical and quantum implementations. In Step A, risk factor model parameters are inferred from historical market data. With the exception of more complex model-fitting routines, there is no specific use case for a quantum algorithm, and the approach is identical to the classical one. Based on current market data, future scenarios of the world are then simulated in Step B. Combining those with risk factor sensitivities leads to the corresponding P&Ls (Step C). These two steps are formulated differently on a quantum compared to a classical computer: there are no Monte Carlo scenarios, with each being a realisation of a random variable. Instead, quantum registers represent the random variables for both the risk factor returns and the P&Ls. In order to obtain the output in the form of a risk measure (Step D), the classical approach makes use of the simulated P&L distribution and its statistics, such as an extreme quantile. A quantum algorithm will encode this last step (see Sect. 3.5) and ultimately rely on [Iterative] Quantum Amplitude Estimation to obtain a measurement

derivation of the underlying P&L distribution, the VaR calculation itself is covered in Sect. 4.3. In Sect. 4.4, following the approach in [91], the model implementation on a quantum computer is presented; a special focus hereby lies on the distribution loading. Sect. 4.5 analyses the estimation results.

4.2 Risk factor model

4.2.1 Return definitions

Let the set of d relevant risk factors for a portfolio, for example, stocks or commodity prices, at a time point t be denoted by

$$\boldsymbol{S_t} := (S_{1,t}, S_{2,t}, \ldots, S_{d,t}).$$

Between any two points t and $t + \Delta t$, the vectors of absolute, logarithmic and relative risk factor returns are given by

$$\begin{aligned}
\boldsymbol{\Delta S_t^{\text{abs}}} &= \boldsymbol{S_{t+\Delta t}} - \boldsymbol{S_t}, \\
\boldsymbol{\Delta S_t^{\log}} &= \ln(\boldsymbol{S_{t+\Delta t}}) - \ln(\boldsymbol{S_t}), \\
\boldsymbol{\Delta S_t^{\text{rel}}} &= \boldsymbol{\Delta S_t^{\text{abs}}} / \boldsymbol{S_t} \approx \boldsymbol{\Delta S_t^{\log}}.
\end{aligned}$$

The approximation for the relative returns follows from series expansion and tends to hold in the case of small returns.

4.2.2 Accommodating empirical risk factor distributions

A common approximation used particularly in Parametric VaR is that risk factor returns over a period Δt follow a multivariate normal distribution:

$$\mathbf{\Delta} S_t^{\log} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma}), \tag{4}$$

with mean vector $\boldsymbol{\mu}$ and covariance matrix $\mathbf{\Sigma}$, both assumed to be constant over time. The associated risk factor distribution is then given by a multivariate extension of (1). An advantage of this normal model is that efficient implementations exist for the distribution loading, i.e. the preparation of quantum registers holding correlated normal random variables (see Sect. 4.4.1).

In many empirical cases, however, marginal probability density functions (PDFs) of risk factor returns are leptokurtic, and the simple normal model is not sufficiently accurate. A more appropriate modelling of actual marginal distributions is one of the main reasons for choosing a Monte Carlo over a Parametric VaR.

Copulas [60, 67] provide a convenient way of separating the marginal risk factor PDFs from the underlying dependence structure. This facilitates both calibration and efficient implementation of the Monte Carlo VaR, even for a large number of risk factors. The Sklar theorem of copula theory states that for real-valued random variables $(X_1, \ldots, X_d)$ with joint cumulative density function (CDF) $F^X(x_1, \ldots, x_d) = \mathbb{P}[X_1 \leq x_1, \ldots, X_d \leq x_d]$, there exists a copula $C : [0, 1]^d \to [0, 1]$ such that

$$F^X(x_1, \ldots, x_d) = C\left(F_1(x_1), \ldots, F_d(x_d)\right).$$

Here $F_k(x_k) = \mathbb{P}[X_k \leq x_k]$ is the marginal distribution of random variable X_k, $k = 1, \ldots, d$.

For extreme market events, reflecting strong 'tail dependence' between the marginal risk factor distributions can be required. Here, as a simplifying assumption and in line with industry practice, a Gaussian copula is chosen:

$$C_K^{\text{Gaussian}}(\mathbf{u}) = \Phi_d(\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_d); \mathbf{K}), \tag{5}$$

where $\Phi(z)$ is the CDF of a standard normal variable, and $\Phi_d(\mathbf{z}; \mathbf{K})$ denotes a joint standard normal multivariate CDF with mean zero and correlation matrix $\mathbf{K}$.[15] For classical and quantum computers, variates reflecting empirical distributions can then be generated efficiently by applying functions G_k such that $X_k = G_k(Z_k)$. Hereby, Z_k are random variables governed by $\boldsymbol{Z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{K})$. This is apparent when first considering the CDF of $\boldsymbol{Z}$ expressed in terms of the Gaussian copula, i.e.

$$F^Z(z_1, \ldots, z_d) = C_K^{\text{Gaussian}}\left(\Phi(z_1), \ldots, \Phi(z_d)\right).$$

[15] The relationship between correlation $\mathbf{K}$ and covariance $\mathbf{\Sigma}$ is given by $\mathbf{K} = \mathbf{D^{-1}\Sigma D^{-1}}$, with $\mathbf{D} = \sqrt{\text{diag}(\mathbf{\Sigma})}$.

With $x_k = G_k(z_k) \Leftrightarrow z_k = G_k^{-1}(x_k)$, the CDF of the targeted random variables can then be written as

$$F^X(x_1, \ldots, x_d) = C_K^{\text{Gaussian}}\left(\Phi(G_1^{-1}(x_1)), \ldots, \Phi(G_d^{-1}(x_d))\right).$$

In classical Monte Carlo simulation, samples $\mathbf{z} = (z_1, \ldots, z_d)$ are drawn from the multivariate normal distribution Φ_d. From this, non-normal variates are obtained through

$$x_k = G_k(z_k). \tag{6}$$

On a quantum computer, a register represents a random variable, and the transformation G_k is applied to each of them. Given the limited number of available qubits, it is convenient to define and calibrate the quantities of interest in the following standardised way, to ensure numerical values of X_k of similar magnitude:

$$X_k := \frac{\Delta S_k^{\text{rel}}}{\sigma_k} = G_k(Z_k), \tag{7}$$

with σ_k obtained from $\mathbf{\Sigma}$ in (4). The aim of the model calibration is to determine the G_k for all k.

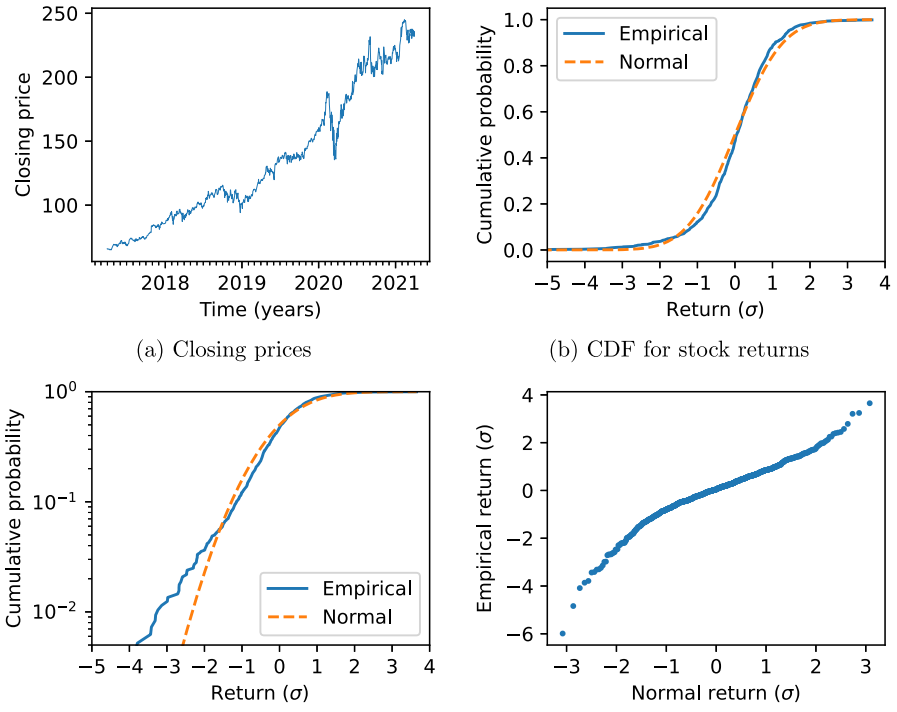
4.2.3 Calibration

Market risk models project the P&L over a short time horizon of length η (years). A typical value found in practice amounts to ten business days. An assumption of 252 business days per year translates into $\eta = 10/252$. The model calibration uses at least one year of historical data prior to the risking date. In the following, four years are chosen. This is helpful in obtaining sufficiently large samples in the tails of the distributions and thereby allowing a good demonstration of the approach.

For illustration purposes, a portfolio driven by two risk factors—the stock prices of Microsoft Corporation (MSFT) and Apple Inc. (AAPL)—is created. S_1 relates to MSFT and S_2 to AAPL. Such a portfolio can consist of various financial instruments. If these were to be solely stocks, the sensitivity to price changes would be constant and equal to one (or minus one for stocks that are sold short). In the more general case, the portfolio with a current value of P will have sensitivities $\delta_k = \frac{\partial P}{\partial S_k}$ to each risk factor and potentially higher-order ones as well.

The reference date for the calibration is selected as 1st April 2021, and the calibration data comprises the period 31st March 2017 through 31st March 2021.[16] The history of the prices for MSFT is plotted in Fig. 2a. By choosing ten-day returns for the calibration, the resulting distribution reflects the targeted risk projection horizon. As an initial step, the individual series of (overlapping) log-returns for the stock prices

[16] Daily closing prices in USD. Data source: *Yahoo! Finance*, http://finance.yahoo.com.



(a) Closing prices

(b) CDF for stock returns

(c) CDF for stock returns [log-scale]

(d) Normal vs. empirical returns

Fig. 2 Marginal CDF. The figures are derived from MSFT stock price data (in USD), as shown in **a**. Ten-day overlapping log-returns are calculated and expressed in units of standard deviation for **b** through **d**. The distribution is leptokurtic; for example, a cumulative probability of 1% corresponds to a normal return of −2.33 standard deviations and an empirical return of −3.26 standard deviations. In **d**, for the cumulative probability associated with each empirical return on the y-axis, the corresponding normal return is plotted on the x-axis (a form of Q–Q plot); a function G as per (7) can be fitted to this data

are calculated (omitting the risk factor index k in the following for clarity):[17]

$$\Delta s_{t_i}^{\log} = \ln(s_{t_{i+10}}) - \ln(s_{t_i}),$$

where s_{t_i} is the historical closing price at time t_i, with $i = 1, 2, \ldots, n$ indexing the business days.

For the actual marginal risk factor calibrations, it is useful to postulate that the mean ('drift') for each stock is equal to zero. This assumption is justifiable given that the estimation of the drifts from historical data points is not only prone to large estimation uncertainty but also not necessarily predictive of the future stock price trends.

[17] The choice of the length of the return window and whether to use overlapping or decimated returns does not impact the facility of the quantum circuit implementation.

In a multivariate normal model as per (4), the resulting parameters read $\boldsymbol{\mu} := (\mathbf{0})$ and

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1.854 & 1.698 \\ 1.698 & 3.446 \end{pmatrix} \cdot 10^{-3}.$$

This covariance matrix equates to standard deviations of the MSFT and AAPL stocks of $\sigma_1 = 0.0431$ and $\sigma_2 = 0.0587$, respectively.[18] The pairwise linear correlation amounts to about +67%.

In line with the standardised form of (7), let

$$x_i = \frac{\Delta s_{t_i}^{\log} - \mu}{\sigma}$$

denote the standardised risk factor series, with mean μ and standard deviation σ as their respective moments. From this, the empirical cumulative probability of each value x_i, f_i, is calculated as[19]

$$f_i = \frac{1}{n} \sum_{j=1}^{n} \mathbb{1}_{x_j \leq x_i}.$$

The result for MSFT is plotted in Fig. 2b (linear scale) and 2c (log-scale).

As outlined in Sect. 4.2.2, the aim is to obtain functions G_k for each return series k as per (7). To this end, Fig. 2d is a scatter plot for MSFT with x_i on the y-axis ('empirical returns') and $\Phi^{-1}(f_i)$ on the x-axis ('normal returns'). G_k is then fitted to this data.[20] Piecewise polynomials of low order will often be sufficient for this step, which is helpful for the implementation on a quantum computer.

For the marginal MSFT returns, a piecewise cubic fit using seven knot points is shown in Fig. 3a. The corresponding density is plotted in Fig. 3b, over a histogram of the empirical returns. Such an approach can capture the specificities of the marginal distribution. Even a simple piecewise linear fit using only two knot points (Fig. 3c) already provides a significant improvement over a normal distribution. In particular, as shown in Fig. 3d, a much-improved fit to the 1% cumulative probability is obtained.

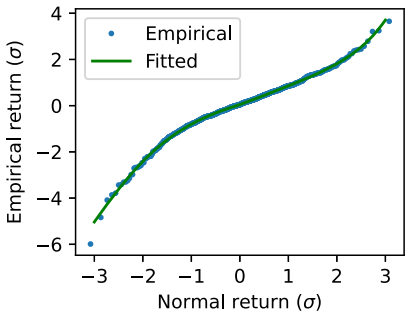
4.3 VaR calculation

The portfolio P&L—between t and $t + \Delta t$—is a random variable, ΔP, and expressed as a function of the changes in the risk factors, here ΔS_1^{abs} and ΔS_2^{abs}. With δ as

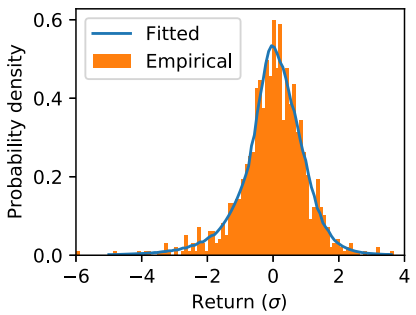
[18] Under the assumption of normality of returns, the annualised and easier-to-interpret values read 22% and 29%.

[19] Even when assuming perfect stationarity, the estimation from limited datasets results in uncertainty in the CDFs. This is not taken into account here but might deserve closer attention when judging quantum estimation errors and quantum noise with respect to their practical implications.

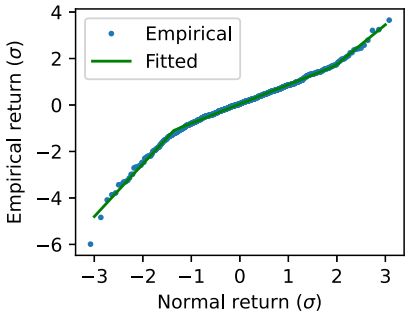
[20] For simplicity, $\boldsymbol{\Delta S_t^{\text{rel}}} \approx \boldsymbol{\Delta S_t^{\log}}$ is assumed here; by applying $x_i \to \frac{\exp(\sigma x_i) - 1}{\sigma}$ before the fit, this approximation can be removed.



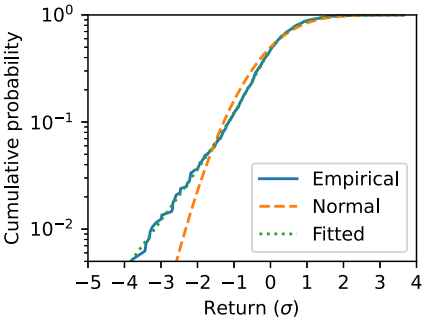
(a) Normal vs. empirical returns: fitted piecewise *cubic* function



(b) Empirical and fitted PDF – using a piecewise *cubic* function



(c) Normal vs. empirical returns: fitted piecewise *linear* function



(d) CDF [log-scale] – empirical, normal and piecewise *linear* fit

Fig. 3 Fitting of marginal CDF. The functions G_k as per (7) are obtained by fitting empirical to normal returns. A piecewise cubic fit is used in **a** and **b**, with the latter demonstrating a 'smoothing effect' on the PDF. **c** and **d** are based on a piecewise linear fit, which already provides a reasonable approximation of the tails of the empirical distribution

the stock-specific portfolio sensitivity to price changes and only retaining first-order terms:[21]

$$\begin{aligned}\Delta P &= \delta_1 \Delta S_1^{\text{abs}} + \delta_2 \Delta S_2^{\text{abs}} \\ &= \delta_1 S_{1,t} \sigma_1 \frac{\Delta S_1^{\text{rel}}}{\sigma_1} + \delta_2 S_{2,t} \sigma_2 \frac{\Delta S_2^{\text{rel}}}{\sigma_2}.\end{aligned}$$

To facilitate the implementation on a quantum computer, the P&L equation is expressed using two constants a and b.[22] Furthermore, (7) is substituted:

[21] The full 'repricing' of the portfolio would be an alternative, but given the computational demand, a sensitivity-based approach is often preferred when using a Monte Carlo-based VaR. Notably, higher-order sensitivities and various other improvements can be added in practice if deemed necessary.

[22] For simplicity, the multiplication by b is performed as part of the distribution loading.

$$\begin{aligned}\Delta P &= \delta_1 S_{1,t}\sigma_1 \left(\frac{\Delta S_1^{\text{rel}}}{\sigma_1} + b\frac{\Delta S_2^{\text{rel}}}{\sigma_2} \right) \\ &= a\left(G_1(Z_1) + bG_2(Z_2)\right)\end{aligned}$$

where

$$\begin{aligned}a &= \delta_1 S_{1,t}\sigma_1 \\ b &= \frac{\delta_2 S_{2,t}\sigma_2}{\delta_1 S_{1,t}\sigma_1}.\end{aligned}$$

The normalised P&L is defined by omitting the constant a:

$$\begin{aligned}\Delta P &= a\Delta P_{\text{norm}} \\ \Delta P_{\text{norm}} &= G_1(Z_1) + bG_2(Z_2).\end{aligned} \tag{8}$$

The VaR is calculated from the quantiles of ΔP_{norm}, multiplied by a in order to obtain the final result.

4.4 Setup for a quantum computer

4.4.1 Distribution loading for probability density functions

A range of algorithms has been proposed to create a joint PDF in a quantum state. The authors of [79] describe how an arbitrary quantum state can be initialised using $2^{q-2} - 2q$ CNOT gates, where q is the total number of qubits used in all quantum registers representing the random variables. [42] describes an algorithm that can be used for 'efficiently integrable' PDFs. While a normal distribution can be handled, there is a restriction on the distributions that can be generated. More recent work attempts to find algorithms that can load arbitrary distributions with fewer gates [25, 96].

The functions G_k as defined in (7) can be applied to reflect the empirical marginal risk factor distributions needed for the VaR calculation. This provides an alternative to the 'direct' approach of [79]. The 'piecewise transformation' technique avoids the problem with the direct method that the number of required gates becomes prohibitive when generating large sets of risk factors as in typical market risk applications. Each approach has its advantages, however, and is presented in turn.

'Direct' loading. The approach of [79] is used to initialise two quantum registers representing the risk factor returns to a vector of amplitudes. This is the implementation of the loading operator $\mathcal{P}_X$ from Sect. 3.4.[23] The amplitudes are calculated using the joint PDF. This can be achieved by expressing the PDF with the help of the copula C:

[23] The 'StatePreparation' class in *Qiskit* is used for this step.

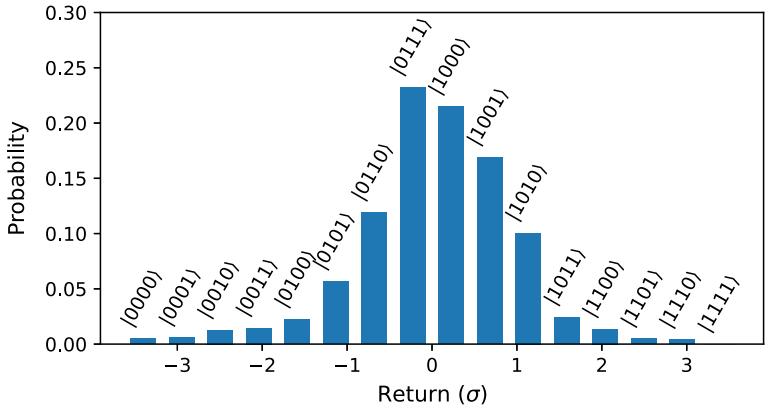


Fig. 4 Marginal PDF of a calibrated quantum state. Using $q = 4$ qubits, the distribution—with $2^4 = 16$ discrete states—is generated by (noise-free) simulation. The underlying CDF is the piecewise cubic fit of MSFT returns as per Sect. 4.2.3

$$f(x_1, \dots x_d) = \frac{\partial^d C}{\partial x_1 \dots \partial x_d} = c\left(F_1(x_1), \dots, F_d(x_d)\right) f_1(x_1) \dots f_d(x_d), \tag{9}$$

whereby c denotes the copula density and $f_k(x_k) = \frac{dF_k}{dx_k}$. For the case of the Gaussian copula as in (5), the density is given by

$$c_K^{\text{Gaussian}}(\boldsymbol{u}) = |\boldsymbol{K}|^{-1/2} \exp(-\boldsymbol{\zeta}^\top (\boldsymbol{K}^{-1} - \boldsymbol{I_d})\boldsymbol{\zeta}/2),$$

with $\boldsymbol{\zeta} := (\Phi^{-1}(u_1), \dots, \Phi^{-1}(u_d))^\top$ and $\boldsymbol{I_d}$ as the identity matrix [67].

In the d-dimensional case, the probabilities, $p_{i_1,\dots,i_d}$, are computed using the joint PDF from (9). The discrete values of x_k are given by $x_k^{i_k}$ with $i_k \in \{0, \dots, n-1\}$ and n indicating the number of possible values of x_k:

$$p_{i_1,\dots,i_d} = f(x^{i_1}, \dots, x^{i_d}) / \sum_{i_1,\dots,i_d} f(x^{i_1}, \dots, x^{i_d}).$$

A 'loaded state' for the MSFT returns based on $q = 4$ qubits is shown in Fig. 4. For this calculation, piecewise cubic fits (as per Fig. 3a) are used to generate the CDF.[24]

Piecewise transformation. An alternative way to load the distribution is to apply transformations—the functions G_k as in (7)—to quantum registers that have been loaded in a multivariate normal state (for which efficient algorithms exist). These transformations are applied to each register in isolation. The number of required gates therefore scales linearly with the number of risk factors, rendering the algorithm

[24] The piecewise linear fit is also a good approximation. Using the piecewise cubic fit illustrates that, for the case of direct loading, the functional form is not a restriction.

suitable for large-scale calculations. The functions G_k themselves are constructed as piecewise transformations.

For the distribution loading, a three-section piecewise linear transformation is applied to a quantum register $|i\rangle_n$, a discretised random variable defined by (2) and based on n qubits. The function is shown in Fig. 3c on the example of MSFT. The transformation can be written as

$$\mathcal{F}(i) := a_p \cdot i + b_p, \quad c_{p-1} \leq i < c_p, \tag{10}$$

with $p = 0, 1, 2$ and $c_{-1} := -\infty$. Integer arithmetic is used: a_p, b_p and c_p are all integers and, as above, $i \in \{0, \ldots, 2^n - 1\}$. The register holding $\mathcal{F}(i)$ requires $n + m$ qubits; the additional m qubits are needed as a result of the multiplication and addition/subtraction operations, i.e. m depends on a_p and b_p. This setup allows a non-integer transformation of a normal return (in units of the standard deviation), $z(i)$, to an empirical return (in units of the standard deviation), $x(i)$. Explicitly, normal returns are discretised by[25]

$$z(i) = Z_{\text{lower}} + \frac{Z_{\text{upper}} - Z_{\text{lower}}}{2^n - 1} i$$

and empirical returns by

$$x(j) = X_{\text{lower}} + \frac{X_{\text{upper}} - X_{\text{lower}}}{2^{n+m} - 1} j.$$

The transformation is given by

$$x(i) = G(z(i)) = X_{\text{lower}} + \frac{X_{\text{upper}} - X_{\text{lower}}}{2^{n+m} - 1} \mathcal{F}(i).$$

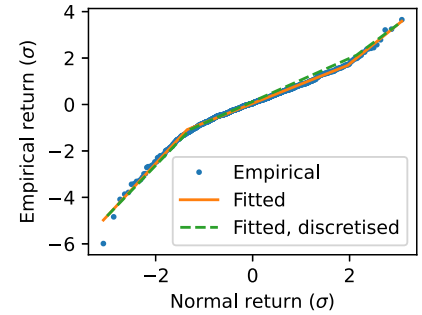
As an example, a three-qubit normal distribution is converted into a five-qubit empirical distribution ($n = 3, m = 2$). Figure 5a shows the transformation of the actual values, and Fig. 5b plots the integer transformation as per (10). For comparison, the piecewise transformation of the real quantities without discretisation is displayed. Five qubits are chosen to represent $\mathcal{F}(i)$, a compromise that provides a sufficiently good fit whilst keeping the number of qubits low.

This form of distribution loading is not identical to the 'direct' approach but instead represents a redistribution of the probability density. The normal distribution in Fig. 5c is transformed into the empirical distribution as per Fig. 5d.

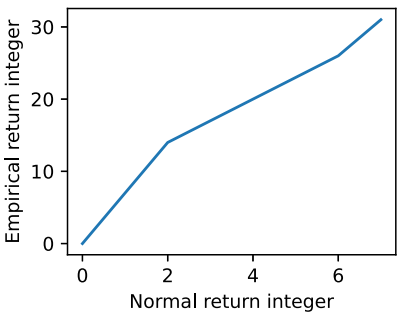
4.4.2 Quantum circuit

Following the approach of [91], QAE (see Sect. 3.3) is used to compute the VaR. Figure 6a shows the layout of the circuit. It calculates cumulative probabilities of the

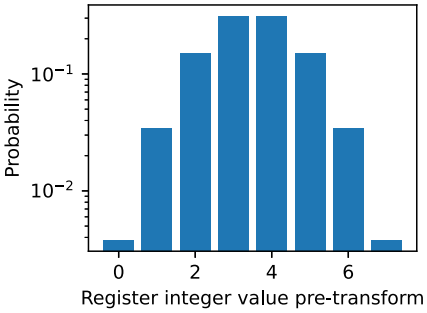
[25] The upper and lower bounds are selected as three standard deviations below and above the mean.



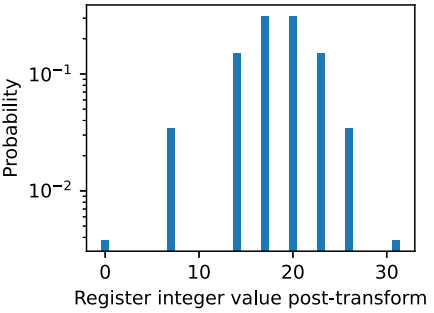
(a) Normal vs. empirical returns: piecewise linear fit and discretised fit



(b) Discretised fit as a transformation of an integer register



(c) Integer register prepared with a normal PDF, pre-transformation [probabilities on log-scale]



(d) Integer register post-transformation [probabilities on log-scale]

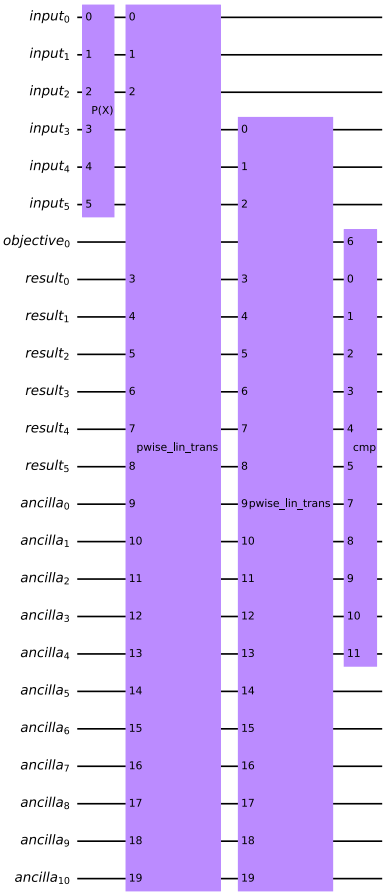
Fig. 5 Piecewise transformation applied to a three-qubit register. **a** and **b** show a piecewise linear fit of the transformation from normal to empirical returns and a discretised form of this transformation that can be applied to the (integer) quantum registers. **c** and **d** provide the PDFs of the quantum registers pre- and post-transformation, respectively, illustrating the redistribution of probabilities. Note that an extra two qubits are added for **d**

normalised P&L ΔP_{norm} as per (8); all numerical values derived and shown subsequently are for the case $a := 1$, which is just a multiplicative constant. Furthermore, an example value of $b := 1$ is chosen, although the circuit can accommodate an arbitrary value by means of modifying the piecewise transformation coefficients.[26]

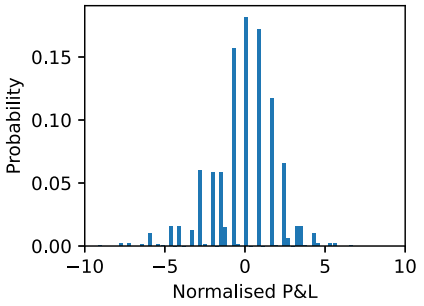
A correlated normal distribution is first loaded into two three-qubit registers by circuit $P(x)$ of Fig. 6a.[27] Each three-qubit random variable is then transformed into an empirical random variable using a piecewise linear transformation circuit and added to the result register; three extra qubits are needed to store the result. Finally, an additional integer comparator—as described in [91]—is used to generate the single-qubit objective, the amplitude of which is to be estimated. A bisection search applied to the CDF is then employed to calculate the VaR (see as well [91] for more details).

[26] For $b = 1$, it is ensured that the P&L contributions from both risk factors are relevant for the portfolio.

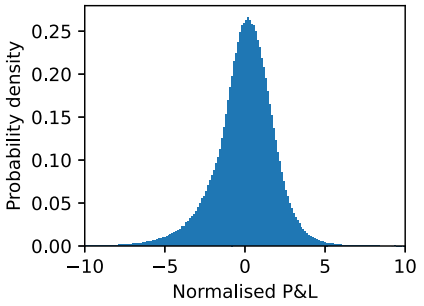
[27] The 'NormalDistribution' class in *Qiskit* is used. The implementation is based on [79], although—importantly—alternatives with a better scaling to a large number of risk factors exist [42].



(a) Quantum circuit used to create a state representing profit and losses



(b) P&L distribution obtained from simulation of the quantum circuit



(c) P&L distribution resulting from classical Monte Carlo simulation

Fig. 6 Quantum circuit for market risk. **a** shows the circuit that obtains a quantum register of P&Ls. As a final step, an integer comparator ('cmp') is applied, which flips the objective qubit if the register value is less than a specified integer; the amplitude of this objective is then estimated using QAE. In order to test the *Qiskit* implementation, one million measurements of the result register are used to construct the probability distribution of the P&L, as shown in **b**. For comparison, the equivalent distribution is obtained by classical Monte Carlo simulation through one million samples drawn from a continuous multivariate normal distribution and transformed using (6); the result is displayed in **c**

The details of the piecewise transformation circuits are shown in Fig. 7. Integer comparators compare the value in the n-qubit register to c_0 and c_1 of (10). The resulting qubits are used to control three in-place multiply-add circuits that apply the three linear transformations of (10) for $p \in \{0, 1, 2\}$. Each of the three performs an in-place multiplication by a classical integer and in-place addition with the result, followed by an addition with a classical integer. This arithmetic makes use of circuits adapted

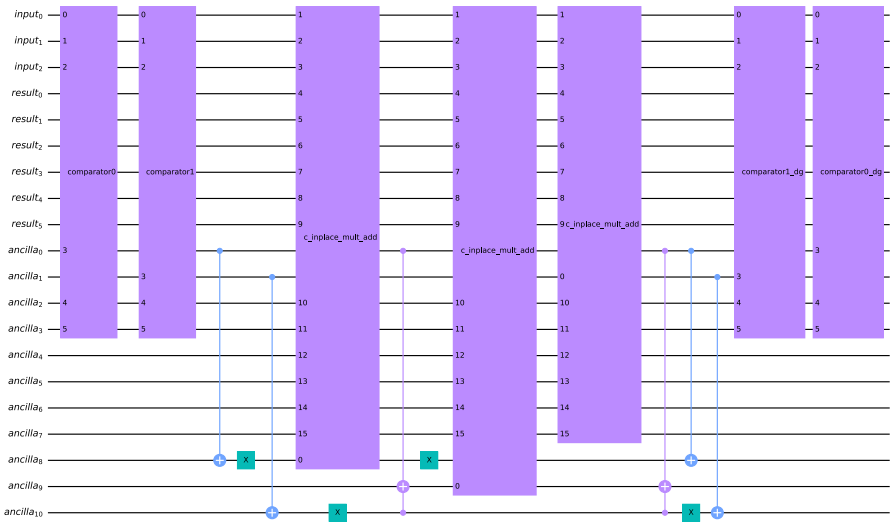


Fig. 7 Quantum circuit for piecewise transformation applied to a three-qubit register. Three conditional in-place multiply-add circuits are used to apply (10) for $p \in \{0, 1, 2\}$. The coefficients a_p and b_p used in the multiply-adds are classical. Two integer comparators store the result of $i \geq c_0$ and $i \geq c_1$ into ancilla qubits *ancilla_0* and *ancilla_1*, respectively. Two additional qubits, *ancilla_8* and *ancilla_9*, are used to calculate booleans $i < c_0$ and $c_0 \leq i < c_1$ from these qubits. *ancilla_8*, *ancilla_9* and *ancilla_1* are then utilised to condition the multiply-add circuits. The multiply-add circuits include an 'uncompute' step, and additional gates (shown) are added to uncompute *ancilla_8*, *ancilla_9* and *ancilla_10*. Finally, the comparator ancillas *ancilla_0* and *ancilla_1* are uncomputed. These steps allow the ancilla qubits to be re-used

from [86].[28] A number of ancilla qubits are required to perform the piecewise transformation. These are 'uncomputed' following each transformation and then re-used for each marginal distribution.

The circuit 'depth', as a measure of size and representing the smallest number of steps required to execute an experiment, depends on the circuit transpiler and the quantum hardware to run on. While gates without common qubits can be processed in the same step, every gate that has at least one qubit in common needs to act on a different step. Using only elementary rotation and conditional NOT gates, the depth amounts to 9,387 ($q = 2$ per risk factor) and 22,589 ($q = 3$), respectively.[29]

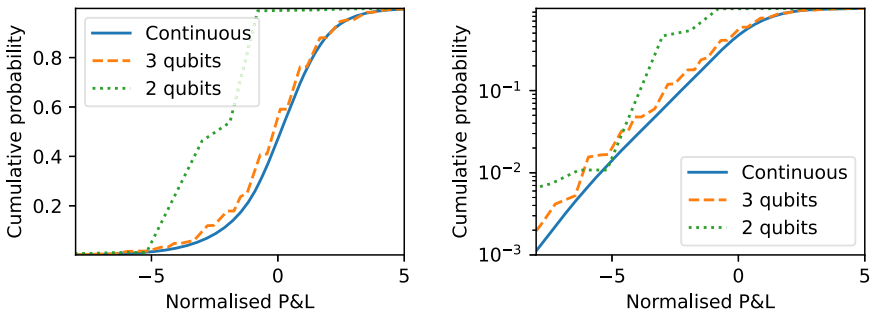
4.5 Estimation results

All computations are carried out on a quantum simulator.[30] The implementation of the quantum circuit is tested by executing the quantum circuit and the final comparator

[28] The ripple-adder of [86], as opposed to an approach based on the Quantum Fourier Transformation, is selected, because it is expected to be more noise-tolerant for the next generation of quantum computers.

[29] The *Qiskit* implementation allows the transpilation of the circuits and provides a 'depth' method to obtain the gate count.

[30] As of November 2022, IBM provides (paid-for) access to systems with 27 qubits, extended to up to 127 qubits on experimental devices; see https://quantum-computing.ibm.com/services?services=systems.



(a) Cumulative normalised P&L probability distribution [linear scale].

(b) Cumulative normalised P&L probability distribution [log-scale].

Fig. 8 Cumulative normalised P&L probability distribution. The normalised P&L refers to the definition in (8). The values obtained using $q = 2$ and $q = 3$ qubits per risk factor are shown, together with the 'continuous' values—i.e. with no discretisation

of Fig. 6a to calculate the probability distribution of the normalised P&L. This is shown in Fig. 6b. The analogue can be achieved classically, by using the PDF of the discretised quantum register output from $P(x)$ of Fig. 6a and the same coefficients for the piecewise approximation as in the quantum case. In this way, it is verified that the same distribution is produced. The classical calculation without using the same discretisation as the quantum circuit—in fact, without discretisation at all—is shown in Fig. 6c.

With this validation complete, the (Iterative) QAE algorithm[31] is applied to the objective qubit of the circuit of Fig. 6a. This is used to obtain the CDF of the normalised P&L as per (8). The cumulative probabilities are shown in Fig. 8a, b which are identical apart from the scale. The results are plotted with three qubits for each of the two correlated normal registers (the case shown in Fig. 7), as well as the two-qubit case for comparison. The result obtained without discretisation error is also provided as a reference. For this comparison, continuous normal distributions are used in place of the discretised versions, in conjunction with a piecewise linear transformation free from discretisation error.[32] This can be thought of as being the result calculated with a large number of qubits. Results for three typical P&L quantiles are found in Table 1.[33] Using $q = 3$ qubits for each normal register provides a tolerable—though still rather large—level of discretisation error. The 99% VaR calculated using three qubits amounts to -6.22 and differs by around 14% from the correct value of -5.44. Practitioners would therefore likely require a larger number of qubits.[34]

[31] Parameters are set to $\alpha = 0.05$ and $\epsilon = 0.01$. Still, any accuracy sufficient to verify consistency with the results of Fig. 6b suffices.

[32] This entails relying on the 'fitted' as opposed to the 'fitted, discretised' function in Fig. 5a.

[33] The ES can be calculated using the same circuit from Fig. 6a See also [91] for more details on a suitable quantum implementation.

[34] For comparison: if one were to rely on a Parametric VaR and hence a multivariate normal distribution of the stock returns, the 99% VaR would read -4.30. The use of empirical marginal distributions is hence material in this example. It even outweighs the discretisation error in the quantum calculation.

Table 1 Estimation results for the market risk example

	VaR quantile		
	1%	2.5%	5%
A. Classical computation			
Continuous	−5.44	−4.21	−3.23
B. Quantum computation—simulator			
$q = 2 \cdot 2$	−6.56 (20%)	−5.13 (22%)	−5.01 (55%)
$q = 2 \cdot 3$	−6.22 (14%)	−4.88 (16%)	−3.77 (17%)

The figures compare the VaR at various quantiles between classical (Panel A) and quantum computations (Panel B). The latter result from applying IQAE, with $2^{10} = 1{,}024$ shots each, on a quantum simulator. The input registers use $q = 2$ and $q = 3$ qubits per risk factor, respectively. Confidence intervals are very tight and hence omitted in the table. Relative unsigned differences between the quantum-derived figures and the references values in Panel A are shown in parentheses

5 Application to counterparty credit risk

5.1 General setup

Counterparty credit risk (CCR) is concerned with cases where one party to a portfolio of instruments—usually bilaterally negotiated financial derivatives—defaults and is not able to honour its obligations.[35] Examples of such transactions are forwards, which allow to lock in an exchange rate such as USD/EUR in advance, and options, which entail the right to purchase or sell assets such as stocks or commodities at a pre-determined price at a later date. While market risk is centred around losses occurring from a decline in the value of instruments, CCR arises from portfolios that have a positive PV and which cannot be realised if the counterparty defaults. Since the surviving party, such as a bank, will typically have hedged the exposure with other—non-defaulted—counterparties, it is left with 'open' positions in this case. Replacing those in the market will in some circumstances only be possible at worse conditions than at origination. CCR is therefore the loss associated with these replacement costs.

Contrary to straightforward loans, the exposure over time versus a particular counterparty is not deterministic if the instruments are exposed to PV changes. This is nearly always the case with derivatives.[36] CCR modelling is about estimating the Potential Future Exposure (PFE).

If trade netting has been agreed between the parties, positive and negative PVs can be offset against each other in the case of a default. A notable mitigant of CCR exposure is the use of collateral, which can be posted between the counterparties or by one or both of them to an independent depositary. As an example, if party A buys an option from party B, B will post the PV of the option at inception as collateral ('variation margin'). This, in turn, renders the resulting exposure equal to zero since A could use

[35] See, for example, the introductory overview to CCR in [40].

[36] Note that CCR requires at least two parties to a trade. Buying a stock at an exchange and holding it therefore does not give rise to CCR. The risk in this case is purely associated with the inherent risk of the stock and its issuing firm.

the collateral to enter a new option trade in case B were to default. As the value of the option evolves over time, the collateral balance is either topped up or decreased—or in some cases, left unchanged.[37] For the purposes of PFE measurement, the collateral balance through time is added to the portfolio PVs to determine the overall counterparty exposure profile.[38] In spite of using variation margin, residual CCR can result from the legal and operational process associated with a default requiring several days or even weeks to complete. A re-hedging in the market by the surviving counterparty might hence only take place with a delay. During this 'slippage' period, the market value of the portfolio can evolve and lead to non-zero replacement costs even after taking any received collateral into account. As a solution, additional 'initial margin' can be posted by both parties, these days typically to a third-party custodian bank. In case of a default, the amount is released to cover any slippage-related costs.[39]

In order to obtain the PFE of a portfolio, one will focus on a projection horizon (T) that covers the period up to the longest maturity across all instruments. The time interval is divided into N, not necessarily equally spaced discrete points $t_i \in [0, T], i = 1, 2, \ldots N$ with $t_i < t_{i+1}$ and $t_0 = 0$. In the first process step, risk drivers of the instruments (e.g. stock prices) are diffused over time, expressed either by means of stochastic processes or through distributions.[40] In certain cases, the resulting risk factor distribution at each t_i is known in closed form; in the most general case, it can be approximated through Monte Carlo simulation. On the back of the joint risk factor distributions across time, the corresponding instrument PVs as well as the resulting portfolio PVs are determined as a second step.[41] In cases where collateral is exchanged between the parties, a third step is added, applying the projected collateral value across time and adding it to the PVs. Since CCR is only concerned with positive balances vis-a-vis a counterparty, the portfolio PVs including collateral are transformed into exposures by flooring them to zero. The result of these steps is a series of exposure distributions $E_{t_i}, i = 1, 2, \ldots N$.[42,43] Given the stochastic nature of the exposure, one will often work with quantile measures, $q_c(E_{t_i})$, to quantify the CCR. Hereby, $c \in \{0.90, 0.95, 0.99\}$ are typical quantiles chosen in practice.[44] One can

[37] If the instrument can have a positive or negative value to the parties, such as in the case of an interest-rate swap, the side that posts (receives) collateral over time can alternate.

[38] If the collateral itself were to be exposed to changes in value over time, for instance, because it consists of bonds or cash in a different currency, these fluctuations would need to be modelled as well.

[39] The protection from CCR through variation and initial margin is still not guaranteed to be all-encompassing since extreme market moves might not be sufficiently covered.

[40] As noted previously, the former imply the latter.

[41] Given the typically much longer projection horizon compared to the market risk case, counterparty credit risk almost exclusively relies on full trade and portfolio repricing and not on sensitivity-based approaches.

[42] The exposure at t_0 is equal to the current portfolio PV plus any collateral balance—and, as such, deterministic.

[43] The integration of a non-zero 'slippage period' would demand a slightly more complex setup, which is not discussed here.

[44] An application closely linked to CCR is the calculation of various valuation adjustments ('xVA'). These combine the PFE, the counterparty default probability over time, the cost of received and posted collateral and other parameters in order to adjust the 'fair value' of a portfolio PV and allow the hedging of related changes. Proposals for a quantum implementation on the example of 'CVA', the Credit Valuation Adjustment, are provided in [2, 92].

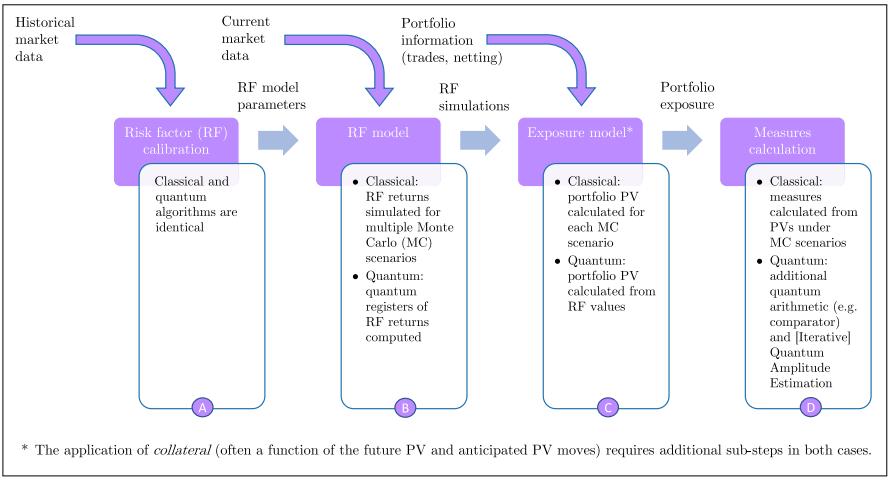


Fig. 9 Components of a counterparty credit risk calculation—comparing classical and quantum implementations. The components of the counterparty credit risk calculation are broadly analogous to those in the market risk case (see Fig. 1). Again, the main implementation difference lies in the move from Monte Carlo scenarios to working directly with random variables in the form of qubit registers

loosely interpret CCR as a through-time variant of market risk, with a focus on the positive rather than the negative tail of the P&L distributions.

The following exploration builds on the market risk example from Sect. 4. In this case, let the portfolio consist of bought call options on each of the two stocks, MSFT and AAPL, and both maturing at time $T = 2$ (years from now). The respective stock prices over time shall be denoted by $S_{k,t}$, $k = 1, 2$. The options give the holder the right to purchase the underlying stock at T and a pre-determined price ('strike') K. If the stock price ends up above K at maturity, the option will be 'in-the-money' and can be exercised against the seller accordingly.[45] For simplicity, let $K = 0$ for both options here, which means that they are always exercised at maturity and, importantly, that their PVs are equal to $S_{k,t}$ for all t. With risk factors and instrument values coinciding in this case, the revaluation over time (see step two above) can be avoided. The initial option premia are assumed to sum up to 100. They have been paid by the buyer, and collateral (variation margin) in the same amount has been posted by the seller; this hence renders the initial exposure $E_0 = 0$. No further collateral is exchanged throughout the life of the instruments. The PFE projection interval $[0, T]$ is reflected by $N = 2$ equidistant time steps of length T/N.

Figure 9 provides a sketch of the components required for a counterparty credit risk calculation and how classical and quantum computer implementations differ.

[45] This exercise can entail buying the stock at K and immediately selling it in the market to lock in a positive amount, or the seller paying the difference between the market price of the stock and the strike directly to the seller ('cash settlement').

5.2 Risk factor model

The risk drivers behind an exposure calculation need to be evolved in a dependent ('correlated') way, as for market risk applications. The historical calibration period in practice usually stretches over three years or more. A particular challenge in CCR modelling is to capture the long-term evolution of risk drivers and their dependence.

For the postulated example portfolio composed of call options, the two stock prices are the only relevant risk drivers.[46] Assuming that these evolve according to (4) as in the market risk case, the means ('drifts') $\boldsymbol{\mu}$ again amount to zero for both stocks. Recalling that the covariance matrix $\boldsymbol{\Sigma}$ in Sect. 4.2.3 is based on ten-day returns, its values are scaled to $\boldsymbol{\Sigma}' = \boldsymbol{\Sigma} \cdot \frac{252}{10}$ in order to reflect an annual horizon.

5.3 Exposure calculation

The starting point is given by $E_0 = 0$. With a time horizon of $T = 2$ (years) and $N = 2$ time points for the process discretisation, the PFE for the options portfolio from the buyer's perspective is to be determined at $t = 1$ and $t = 2(= T)$. In each of these periods, the joint stock returns are governed by $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}')$. This, in turn, describes the PV evolution of the two options over time and, as such, determines the counterparty exposure for the option holder.

Ignoring a flooring to zero for better illustration, the CDFs of the respective exposures E_t are described by the distributions of PVs, including the initial collateral. The 90%, 95% and 99% quantiles serve as dedicated exposure measures.

5.4 Setup for a quantum computer

The approach follows the technique for measuring credit risk in [91], extended to a multi-period setup.[47] The steps to reflect the task on a quantum computer consist in

- loading the probability distributions (see Sect. 3.4);
- adding the distributions across time steps;[48]
- using (Iterative) Quantum Amplitude Estimation (see Sect. 3.3) to determine the values of the CDF and, from that, the exposure quantiles.

For the two-dimensional risk factor distribution, assume that each of the marginal stock return distributions is discretised using q qubits. Repeating this over all N time steps results in $2qN$ qubits required for the quantum setup. The addition of the distributions across time steps as well as the final IQAE step demand additional qubits. In Fig. 10, the case of $q = 2$ and $N = 2$ is depicted in a stylised circuit diagram. Figure 11 provides a snapshot of the first few gates when preparing the actual quantum circuit scheme. In a two-time step setup and again using only elementary

[46] If the strikes of the options were to be different from zero, other risk factors could become relevant, such as the 'implied volatility'; see [47] for more background information.

[47] For ways to implement discrete (Markov) processes on a quantum computer, see the discussions in [10, 68]. In [95], the implementation of continuous-time stochastic processes is examined.

[48] In a more general setup, one would typically have additional revaluation steps at each risk horizon.

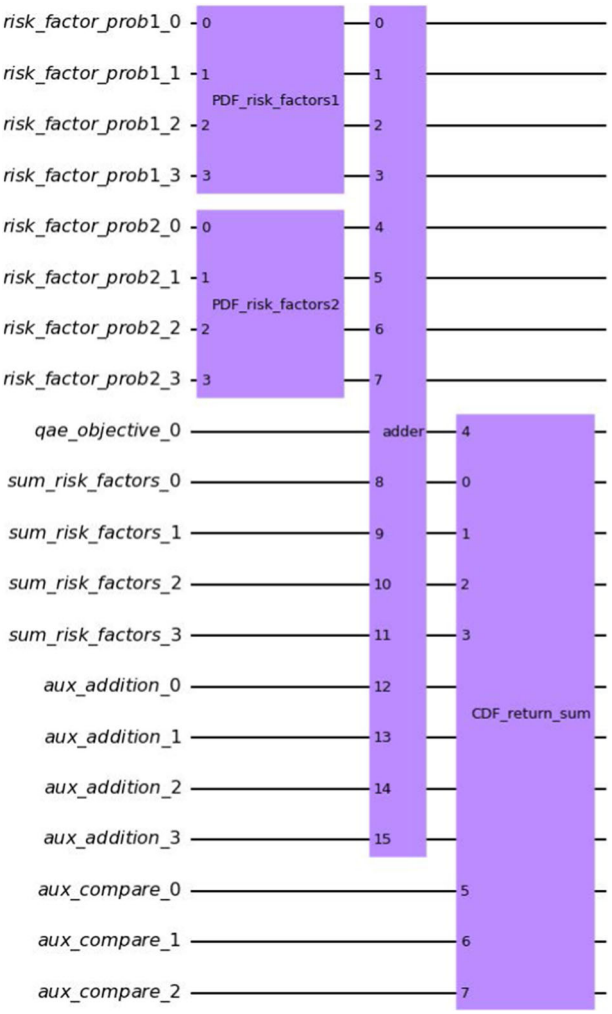


Fig. 10 Stylised quantum circuit for counterparty credit risk. The setup shows a two-period setting. The first block of qubits (*risk_factor_prob1*) reflects the two risk factors, discretised with two qubits each. The second block (*risk_factor_prob2*) is the equivalent set for the second period. *qae_objective* is the objective qubit required for the (iterative) quantum amplitude estimation. *sum_factor_risk_factors* serve for the addition of the risk factor distributions. The final *CDF_return_sum* are used to evaluate the CDF at a given point. *aux_addition* and *aux_compare* are the required ancillaries for the circuit

rotation and conditional NOT gates, the circuit depth reads 1,015 ($q = 2$ per risk factor) and 2,237 ($q = 3$), respectively.

Note that it is not obvious how to create a single quantum circuit that can provide measures for distributional quantiles at several time points as an output. A pragmatic way to generate exposure figures over multiple forecasting horizons is to use separate quantum circuits for each, which can be generated algorithmically in a straightforward manner.

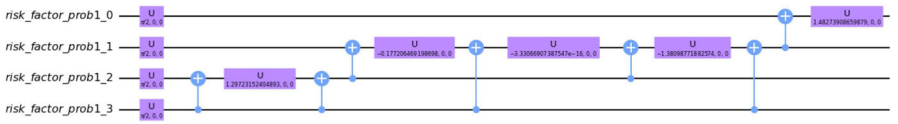


Fig. 11 Quantum circuit for counterparty credit risk—gate decomposition. The graph shows the first set of gates behind the loading of the probability distribution for the first risk factor, as per Fig. 10. $U(\theta, \phi, \lambda)$ standards for single-qubit gate and its Euler rotation angles. In the given case, $U(\theta, 0, 0)$ represents rotations along the Y-axis with angle θ. The two-qubit '+' operations are conditional NOT ('CNOT') gates. Using only these two elementary building blocks, the entire circuit with its 20 qubits (of which twelve are 'state qubits') has a depth of 1,015—a number that typically also depends on the optimisation during the transpilation.

5.5 Estimation results

The main output from the estimation is the CDF of the exposure at a given time point. Results are again obtained with the help of a quantum simulator. In Fig. 12, the classically derived CDF are compared to those from the quantum computation; for the latter, $q = 2$ or $q = 3$ qubits are used for each of the risk factor distributions.[49] In the given example, the 'correct' CDF is depicted (right scale), together with its 90% quantile as a typical risk quantity in the CCR context.

At both $t = 1$ and $t = 2$ (years), when using only $q = 2$ qubits per risk factor, the resulting CDF is very far off (left scale). This is driven by the fact that the risk factor distributions are discretised only very coarsely (four points each). With $q = 3$ qubits per risk factor (eight points each), the approximation is much better. In the latter case, the deviation from the correct CDF is well below 0.05 across the range for both projection horizons.

Table 2 summarises how the CDF estimations translate into actual exposure figures. At inception at $t = 0$, the exposure is zero. The table provides the PFE at both time horizons, $t = 1$ and $t = 2$, and for the 90%, 95% and 99% quantile.

The results from the classical computation—here obtained as the sum of two multivariate normal distributions and hence available in closed form—are shown in Panel A. These figures reflect scenarios where the (highly correlated) stock prices increase substantially, giving rise to high PVs of the options in the portfolio. In spite of the initial collateral exchange, a counterparty default could result in a sizable undercollateralisation and associated replacement costs for the option holder.

As evidenced in Panel B, in line with results for the CDF in Fig. 12, using only two qubits per risk factor leads to significant errors in the exposure estimates.[50] This renders those essentially unusable in practice. The case of three qubits per factor limits the PFE errors to about ±10%, which is more tolerable and reaches a level of approximation that might be suitable for practical applications in risk measurement. For example, the actual 90% quantile one and two years ahead reads 82 and 134, respectively. The quantum equivalents with three qubits per risk factor are 75 and 127, i.e. off by 9% and 5%.

[49] The parameters for the IQAE as per [41] are once again chosen as $\alpha = 0.05$ and $\varepsilon = 0.01$.

[50] As in the market risk example from Sect. 4, the 'error' is defined as the relative (unsigned) difference between the estimated and the correct value.

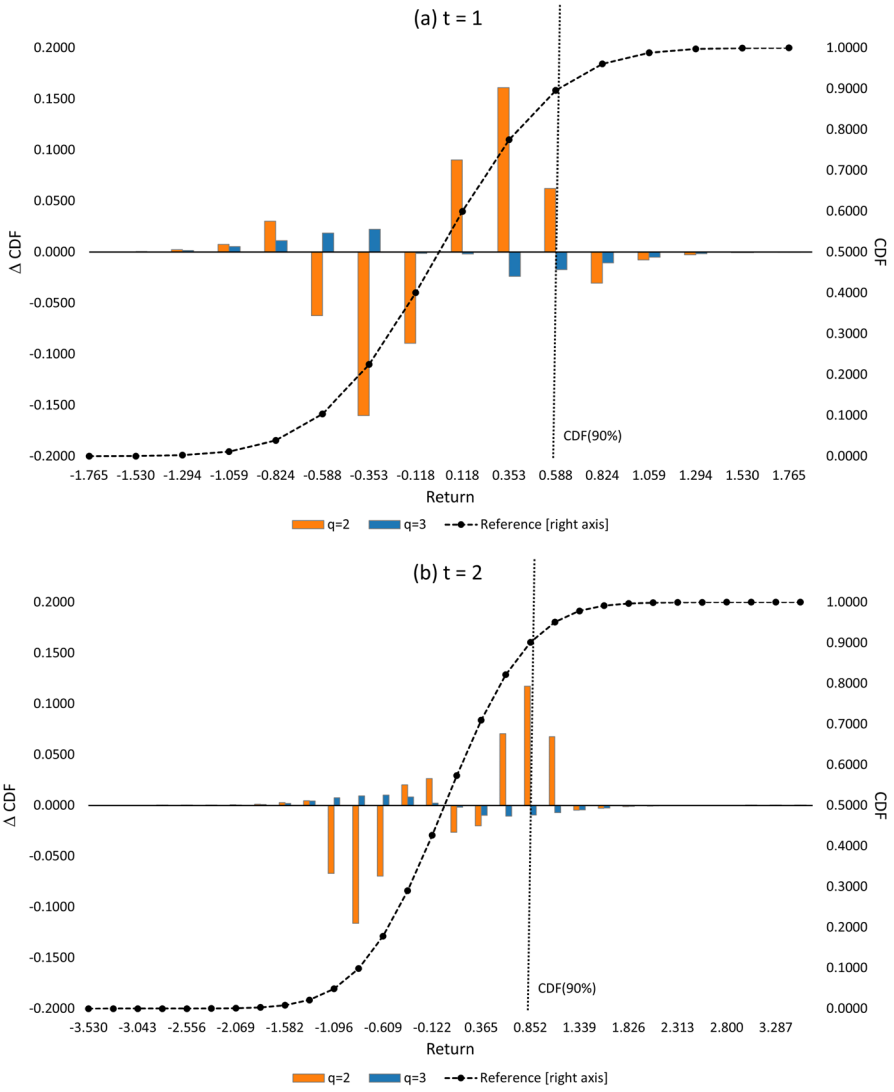


Fig. 12 Distribution functions for the counterparty credit risk example. For two time horizons, (a) $t = 1$ and (b) $t = 2$ (years), the figures compare the CDF of the PFE between classical and quantum computation. For the example at hand, the results for the classical case are known in closed form (dotted CDF, right scale). The quantum estimations are obtained by using $q = 2$ or $q = 3$ qubits to reflect each of the distributions; notably, the actual quantum circuits require a series of additional (ancillary) qubits to execute the required operations. The figures show the differences in CDF across the distribution range (left scale). The 90% quantile, as a typical quantity to measure counterparty credit risk exposure, is marked explicitly (vertical dotted line). Confidence intervals around the quantum estimates, obtained by applying IQAE with $2^{10} = 1{,}024$ shots each, are very tight and hence omitted in the figures. Notably, they are of much lower magnitude than the differences in the CDFs themselves.

Table 2 Estimation results for the counterparty credit risk example

	PFE quantile					
	t=1			t=2		
	90%	95%	99%	90%	95%	99%
A. Classical computation						
Closed-form	82	116	197	134	197	367
B. Quantum computation—simulator						
$q = 2 \cdot 2$	93 (14%)	104 (10%)	113 (43%)	211 (72%)	252 (34%)	288 (25%)
$q = 2 \cdot 3$	75 (9%)	109 (6%)	178 (10%)	127 (5%)	186 (5%)	343 (6%)

The figures compare the PFE at various quantiles between classical (Panel A) and quantum computation (Panel B). For the example at hand, the results for the classical case are known in closed form. Two time horizons, $t = 1$ and $t = 2$ (years), are considered. With two risk factors determining the PFE here, the quantum estimations are obtained by using $q = 2$ and $q = 3$ qubits to reflect each of the distributions; notably, the actual quantum circuits require a series of additional ancillary qubits to execute the required operations. Panel B reports the estimates from applying IQAE, with $2^{10} = 1{,}024$ shots each, on a quantum simulator. Confidence intervals are very tight and hence omitted in the table. Relative unsigned differences between the quantum-derived figures and the reference values in Panel A are shown in parentheses

6 Influence of quantum noise

The analyses in Sects. 4 and 5 assume ideal, not yet available quantum hardware. Besides the discussed loss of accuracy from a limited number of qubits, quantum circuits experience 'noise' in practice. This results from quantum processors being susceptible to their environment and losing their quantum state due to quantum decoherence [35, 48]. The theoretical quadratic speed-up compared to classical Monte Carlo simulation is therefore highly dependent on whether this quantum error can be contained and ideally eliminated in future [5]. Until this is achieved, quantum noise needs to be factored into any results obtained from circuits like those proposed here for risk measurement. Otherwise, certain conclusions would not be valid: for instance, even when adding more and more qubits to a system, the measurement of financial risks could still be of no use if noise levels distorted the resulting figures.

While the aspect of quantum noise lends itself to further in-depth research, at least an approximate impact analysis is required for the cases discussed here. Understanding the sensitivity of the measurements to typical noise levels sheds light on how much noise can be tolerated to render results accurate enough for practical applications.

The magnitude of quantum noise on current state-of-the-art gate-based devices amounts to about 0.1% for each gate, somewhat varying, for example, based on how many qubits a gate is applied to [39]. In the following, all one- and two-qubit gates in the quantum circuit as well as the final measurement are subjected to these noise levels to form a simplified 'noise model'. Practically, this entails flipping each bit with the given probability on every affected operation in the circuit. Continuing the work

on a quantum simulator,[51] the model is applied to the example for counterparty credit risk from Sect. 5.[52]

Using $q = 3$ qubits per risk factor and time point $t = 1$, Fig. 13a compares the CDF for the PFE in the case of no noise as well as with the defined noise model. It is visually evident that the resulting perturbation can have a non-negligible influence on the risk measurement. In Fig. 13b, the noise level is varied around the base case of 0.1% with a factor of 2^i, $i = -3, -2, \ldots, +3$; from this, the resulting relative errors in the CDF quantiles relative to the 'noise-free' case are calculated. For instance, the 90% quantile in the base case with 0.1% of noise exhibits a relative error of about 15%. For lower noise levels, the relative error in the quantile(s) tapers down, while it increases substantially in the case of more pronounced noise.

According to Table 2, even for the promising case with $q = 3$ qubits per risk factor, the typical noise levels distort the estimations and—given relative errors easily exceeding $\pm 25\%$—render them essentially unusable.

7 Conclusion and outlook

The study investigates the feasibility of implementing real-world risk measurement applications on quantum computers. Typical use cases in financial institutions such as banks require large-scale solutions with usually several thousands of risk factors. The core challenge is to project those at one or over several future time points and revalue portfolios of instruments accordingly. Extending previous research, this paper contributes a feasibility analysis with respect to running realistic market and counterparty credit risk applications on quantum devices.

On a conceptual level, modelling risk factors and simulating their joint behaviour that ultimately determines the relevant distributions for risk measurement is feasible. The single-horizon nature of a market risk measure renders the setup more accessible than for the case of counterparty credit risk, which demands multiple, longer-term projections. Moving away from idealised normal distributions and potentially even a linear dependence structure amongst risk factors is achievable.

The central limiting aspect from a practical angle is the far too-limited capacity in terms of available qubits on real-world devices that would be necessary to reflect the large set of required risk factors with sufficient accuracy and across multiple forecasting horizons. Specific optimisations such as in-place additions of risk factors are possible, but this does not materially address the limitation. A secondary though still important aspect is quantum noise: even for basic examples of VaR and PFE calculations, the accuracy of risk estimations is quickly downgraded beyond a tolerable level. Loading realistic probability distributions to quantum registers and representing non-trivial payoffs and pricing functions are additional obstacles.

In the current 'noisy intermediate-scale quantum' (NISQ) era, the path to real-world applications in risk measurement depends heavily on progress in the various active

[51] Running a circuit on actual hardware (e.g. IBM devices) could allow a slightly more realistic 'noise pattern', but the results would ultimately be qualitatively similar.

[52] An application of the noise pattern to the more complex market risk example from Sect. 4 is computationally too expensive when run on a quantum simulator.

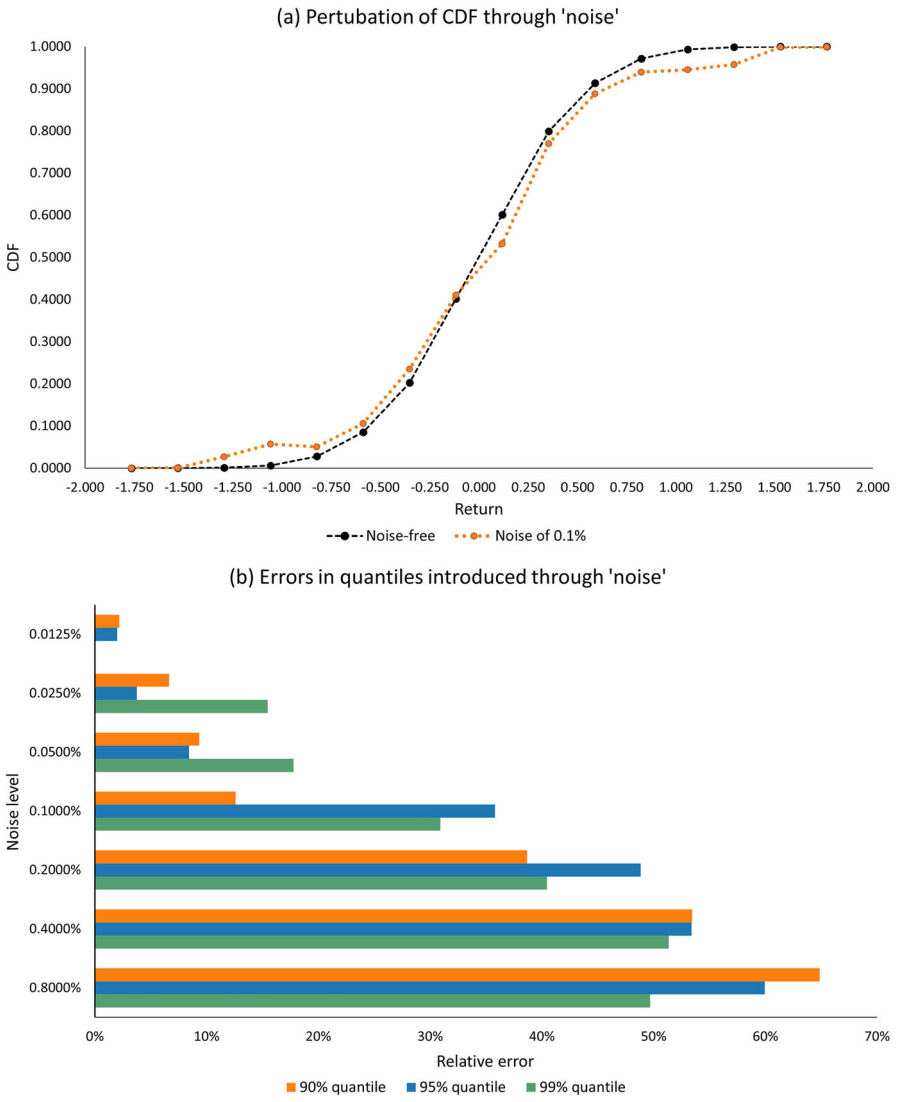


Fig. 13 Influence of 'noise' in the counterparty credit risk example. When adding 'noise' of 0.1% to all one- and two-qubit gates in the quantum circuit as well as for the final measurement, **a** shows the influence on the CDF for a horizon of $t = 1$ (years) and $q = 3$ (qubits) for both risk factors. **b** illustrates how different levels of noise translate into errors in the estimation of typical PFE quantiles

areas of quantum research. Besides the hardware and its accessibility, error control and mitigation are essential topics to be addressed. Hereby, the trade-off between model and circuit complexity on one side and acceptable measurement error tolerance on the other side need to be operationalised for practical use. The optimal interaction between CPU, GPU and QPU is another field for further exploration, specifically for risk applications. In [89], three critical attributes for quantum computing performance are put forward—

quality (measured by quantum volume), speed (measured by circuit layer operations per second), and scale (measured by number of qubits). The ultimate aim is to define holistic benchmarks that capture all system components, are appropriate for real-world applications, without being too cumbersome to execute and difficult to interpret.

Overall, the analysis evidences that the number of qubits available on current hardware, and even within quantum simulators, is still far too low to render an actual business application viable. This constraint might be eased in the not-too-distant future, however, based on claims that hardware with more than 1,000 qubits are deemed achievable by 2023 [34]. Nevertheless, only once the stated target of 1,000,000 qubits on a fault-tolerant quantum computer by 2029 is reached [17], at least certain computationally intensive algorithms and applications from the financial engineering world might be routinely handled with quantum hardware.

Given the comparatively new technology, applying quantum computing—especially in established areas such as risk measurement—will face certain scepticism at the beginning. This is even more relevant when also applied for calculating regulatory capital requirements and, as such, demanding approval from supervisors. The initial enthusiasm by banks is already wearing a bit thin meanwhile, given that ready-to-use technology is still a long way off [22]. The business case from a computational angle is not very strong in areas such as market and counterparty credit risk since running 'traditional' calculations even in large financial institutions is currently a matter of minutes or at most a few hours. A quadratic speed-up of Monte Carlo simulations is hence not removing major obstacles. The pricing of derivatives, even complex ones, has already been established in principle by now—with promising short-term feasibility [18]. However, the commercial need for a large-scale roll-out is not apparent even here, given that the speed-up is ultimately limited and that latency for non-vanilla derivatives is not a significant concern in practice. The same might not be true for so far intractable problems such as portfolio optimisation, which are therefore more suitable contenders for a first wave of real-world (risk) applications.

Data availability The stock prices used in the study are publicly available (*Yahoo! Finance*).

References

1. Albareti, F.D., Ankenbrand, T., Bieri, D., Hänggi, E., Lötscher, D., Stettler, S., Schöngens, M.: A Structured Survey of Quantum Computing for the Financial Industry. Working paper, BME Inntech, April (2022). arXiv:2204.10026 [q-fin]
2. Alcazar, J., Cadarso, A., Katabarwa, A., Mauri, M., Peropadre, B., Wang, G., Cao, Y.: Quantum algorithm for credit valuation adjustments. New J. Phys. **24**(2), 023036 (2022)

3. Alcazar, J., Leyton-Ortega, V., Perdomo-Ortiz, A.: Classical versus quantum models in machine learning: insights from a finance application. Mach. Learn.: Sci. Technol. **1**(3), 035003 (2020)
4. An, D., Linden, N., Liu, J.-P., Montanaro, A., Shao, C., Wang, J.: Quantum-accelerated multilevel Monte Carlo methods for stochastic differential equations in mathematical finance. Quantum **5**, 481 (2021)
5. Babbush, R., McClean, J.R., Newman, M., Gidney, C., Boixo, S., Neven, H.: Focus beyond quadratic speedups for error-corrected quantum advantage. PRX Quant. **2**, 010103 (2021)
6. Barkoutsos, P., Nannicini, G., Robert, A., Tavernelli, I., Wörner, S.: Improving variational quantum optimization using CVaR. Quantum **4**(256), 1 (2020)
7. Benedetti, M., Lloyd, E., Sack, S., Fiorentini, M.: Parameterized quantum circuits as machine learning models. Quant. Sci. Technol. **4**(4), 043001 (2019)
8. Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., Lloyd, S.: Quantum machine learning. Nature **549**(7671), 195–202 (2017)
9. Black, F., Scholes, M.: The pricing of options and corporate liabilities. J. Polit. Econ. **81**, 637–654 (1973)
10. Blank, C., Park, D.K., Petruccione, F.: Quantum-enhanced analysis of discrete stochastic processes. NPJ Quant. Inf. **7**(126), 1 (2021)
11. Bobier, J.-F., Binefa, J.-M., Langione, M., Kumar, A.: It's Time for Financial Institutions to Place Their Quantum Bets. Boston Consulting Group, Report, Boston Consulting Group, October (2020)
12. Bouland, A., Dam, W. van, Joorati, H., Kerenidis, I., Prakash, A.: Prospects and Challenges of Quantum Finance. Working paper, QC Ware Corporation, November (2020). arXiv:2011.06492 [q-fin.CP]
13. Braine, L., Egger, D.J., Glick, J., Wörner, S.: Quantum Amplitude Amplification and Estimation. IEEE Trans. Quant. Eng. **2**, 1–8 (2021)
14. Brassard, G., Hoyer, P., Mosca, M., Tapp, A.: Quantum Amplitude Amplification and Estimation. Working paper, University of Montreal, May (2000). arXiv:quant-ph/0005055
15. Braun, M., Decker, T., Hegemann, N., Kerstan, S., Schaefer, C.: A Quantum Algorithm for the Sensitivity analysis of Business Risks. Working paper, JoS Quantum GmbH, March (2021). arXiv:2103.05475 [quant-ph]
16. Carrera Vazquez, A., Wörner, S.: Efficient state preparation for quantum amplitude estimation. Phys. Rev. Appl. **15**, 034027 (2021)
17. Castellanos, S.: Google Aims for Commercial-Grade Quantum Computer by 2029. The Wall Street Journal, May (2021)
18. Chakrabarti, S., Krishnakumar, R., Mazzola, G., Stamatopoulos, N., Wörner, S., Zeng, W.J.: A threshold for quantum advantage in derivative pricing. Quantum **5**(463), 1 (2021)
19. Clancy, L.: Barclays (and others) strive for machine learning at quantum speed. Risk.net, March (2022)
20. Clancy, L.: Dutch banks seek quantum edge for stress tests. Risk.net, October (2020)
21. Clancy, L.: Quantum kit offers HFTs '100-fold' speed boost. Risk.net, June (2021)
22. Clancy, L.: Science friction: some tire of waiting for quantum's leap. Risk.net, October (2020)
23. Coyle, B., Henderson, M., Le, J.C.J., Kumar, N., Paini, M., Kashefi, E.: Quantum versus classical generative modelling in finance. Quant. Sci. Technol. **6**(2), 024013 (2021)
24. Cruise, J.R., Gillespie, N.I., Reid, B.: Practical Quantum Computing: the Value of Local Computation. Working paper, Riverlane, September (2020). arXiv:2009.08513 [quant-ph]
25. Dasgupta, K., Paine, B.: Loading Probability Distributions in a Quantum Circuit. Working paper, IBM Research, August (2022). arXiv:2208.13372 [quant-ph]
26. Dasgupta, S., Banerjee, A.: Quantum Annealing Algorithm for Expected Shortfall Based Dynamic Asset Allocation. Working paper, Oak Ridge National Laboratory (2020). arXiv:1909.12904 [q-fin.RM]
27. Doriguello, J.F., Luongo, A., Bao, J., Rebentrost, P., Santha, M.: Quantum Algorithm for Stochastic Optimal Stopping Problems. Working paper, National University of Singapore, November (2021). arXiv:2111.15332 [quant-ph]
28. Dupire, B.: Pricing with a smile. Risk **7**, 18–20 (1994)
29. Egger, D.J., Gambella, C., Marecek, J., McFaddin, S., Mevissen, M., Raymond, R., Simonetto, A., Wörner, S., Yndurain, E.: Quantum computing for finance: state-of-the-art and future prospects. IEEE Trans. Quant. Eng. **1**, 1–24 (2020)
30. Egger, D.J., Gutiérrez, R.G., Mestre, J.C., Wörner, S.: Credit risk analysis using quantum computers. IEEE Trans. Comput. **70**(12), 2136–2145 (2020)

31. Emmanoulopoulos, D., Dimoska, S.: Quantum Machine Learning in Finance: Time Series Forecasting. Working paper, Barclays, February (2022). arXiv:2202.00599 [quant-ph]
32. Fedorov, A., Gisin, N., Beloussov, S., Lvovsky, A.: Quantum Computing at the Quantum Advantage Threshold: a Down-to-business Review. Working paper, Schaffhausen Institute of Technology, March (2022). arXiv:2203.17181 [quant-ph]
33. Fontanela, F., Jacquier, A., Oumgari, M.: A quantum algorithm for linear PDEs arising in finance. SIAM J. Financ. Math. **12**(4), SC98–SC114 (2021)
34. Gambetta, J.: IBM's Roadmap for Scaling Quantum Technology. Research blog, IBM Research, September (2020)
35. Georgopoulos, K., Emary, C., Zuliani, P.: Modeling and simulating the noisy behavior of near-term quantum computers. Phys. Rev. A **104**, 062432 (2021)
36. Giurgica-Tiron, T., Kerenidis, I., Labib, F., Prakash, A., Zeng, W.: Low depth algorithms for quantum amplitude estimation. Quantum **6**, 745 (2022)
37. Gòmez, A., Leitao, A., Manzano, A., Musso, D., Nogueiras, M.R., Ordónez, G., Vózquez, C.: A survey on quantum computational finance for derivatives pricing and VaR. Arch. Comput. Methods Eng. **29**, 4137–4163 (2022)
38. Gonzalez-Conde, J., Rodríguez-Rozas, A., Solano, E., Sanz, M.: Pricing Financial Derivatives with Exponential Quantum Speedup. Working paper, University of the Basque Country, February (2022). arXiv:2101.04023 [quant-ph]
39. Google Quantum, A.I.: Exponential suppression of bit or phase errors with cyclic error correction. Nature **595**, 383–387 (2021)
40. Gregory, J.: The xVA Challenge: Counterparty Risk, Funding, Collateral, Capital and Initial Margin, 4th edn. Wiley & Sons, Chichester (2020)
41. Grinko, D., Gacon, J., Zoufal, C., Wörner, S.: Iterative quantum amplitude estimation. NPJ Quant. Inf. **7**(52), 1 (2021)
42. Grover, L., Rudolph, T.: Creating Superpositions that Correspond to Efficiently Integrable Probability Distributions. Working paper, Bell Labs, August (2002). arXiv:quant-ph/0208112
43. Herbert, S.: Quantum Monte-Carlo Integration: The Full Advantage in Minimal Circuit Depth. Working paper, Cambridge Quantum Computing, September (2022). arXiv:2105.09100 [quant-ph]
44. Herman, D.A., Googin, C., Liu, X., Galda, A., Safro, I., Sun, Y., Pistoia, M., Alexeev, Y.: A Survey of Quantum Computing for Finance. Working paper, JPMorgan Chase Bank, June (2022). arXiv:2201.02773 [quant-ph]
45. Hidary, J.D.: Quantum Computing: An Applied Approach, 2nd edn. Springer Nature, Cham (2021)
46. Hodson, M., Ruck, B., Ong, H. H. C., Garvin, D., Dulman, S.: Portfolio Rebalancing Experiments Using the Quantum Alternating Operator Ansatz. Working paper, Rigetti Computing, November (2019). arXiv:1911.05296 [quant-ph]
47. Hull, J.C.: Options, Futures, and Other Derivatives, 11th edn. Pearson, New York (2021)
48. Johnstun, S., Van Huele, J.-F.: Understanding and compensating for noise on IBM quantum computers. Am. J. Phys. **89**(10), 935–942 (2021)
49. Jorion, P.: Value at Risk: The New Benchmark for Managing Financial Risk, 3rd edn. McGraw-Hill, New York (2006)
50. Kaneko, K., Miyamoto, K., Takeda, N., Yoshino, K.: Quantum speedup of Monte Carlo integration with respect to the number of dimensions and its application to finance. Quant. Inf. Process. **20**(185), 1 (2021)
51. Kastroyano, M., Pancotti, N.: A Highly Efficient Tensor Network Algorithm for Multi-asset Fourier Options Pricing. Working paper, Amazon Quantum Solutions Lab, March (2022). arXiv:2203.02804 [quant-ph]
52. Kondratyev, A.: Non-differentiable leaning of quantum circuit born machine with genetic algorithm. Wilmott **2021**(114), 50–61 (2021)
53. Kondratyev, A., Schwarz, C.: The market generator. Risk, Feburary (2020)
54. Kubo, K., Miyamoto, K., Mitarai, K., Fujii, K.: Pricing Multi-asset Derivatives by Variational Quantum Algorithms. Working paper, Mercari Inc., July (2022). arXiv:2207.01277 [quant-ph]
55. Kurek, M.: Quantum Technologies, Patent. Publications and Investments. Report, Ecole Polytechnique, September (2020)
56. Lim, D., Rebentrost, P.: A Quantum Online Portfolio Optimization Algorithm. Working paper, National University of Singapore, August (2022). arXiv:2208.14749 [quant-ph]

57. Mari, A., Bromley, T.R., Killoran, N.: Estimating the gradient and higher-order derivatives on quantum hardware. Phys. Rev. A **103**, 012405 (2021)
58. Martin, A., Candelas, B., Rodríguez-Rozas, A., Martín-Guerrero, J.D., Chen, X., Lamata, L., Orús, R., Solano, E., Sanz, M.: Toward pricing financial derivatives with an IBM quantum computer. Phys. Rev. Res. **3**, 013167 (2021)
59. McNeil, A.J., Frey, R., Embrechts, P.: Quantitative Risk Management: Concepts. Techniques and Tools. Revised edition, Princeton University Press, Princeton (2015)
60. Milek, J.: Quantum Implementation of Risk Analysis-relevant Copulas. Working paper, ETH Zurich, March (2020). arXiv:2002.07389 [stat.ME]
61. Milne, A., Rounds, M., Goddard, P.: Optimal Feature Selection in Credit Scoring and Classification using a Quantum Annealer. Working paper, 1QB Information Technologies, April (2017)
62. Miyamoto, K.: Bermudan option pricing by quantum amplitude estimation and Chebyshev interpolation. EPJ Quant. Technol. **9**, 3 (2022)
63. Miyamoto, K.: Quantum Algorithm for Calculating Risk Contributions in a Credit Portfolio. Working paper, Osaka University, January (2022). arXiv:2201.11394 [quant-ph]
64. Miyamoto, K., Kubo, K.: Pricing multi-asset derivatives by finite-difference method on a quantum computer. IEEE Trans. Quant. Eng. **3**, 1–25 (2022)
65. Miyamoto, K., Shiohara, K.: Reduction of qubits in a quantum algorithm for Monte Carlo simulation by a pseudo-random-number generator. Phys. Rev. A **102**, 022424 (2020)
66. Nakaji, K.: Faster amplitude estimation. Quant. Inf. Comput. **20**(13&14), 1109–1122 (2020)
67. Nelsen, R.B.: An Introduction to Copulas, 2nd edn. Princeton University Press, New York (2007)
68. Nikolov, P., Galabov, V.: Markov Process Simulation on a Real Quantum Computer. Working paper, University of Sofia, November (2019)
69. Orús, R., Mugel, S., Lizaso, E.: Quantum computing for finance: overview and prospects. Rev. Phys. **4**, 1–12 (2019)
70. Orús, R., Mugel, S., Lizaso, E.: Forecasting financial crashes with quantum computing. Phys. Rev. A **99**, 060301 (2019)
71. Pistoia, M., Ahmad, S.F., Ajagekar, A., Buts, A., Chakrabarti, S., Herman, D., Hu, S., Jena, A., Minssen, P., Niroula, P., Rattew, A., Sun, Y., Yalovetzky, R.: Quantum Machine Learning for Finance. Working paper, JP Morgan Chase Bank, September (2021). arXiv:2109.04298 [quant-ph]
72. Radha, S.K.: Quantum Option Pricing Using Wick Rotated Imaginary Time Evolution. Working paper, Case Western Reserve University, January (2021). arXiv:2101.04280 [quant-ph]
73. Ramos-Calderer, S., Pérez-Salinas, A., García-Martín, D., Bravo-Prieto, C., Cortada, J., Planagumà, J., Latorre, J.I.: Quantum unary approach to option pricing. Phys. Rev. A **103**, 032414 (2021)
74. Rao, P., Yu, K., Lim, H., Jin, D., Choi, D.: Quantum Amplitude Estimation Algorithms on IBM Quantum Devices. Working paper, Stony Brook University, August (2020). arXiv:2008.02102 [quant-ph]
75. Rebentrost, P., Gupt, B., Bromley, T.R.: Quantum computational finance: Monte Carlo pricing of financial derivatives. Phys. Rev. A **98**, 022321 (2018)
76. Rebentrost, P., Lungo, A., Bosch, S., Lloyd, S.: Quantum Computational Finance: Martingale Asset Pricing for Incomplete Markets. Working paper, National University of Singapore, September (2022). arXiv:2209.08867 [quant-ph]
77. Sakuma, T.: Application of Deep Quantum Neural Networks to Finance. Working paper, Soka University, December (2020). arXiv:2011.07319 [q-fin.CP]
78. Sakuma, T.: Quantum Circuit Learning to Compute Option Prices and Their Sensitivities. Working paper, Soka University, September (2021)
79. Shende, V.V., Bullock, S.S., Markov, I.L.: Synthesis of quantum-logic circuits. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **25**(6), 1000–1010 (2006)
80. Stamatopoulos, N., Egger, D.J., Sun, Y., Zoufal, C., Iten, R., Shen, N., Wörner, S.: Option pricing using quantum computers. Quantum **4**(291), 1 (2020)
81. Stamatopoulos, N., Mazzola, G., Woerner, S., Zeng, W.J.: Towards quantum advantage in financial market risk using quantum gradient algorithms. Quantum **6**, 770 (2022)
82. Suzuki, Y., Uno, S., Raymond, R., Tanaka, T., Onodera, T., Yamamoto, N.: Amplitude estimation without phase estimation. Quant. Inf. Process. **19**(75), 1 (2020)
83. Tang, H., Pal, A., Wang, T.-Y., Qiao, L.-F., Gao, J., Jin, X.-M.: Quantum computation for pricing the collateralized debt obligations. Quant. Eng. **3**(4), e84 (2021)
84. Tang, H., Wu, W., and Jin, X.-M.: Quantum Computation for Pricing Caps using the LIBOR Market Model. Working paper, Shanghai Jiao Tong University, July (2022). arXiv:2207.01558 [quant-ph]

85. Uhlenbeck, G., Ornstein, L.: On the theory of the Brownian Motion. Phys. Rev. **36**, 823–841 (1930)
86. Vedral, V., Barenco, A., Ekert, A.: Quantum networks for elementary arithmetic operations. Phys. Rev. A **54**(1), 147–153 (1996)
87. Venturelli, D., Kondratyev, A.: Beyond Markowitz with Quantum Annealing. Risk, June (2019)
88. Veselý, M.: Application of Quantum Computers in Foreign Exchange Reserves Management. Working paper, Czech National Bank, March (2022). arXiv:2203.15716 [econ.GN]
89. Wack, A., Paik, H., Javadi-Abhari, A., Jurcevic, P., Faro, I., Gambetta, J.M., Johnson, B.R.: Quality, Speed, and Scale: Three Key Attributes to Measure the Performance of Near-term Quantum Computers. Working paper, IBM Quantum, October (2021). arXiv:2110.14108 [quant-ph]
90. Waters, R.: Wall Street banks ramp up research into quantum finance. Financial Times, January (2020)
91. Wörner, S., Egger, D.J.: Quantum risk analysis. NPJ Quant. Inf. **5**(15), 1 (2019)
92. Yu Han, J., Rebentrost, P.: Quantum Advantage for Multi-option Portfolio Pricing and Valuation Adjustments. Working paper, Centre for Quantum Technologies, Singapore, March (2022). arXiv:2203.04924 [quant-ph]
93. Zhu, D., Shen, W., Giani, A., Majumder, S. R., Neculaes, B., Jori, S.: Copula-based Risk Aggregation with Trapped Ion Quantum Computers. Working paper, IonQ Inc., June (2022). arXiv:2206.11937 [quant-ph]
94. Zhuang, X.-N., Chen, Z.-Y., Wu, Y.-C., Guo, G.-P.: Quantum computational quantitative trading: high-frequency statistical arbitrage algorithm. New J. Phys. **24**(7), 073036 (2022)
95. Zhuang, X.-N., Chen, Z.-Y., Xue, C., Wu, Y.-C., and Guo, G.-P.: Quantum Encoding and Analysis on Continuous Stochastic Process. Working paper, Origin Quantum Computing, August (2022). arXiv:2208.02364 [quant-ph]
96. Zoufal, C., Lucchi, A., Wörner, S.: Quantum generative adversarial networks for learning and loading random distributions. NPJ Quant. Inf. **5**(103), 1 (2019)
97. Zoufal, C., Lucchi, A., Wörner, S.: Variational quantum Boltzmann machines. Quant. Mach. Intell. **3**(7), 1 (2021)