Credit Risk Analysis using Quantum Computers

Daniel J. Egger,[1] Ricardo García Gutiérrez,[2] Jordi Cahué Mestre,[2] and Stefan Woerner[1,*]

[1] *IBM Research – Zurich*
[2] *IBM Spain*

(Dated: July 9, 2019)

We present and analyze a quantum algorithm to estimate credit risk more efficiently than Monte Carlo simulations can do on classical computers. More precisely, we estimate the economic capital requirement, i.e. the difference between the Value at Risk and the expected value of a given loss distribution. The economic capital requirement is an important risk metric because it summarizes the amount of capital required to remain solvent at a given confidence level. We implement this problem for a realistic loss distribution and analyze its scaling to a realistic problem size. In particular, we provide estimates of the total number of required qubits, the expected circuit depth, and how this translates into an expected runtime under reasonable assumptions on future fault-tolerant quantum hardware.

I. INTRODUCTION

Economic Capital, a key tool of risk management, is computed by financial service firms to determine the amount of risk capital that they require to remain solvant in the face of adverse yet realistic conditions [1]. Financial service firms are exposed to many forms of risk [2] such as credit risk which is the risk of a monetary loss resulting from a counterparty failing to meet a financial obligation [3, 4]. For instance, a payment may not be made in due time or at all. Risk metrics such as Value at Risk and the Economic Capital Requirement (ECR) are often calculated for many different scenarios. Monte Carlo (MC) simulations are thus the method of choice for this task.

In a MC simulation a parameter is estimated by building a distribution obtained by taking M samples from the model input distributions. The error on the resulting estimation scales as $\mathcal{O}(1/\sqrt{M})$ [5]. Evaluating credit risk with MC is a rare-event simulation problem which requires many samples thereby making MC computationally costly [6]. Importance sampling reduces the computational cost by lowering the constants but does not change the asymptotic rate of convergence.

Quantum computers process information using the laws of quantum mechanics [7]. This opens up novel ways of addressing various computational tasks. Problems that may benefit from quantum computing include quantum chemistry calculations [8, 9], machine learning [10], and finance [11–14]. Recently, it has been shown how the Quantum Amplitude Estimation (QAE) algorithm can be used to analyze financial risk measures [11] or to price financial derivatives [15] with a quadratic speedup.

In Section II, we formally define the economic capital requirement as well as the two different uncertainty models considered. In Section III, we build on previous work [11] and discuss how to implement the quantum algorithms on a gate based quantum computer. In Section IV, we show simulation results for small instances of the considered models. Section V analyzes the scaling of the algorithm for problems of realistic size as well as the resulting quantum advantage.

II. CREDIT RISK ANALYSIS

ECR summarizes in a single figure the amount of capital (or own funds) required to remain solvent at a given confidence level (usually linked to the risk appetite or target solvency rating) and a time horizon (usually one year). It is a complementary metric to the regulatory capital requirements that refers to the amount of own funds required following regulatory criteria and rules [16]. In this paper, we consider only the ECR related to default risk, which is the loss that occurs when an obligor does not fulfill the repayment of a loan. The main components of an ECR model for a portfolio of assets are the single-asset default probabilities, the loss given default, and the correlation among the single-asset default events. In the following, we first introduce a general form of the credit risk analysis problem considered in this manuscript and then define concrete models in detail.

For a portfolio of K assets the multivariate random variable $(L_1, ..., L_K) \in \mathbb{R}^K_{\geq 0}$ denotes each possible loss associated to each asset. The expected value of the total loss $\mathcal{L} = \sum_{k=1}^K L_k$ is $\mathbb{E}[\mathcal{L}] = \sum_{k=1}^K \mathbb{E}[L_k]$. The Value at Risk (VaR) for a given confidence level $\alpha \in [0, 1]$ is defined as the smallest total loss that still has a probability greater than or equal to α, i.e.,

$$\mathrm{VaR}_\alpha[\mathcal{L}] = \inf_{x \geq 0} \{x \mid \mathbb{P}[\mathcal{L} \leq x] \geq \alpha\}. \tag{1}$$

The ECR at confidence level α is thus defined as

$$\mathrm{ECR}_\alpha[\mathcal{L}] = \mathrm{VaR}_\alpha[\mathcal{L}] - \mathbb{E}[\mathcal{L}]. \tag{2}$$

Common values of α for ECR found in the finance industry are around 99.9%.

In a first model, we assume that all losses are independent and can be expressed as $L_k = \lambda_k X_k$ where $\lambda_k > 0$

* wor@zurich.ibm.com

is the loss given default (LGD) and $X_k \in \{0,1\}$ is a corresponding Bernoulli random variable. The probability that $X_k = 1$, i.e., a loss for asset k, is p_k. The expected loss of the portfolio $\mathbb{E}[\mathcal{L}] = \sum_{k=1}^{K} \lambda_k p_k$ is easier to evaluate than $\mathrm{VaR}_\alpha[\mathcal{L}]$, which usually requires a Monte Carlo simulation.

We extend this simple uncertainty model to a more realistic one, where the defaults X_k are no longer independent but follow a conditional independence scheme [17]. Given a realization z of a latent random variable $\mathcal{Z}$, the Bernoulli random variables $X_k \mid \mathcal{Z} = z$ are assumed independent, but their default probabilities p_k depend on z. We follow [17] and assume that $\mathcal{Z}$ follows a standard normal distribution and that

$$p_k(z) = F\left(\frac{F^{-1}(p_k^0) - \sqrt{\rho_k}z}{\sqrt{1-\rho_k}}\right), \tag{3}$$

where p_k^0 denotes the default probability for $z = 0$, F is the cumulative distribution function (CDF) of the standard normal distribution, and $\rho_k \in [0,1)$ determines the sensitivity of X_k to $\mathcal{Z}$. This scheme is similar to the one used for regulatory purposes in the Basel II (and following) Internal Ratings-Based (IRB) approach to credit risk [18, 19], and is called the *Gaussian conditional independence model* [17].

In order to scale the model to a larger number of assets, one can aggregate subsets of similar assets into random variables $L_k \geq 0$ that take more than two values. We briefly discuss this approach and the overall scaling of our algorithm to real world problems in Section V.

In the following sections, we show how the ECR for the presented model can be estimated on a gate-based quantum computer with QAE resulting in a quadratic speedup over classical Monte Carlo simulations.

III. QUANTUM ALGORITHM

For the models introduced in Section II, the expected total loss $\mathbb{E}[\mathcal{L}]$ can be efficiently computed classically, see Appendix A. Thus, we focus on quantum algorithms to estimate $\mathrm{VaR}_\alpha[\mathcal{L}]$. For more details on the estimation of expected values using QAE we refer to [11].

To apply QAE, we map the problem of interest to a quantum operator $\mathcal{A}$ acting on $n+1$ qubits such that:

$$\mathcal{A}\left|0\right\rangle_{n+1} = \sqrt{1-a}\left|\psi_0\right\rangle_n \left|0\right\rangle + \sqrt{a}\left|\psi_1\right\rangle_n \left|1\right\rangle, \tag{4}$$

where $a \in [0,1]$. The probability to measure $|1\rangle$ in the last qubit, i.e., a, corresponds to the (normalized) property of interest. From $\mathcal{A}$ we construct a quantum operator

$$\mathcal{Q} = \mathcal{A}\mathcal{S}_0\mathcal{A}^\dagger\mathcal{S}_{\psi_0}, \tag{5}$$

where $S_0 = \mathbb{I} - 2\left|0\right\rangle_{n+1}\left\langle 0\right|_{n+1}$ and $\mathcal{S}_{\psi_0} = \mathbb{I} - 2\left|\psi_0\right\rangle_n\left|0\right\rangle\left\langle\psi_0\right|_n\left\langle 0\right|$. Every application of $\mathcal{Q}$ corresponds to one *quantum sample*. QAE allows us to estimate a with an estimation error that is bounded by

$$\frac{2\sqrt{a(1-a)}\pi}{M} + \frac{\pi^2}{M^2} = \mathcal{O}\left(\frac{1}{M}\right), \tag{6}$$

where M corresponds to the number of quantum samples [11, 20]. QAE has a success probability of 81%, thus, by repeating only a few times and taking the median result the algorithm succeeds almost with certainty. This leads to a quadratic speedup over classical Monte Carlo simulations, where the estimation error behaves as $\mathcal{O}(1/\sqrt{M})$, where M now denotes the number of *classical samples*. A more detailed discussion of QAE can be found in Appendix B.

To estimate VaR, we use QAE to efficiently evaluate the CDF of the total loss, i.e., we will construct $\mathcal{A}$ such that $a = \mathbb{P}[\mathcal{L} \leq x]$ for a given $x \geq 0$, and apply a bisection search to find the smallest $x_\alpha \geq 0$ such that $\mathbb{P}[\mathcal{L} \leq x_\alpha] \geq \alpha$, which implies $x_\alpha = \mathrm{VaR}_\alpha[\mathcal{L}]$ [11].

Mapping the CDF of the total loss to a quantum operator $\mathcal{A}$ requires three steps. Each step corresponds to a quantum operator. First, $\mathcal{U}$ loads the uncertainty model. Second, $\mathcal{S}$ computes the total loss into a quantum register with n_S qubits. Last, $\mathcal{C}$ flips a target qubit if the total loss is less than or equal to a given level x which is used to search for VaR_α. Thus, we have $\mathcal{A} = \mathcal{C}\mathcal{S}\mathcal{U}$ and Fig. 1 illustrates the corresponding circuit on a high level.

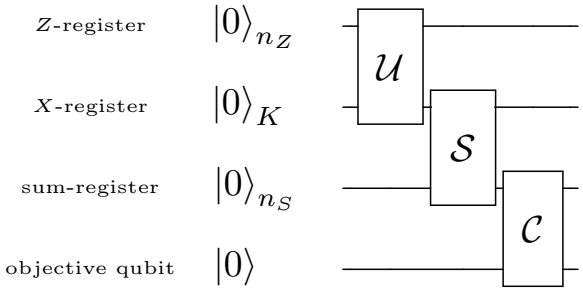

FIG. 1. High level circuit of the operator $\mathcal{A}$ used to evaluate the CDF of the total loss: the first qubit register with n_Z qubits represents $\mathcal{Z}$, the second qubit register with K qubits represents the X_k, the third qubit register with n_S qubits represents the sum of the losses, i.e., the total loss, and the last qubit is flipped to $|1\rangle$ if the total loss is less than or equal to a given x. The operators $\mathcal{U}$, $\mathcal{S}$, and $\mathcal{C}$ represent the loading of uncertainty, the summation of losses, and the comparison to a given x, respectively.

The estimation error given in Eq. (6) also depends on the exact result a. In particular, if a is close to 0 or 1 the constant in the error bound becomes very small. When computing VaR_α, we want to find the minimal threshold such that the estimated probability is larger than or equal to α. Thus, we can replace a in Eq. (6) by α to get a better error bound. When $\alpha = 99.9\%$ the error bound is approximately

$$\frac{1}{5M} + \frac{\pi^2}{M^2} \tag{7}$$

which is independent of the other properties of the problem. In other words, QAE is particularly good at estimating tail probabilities of distributions.

We now discuss the operators $\mathcal{U}$, $\mathcal{S}$, and $\mathcal{C}$ in more detail. When the default events $\{X_1, ..., X_K\}$ are uncorrelated we can encode the X_k of each asset in the state of a corresponding qubit by applying to qubit k a Y-rotation $R_Y(\theta_p^k)$ [7] with angle $\theta_p^k = 2\arcsin(\sqrt{p_k})$. Therefore the loading operator is

$$\mathcal{U} = \bigotimes_{k=1}^{K} R_Y(\theta_p^k). \tag{8}$$

This prepares qubit k in the state $\sqrt{1-p_k}\,|0\rangle + \sqrt{p_k}\,|1\rangle$ for which the probability to measure $|1\rangle$ is p_k. The $|1\rangle$ state of qubit k thus corresponds to a loss for asset k.

To adjust $\mathcal{U}$ to include correlations between the default events, we add another register with n_Z qubits to represent $\mathcal{Z}$. The random variable $\mathcal{Z}$ follows a standard normal distribution. We use a truncated and discretized approximation with 2^{n_Z} values, where we consider an affine mapping $z_i = a_z i + b_z$ from $i \in \{0, ..., 2^{n_Z} - 1\}$ to the desired range of values of $\mathcal{Z}$. Any discretized and truncated log-concave distribution, such as $\mathcal{Z}$, can be efficiently represented in a quantum register by an operator $\mathcal{U}_Z$ built from controlled rotations [21]. The qubit register representing $\mathcal{Z}$ is then used to control the rotation angles $\theta_p^k(z) = 2\arcsin(\sqrt{p_k(z)})$ that prepare the qubits representing the X_k. For simplicity, we use a first order approximation of $\theta_p^k(z)$ and include the affine mapping from z (a value of the normal distribution) to i (an integer represented by n_Z qubits), i.e., $\theta_p^k(z_i) \approx a_k i + b_k$. This affine dependency of the rotation angles θ_p^k with respect to $\mathcal{Z}$ can be constructed with a controlled rotation, see Fig. 2. Higher order approximations of $\theta_p^k(z)$ can be implemented using multi-controlled rotations. Furthermore, by using quantum arithmetic one could also compute $\theta_p^k(Y)$ directly [11].

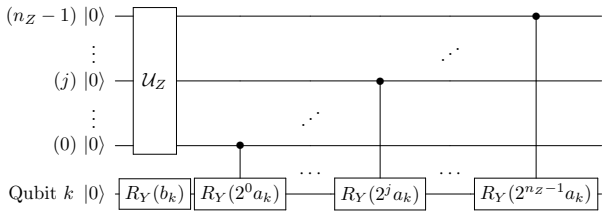

FIG. 2. Affine dependency of X_k on $\mathcal{Z}$: The qubit representing X_k is prepared using Y-rotations controlled by the qubits representing $\mathcal{Z}$. Since the rotation angles are additive this construction rotates qubit k by an angle $a_k z + b_k$.

The ability to efficiently construct the uncertainty model is a crucial part in QAE-based algorithms, and if not handled carefully can diminish the potential quantum advantage. The previous discussion shows that the Gaussian conditional independence model is particularly suitable for efficient loading in a quantum computer. However, the depth of the circuit implementing $\mathcal{U}$, shown in Fig. 2, scales as $\mathcal{O}(n_z K)$, i.e. linear in the number of assets. By adding $\mathcal{O}(K)$ ancilla qubits, the scaling of the circuit depth can be reduced to $\mathcal{O}(\log K)$, which can lead to a potential speed-up. The additional qubits provide the compute space to perform more operations in parallel. Depending on the number of available qubits and the complexity of the rest of the algorithm, the number of ancillas can also be set to a smaller value to achieve an optimal overall performance. The efficient implementation of $\mathcal{U}$ is discussed in detail in Sec. V.

Next, we need to compute the resulting total loss for every realization of the X_k. Therefore, we use a weighted sum operator

$$\begin{aligned}\mathcal{S} : &|x_1, \cdots, x_K\rangle_K\, |0\rangle_{n_S} \\ &\mapsto |x_1, \cdots, x_K\rangle_K\, |\lambda_1 x_1 + \cdots + \lambda_K x_K\rangle_{n_S}\,,\end{aligned} \tag{9}$$

where $x_k \in \{0, 1\}$ denote the possible realizations of X_k. We set $n_S = \lfloor \log_2(\lambda_1 + \cdots + \lambda_K) \rfloor + 1$ to represent in the second register all possible values of the sum of the losses given default λ_k, assumed to be integers. An efficient implementation of S is discussed in Sec. V.

Last, we need an operator that compares a particular loss realization to a given x and then flips a target qubit from $|0\rangle$ to $|1\rangle$ if the loss is less than or equal to x. This operator is defined by

$$\mathcal{C} : |i\rangle_{n_S}\, |0\rangle \mapsto \begin{cases} |i\rangle_{n_S}\, |1\rangle & \text{if } i \leq x, \\ |i\rangle_{n_S}\, |0\rangle & \text{otherwise.} \end{cases} \tag{10}$$

An efficient implementation of $\mathcal{C}$ is discussed in Sec. V.

In the remainder of this paper we apply this algorithm to a small illustrative example using classical simulations of a quantum computer and we discuss the scaling to problems of realistic size.

IV. RESULTS

In this section, we analyze the performance of the quantum algorithm for an illustrative example with $K = 2$ assets. The losses given default λ_k, the default probabilities p_k^0, and the sensitivities ρ_k are given in Tab. I. Within this section we set $n_Z = 2$, and from the λ_k it follows that $n_S = 2$. Thus, $\mathcal{A}$ is operating on seven qubits that represent this problem on a quantum computer, including the objective qubit.

TABLE I. Problem parameters for the two-assets example.

asset number k	loss given default λ_k	default prob. p_k^0	sensitivity ρ_k
1	1	0.15	0.1
2	2	0.25	0.05

To simulate our algorithm we input the circuit for $\mathcal{A}$ to the QAE sub-routine implemented in *Qiskit* [22] and

perform the bisection search using the result to find x_α. Since $n_S = 2$, the bisection search requires at most two steps, as shown in Fig. 3. Note that QAE requires one additional ancilla qubit to implement $\mathcal{Q}$ and we use four evaluation qubits giving us 16 quantum samples. In total, this experiment requires 12 qubits that we simulate using classical computers.

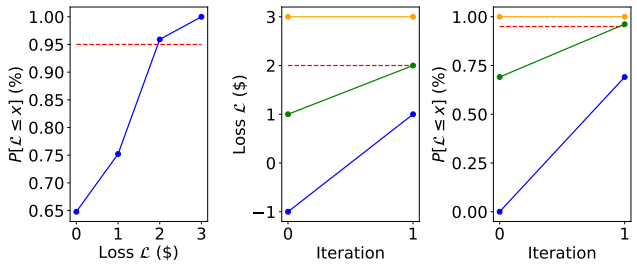

FIG. 3. Cumulative distribution function (left) of total loss $\mathcal{L}$ (blue) and target level of 95% (red). Bisection search to compute VaR (middle / right): Upper bound (orange), lower bound (blue), estimate (green), and exact value (red dashed line). Here, we set $\alpha = 95\%$ and $m = 4$.

V. SCALING TO REAL WORLD PROBLEM

We analyze the scaling of the quantum algorithm for problem sizes relevant to the finance industry. In particular, we analyze the circuit depth as a function of the number of assets K, to estimate the expected runtime on a fault-tolerant quantum computer [23–25]. We consider a gate decomposition into the Clifford + T gate set and mainly focus on the circuit depth in terms of T-gates, since they are the most expensive gates in a fault-tolerant quantum computer [26]. By using ancilla qubits, Toffoli gates can be constructed with a T-depth of one [27], thus, we treat the two as equivalent in our runtime analysis. Clifford gates, such as for instance CNOT-gates, are considered to be orders of magnitudes faster than T-gates and we mostly ignore them in the following [25, 28].

Our algorithm mainly consists of $\mathcal{A}$, multiple applications of (controlled) $\mathcal{Q}$, and an inverse quantum Fourier transform (QFT) at the end. The complexity of the inverse QFT scales at most quadratically with the number of evaluation qubits m, and is orders of magnitude smaller than the rest of the algorithm, since we assume $K \gg m$ and since the inverse QFT is only applied once. Furthermore, the inverse QFT can even be approximated using $\mathcal{O}(n \log(n))$ T-gates [29], and, as discussed later in this section, it has been recently shown that Quantum Phase Estimation (QPE, includes the inverse QFT) can be omitted completely in QAE [30]. Therefore, we ignore the contribution of the inverse QFT to the overall runtime.

Since the controlled powers of $\mathcal{Q}$ will dominate the runtime, we focus on the T/Toffoli-gates in $\mathcal{Q}$. Eq. (5) implies that the controlled-$\mathcal{Q}$ operator in QAE requires controlling only the reflections $\mathcal{S}_0$ and $\mathcal{S}_{\psi_0}$. Indeed, $\mathcal{A}$ and $\mathcal{A}^\dagger$ are left uncontrolled and cancel each other when the control qubit of $\mathcal{Q}$ is in state $|0\rangle$, since in this case $\mathcal{S}_0$ is not applied.

We now argue that $\mathcal{S}_0$ and $\mathcal{S}_{\psi_0}$ do not dominate the runtime. The reflection $\mathcal{S}_{\psi_0}$ can be implemented using an ancilla qubit and a phase kickback: an X-gate prepares the ancilla qubit in state $|1\rangle$, then the objective qubit of $\mathcal{A}$ is used to control a Z-gate targeting the ancilla qubit, a final X-gate uncomputes the ancilla qubit. This gate sequence transforms the objective qubit of $\mathcal{A}$ from $\alpha|0\rangle + \beta|1\rangle$ to $\alpha|0\rangle - \beta|1\rangle$ [7] which is equivalent to the action of $\mathcal{S}_{\psi_0}$. For a controlled application of $\mathcal{S}_{\psi_0}$ we replace the single-controlled Z-gate by a double-controlled Z-gate where the second control is an evaluation qubit. Thus, $\mathcal{S}_{\psi_0}$ can be ignored in the overall runtime analysis as it can be implemented using a single Toffoli-gate (within the double-controlled Z-gate, exploiting that $Z = HXH$).

We implement $\mathcal{S}_0$ using the same construction as for $\mathcal{S}_{\psi_0}$ but with the single-controlled Z-gate replaced by a multi-controlled Z-gate that only acts if all qubits $\mathcal{A}$ operates on are in state $|0\rangle$. However, if the sum-register is in state $|0\rangle_{n_S}$ then the K qubits representing the X_k's are also in state $|0\rangle_K$ and vice versa, since $\lambda_k > 0$ for all k. Thus, instead of controlling the Z-gate with all state qubits, we only need to control it by the n_Z qubits representing $\mathcal{Z}$, the n_S qubits representing the total loss, and the objective qubit of $\mathcal{A}$. Since multi-controlled gates can be implemented with logarithmic depth and a linear number of ancillas [31, 32], we can also ignore the contribution of (controlled) $\mathcal{S}_0$ to the total runtime.

The previous discussion in this section implies that the multiple applications of $\mathcal{A}$ dominate the total runtime. For m evaluation qubits, $\mathcal{A}$ is called $n_S(2^{m+1}-1)$ times: once for the initial state preparation, twice for each of the $2^m - 1$ applications of $\mathcal{Q}$, and everything is repeated at most n_S times for the bisection search to estimate VaR.

Since QAE is a probabilistic algorithm, we need to run it multiple times. However, 25 repetitions are already sufficient to achieve a success probability of 99.75% when using the median result [11]. These are independent repetitions that could be parallelized on multiple separate quantum computers, thus, we do not include this additional overhead.

In the following, we analyze the circuit depth of $\mathcal{A}$. How to efficiently implement the operators $\mathcal{U}$, $\mathcal{S}$, and $\mathcal{C}$ and the assumptions made, e.g., on approximation errors, is discussed in Appendices C, D, and E, respectively. The resulting circuit depths in terms of T/Toffoli-gates is stated in Table II.

The total number of qubits will scale like $\mathcal{O}(K)$, since we represent every asset with a single qubit and the required ancillas also scale linearly in K. We are mainly interested in an estimation of the overall runtime, and thus, we will not further elaborate on the exact number of required qubits.

operator	circuit depth (T/Toffoli-gates)
$\mathcal{U}$	$26 + 28n_Z$
$\mathcal{S}$	$\log_2(K)(\lfloor\log_2(n_S)\rfloor + \lfloor\log_2(n_S/3)\rfloor + 7)$
$\mathcal{C}$	$2\lfloor\log_2(n_S - 1)\rfloor + 9$

TABLE II. Bounds on circuit depth of the operators $\mathcal{U}$, $\mathcal{S}$, $\mathcal{C}$ in terms of CNOT and Toffoli-gates, see Appendices C, and D, E, for more details.

In the remainder of this section, we consider $K = 2^{20}$, i.e., a portfolio of about one million assets, and assume $n_Z = 10$, and $n_S = 30$. This implies that we discretize $\mathcal{Z}$ with $1,024$ different values, and that we assume the average of λ_k is at most $1,024 = 2^{n_S}/K$, otherwise n_S would be too small to represent the maximal possible sum of losses. Furthermore, we assume $m = 10$, which achieves an accuracy of 0.06%-points for $\alpha = 99.9\%$.

Inserting these numbers into the formulas in Table II leads to a T/Toffoli-depth for $\mathcal{A}$ of about $N_{\mathrm{T}}^{\mathcal{A}} = 600$. For the overall QAE, this implies a depth of

$$n_S(2^{m+1} - 1)N_{\mathrm{T}}^{\mathcal{A}}. \tag{11}$$

which evaluates to a T/Toffoli-depth of approximately 37 million gates.

Up to now, we have not considered the impact of the limited connectivity of quantum processors, i.e., the fact that we need to introduce SWAP-gates to realize CNOT-gates or Toffoli-gates between qubits that are not physically connected. It has been empirically shown in [11] for a related application that mapping comparable circuits to a realistic topology led to an increase in the number of CNOT-gates of about a factor of two. Since the runtime is dominated by the time for the T-gates, doubling the number of CNOT gates in our circuit should not significantly affect the overall runtime. We therefore ignore the impact of limited connectivity. Additionally, compiling the quantum circuits can be done in advance to produce a template circuit usable for a concrete problem. Thus, the actual compilation time and circuit-optimization time is not added in our analysis.

We now assume that error-corrected T/Toffoli-gates can be executed in 10^{-4} seconds [28]. With this clock rate the 37 million gates obtained from Eq. (11) result in an estimated overall runtime of around one hour. Removing the QPE from QAE not only allows to remove the inverse QFT but also reduces the overall circuit depth by a factor of two by allowing us to parallelize on two quantum devices [30]. This results in an estimated runtime of 30 minutes to estimate the VaR for a one-million-asset portfolio.

Classical simulations of large portfolios are a big computation problem which requires significant time and hardware resources [33–35]. To reduce classical simulation times, approximations are used and similar assets are aggregated in batches described by more complex random distributions. The same methods can also be applied to our quantum algorithm and should be able to achieve similar improvements, potentially reducing the expected runtime of 30 minutes for one million assets to near real-time. Furthermore, aggregating similar assets can also help to reduce the required number of qubits.

Unlike for classical algorithms, estimating the Conditional Value at Risk (CVaR, or Expected Shortfall) can be achieved without much additional overhead, since it is just one additional (slightly more expensive) application of QAE without the bisection search [11].

VI. CONCLUSION

In this paper we developed and analyzed a quantum algorithm to estimate ECR with a quadratic speedup. We have demonstrated the algorithm using a simulation and analyzed the scaling and expected runtime for realistic problem sizes under reasonable assumptions on future quantum computers. Furthermore, we argued that our results also hold for more complex uncertainty models or other objectives, such as CVaR, without much additional overhead. Although there is still a long way to go in terms of hardware development, this implies a huge potential for quantum computing in credit risk analysis. Further research in algorithms can help to reduce the number of required qubits as well as the gate depth.

Within this paper, we made assumptions on the performance of future quantum hardware. We tried to make our analysis as transparent as possible to allow adjustments of our results in case of new insights on future hardware or algorithmic components. Until quantum computers of the required scale are available, a lot of research needs to take place also with focus on quantum algorithms, error correction, and circuit optimization. Thus, it would not be surprising, if our assumption will turnout to be conservative, implying an even larger potential for the technology, than outlined in the present manuscript.

ACKNOWLEDGMENTS

The authors want to thank Joan Francesc Vidal Villalón and Santiago Murillo Pavas from CaixaBank for the inspiring discussion on this important use cases, and James Wootton as well as Dmitri Maslov for their valuable insights on quantum error-correction and gate decomposition.

Appendix A: Expected Total Loss

In the following, we show how the total loss introduced in Section II can be efficiently computed classically. For the first model in which the default events are indepen-

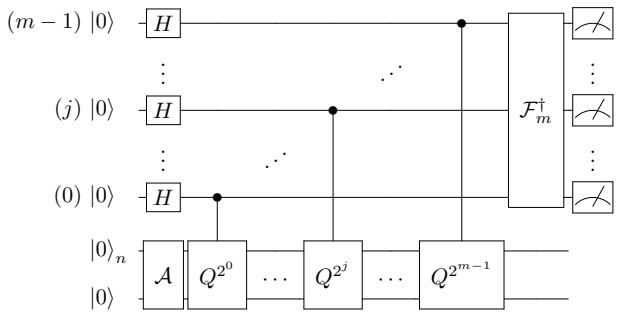

FIG. 4. The quantum circuit of amplitude estimation.

dent the expected total loss is given by

$$\mathbb{E}[\mathcal{L}] = \sum_{k=1}^{K} \lambda_k p_k. \tag{A1}$$

This is due to the linearity of the expected value and the independence of the random variables X_k.

For the second model, we exploit the conditional independence assumption, which allows us to compute the expected loss as

$$\mathbb{E}[\mathcal{L}] = \int_{z=-\infty}^{\infty} \sum_{k=1}^{K} \lambda_k p_k(z) f(z) dz, \tag{A2}$$

where f denotes the probability density function of the standard normal distribution. This term can be efficiently approximated classically using numerical integration.

Appendix B: Amplitude Estimation

The advantage for credit risk analysis comes from the quantum amplitude estimation (QAE) algorithm [20], which provides a quadratic speed-up over classical Monte-Carlo simulations [11, 15, 36, 37]. Suppose a unitary operator $\mathcal{A}$ as defined in Eq. (4). QAE allows the efficient estimation of a, i.e., the probability of measuring $|1\rangle$ in the last qubit. This estimation is obtained with an operator Q, given in Eq. (5), and Quantum Phase Estimation [38]. QAE requires m additional evaluation qubits and $M = 2^m - 1$ applications of Q. The m qubits, initialized to an equal superposition state by Hadamard gates, are used to control different powers of Q. After applying the inverse Quantum Fourier Transform, their state is measured resulting in an integer $y \in \{0, ..., M-1\}$, which is classically mapped to the estimator $\tilde{a} = \sin^2(y\pi/M) \in [0, 1]$, see the circuit in Fig. 4. The estimator $\tilde{a}$ satisfies the error bound provided in Eq. (6) with probability of at least $8/\pi^2$. This represents a quadratic speedup compared to the $\mathcal{O}\left(M^{-1/2}\right)$ convergence rate of classical Monte Carlo methods [39].

Appendix C: Uncertainty Model

Every X_k-qubit needs to be prepared using an uncontrolled Y-rotation as well as n_Z controlled Y-rotations. On an error-corrected quantum computer Y-rotations can be realized with a T-depth of about $3\log_2(1/\epsilon) - 4$ and controlled Y-rotations with a T-depth of about $3\log_2(1/\epsilon) - 2$, where $\epsilon > 0$ is the approximation error of the resulting unitary [40, 41]. Throughout this section we assume $\epsilon = 2^{-10} \approx 10^{-3}$, which implies a T-depth of 26 for uncontrolled Y-rotations and a T-depth of 28 for controlled Y-rotations.

A straight-forward implementation of $\mathcal{U}$ would require first K uncontrolled Y-rotations followed by $n_Z K$ controlled Y-rotations — from the n_Z qubits representing $\mathcal{Z}$ to all K qubits representing the X_k — with depth of K controlled Y-rotations, since we can apply n_Z rotations in parallel for $K \geq n_Z$. In this analysis we ignore the preparation of $\mathcal{U}_Z$ as it can be done efficiently [21] and does not depend on K, thus, has a negligible impact if $K \gg n_Z$.

A depth of $\mathcal{O}(K)$ is prohibitive for large portfolios due to the required T-gates. To implement $\mathcal{U}$ more efficiently, we duplicate the $\mathcal{Z}$-qubits $(w-1)$-times, i.e., in total we have w entangled copies of the n_Z qubits representing $\mathcal{Z}$. This requires $n_Z(w-1)$ ancilla qubits and $2n_Z w$ CNOT-gates, with a resulting CNOT-depth of $2\log_2(w)$, since every entangled copy can be reused to prepare more copies. The factor 2 appears since we should uncompute this preparation at the end. Having w copies of $\mathcal{Z}$ allows us to parallelize the preparation of the X_k-qubits, achieving a depth of $n_Z K/w$ controlled Y-rotations. To minimize the T-depth, we set $w = K$, i.e., we add $n_Z(K-1)$ entangled copies with a CNOT-depth of $\log_2(K)$, leading to a Y-rotation-depth of n_Z, independent of K. Combining this with the T-depth for Y-rotations leads to total T-depth for $\mathcal{U}$ of $26 + 28n_Z$.

Appendix D: Weighted Sum Operator

Next, we analyze the implementation of $\mathcal{S}$. Again, we can significantly reduce the circuit depth using additional ancilla qubits. We apply a divide and conquer approach and first sum up pairs of assets, then pairs of the resulting sums and so on until we computed the total sum.

This implies that we start with a weighted-sum operator as outlined in Sec. III and also introduced and discussed in detail in [15], and then continue with adder circuits [42] to iteratively combine the intermediate results.

For simplicity, we consider average values in the following analysis of the circuit depth. Since we have K assets and assume the total loss can always be represented with n_S qubits, the loss per asset can be represented using on average at most $\log_2(2^{n_S}/K) = n_S - \log_2(K)$ qubits. On average, the intermediate values at most double from one iteration to the next, i.e. we need to add at most one

qubit per iteration to each intermediate result and after $\log_2(K)$ iterations we have computed the total loss.

Within every iteration, we assume that the individual intermediate results can be computed in parallel. For iteration i, $i = 0, \ldots, \log_2(K)-1$, we assume the values are represented each by $n_i = (n_S - \log_2(K) + i)$ qubits. An adder circuit on n qubits can be realized with a Toffoli-depth of

$$\lfloor \log_2(n) \rfloor + \lfloor \log_2(n/3) \rfloor + 7, \tag{D1}$$

using a linear number of ancilla qubits [43]. Thus, we can bound the overall Toffoli-depth by

$$\log_2(K)(\lfloor \log_2(n_S) \rfloor + \lfloor \log_2(n_S/3) \rfloor + 7). \tag{D2}$$

Appendix E: Fixed Value Comparator

A fixed value comparator, i.e., a comparator that takes a fixed value to compare to as a classical input, can be based on adder circuits. The result reported in Table II, i.e., a Toffoli-depth of $2\log_2(n_S - 1) + 9$, is taken from [43] where a construction of adders and comparators is introduced and analyzed in detail. To achieve the logarithmic scaling, a linear number of ancilla qubits needs to be added.

[1] Bruce Porteous, Louise McCulloch, and Pradip Tapadar, "An approach to economic capital for financial services firms," Risk , 28–31 (2003).

[2] Bruce Porteous, "Managing post-convergence risks in financial conglomerates," Risk , 21–24 (2002).

[3] Risk Management Group of the Basel Committee on Banking Supervision, "Principles for the management of credit risk," (2000).

[4] Sylvain Bouteillé and Diane Coogan-Pushner, *The Handbook of Credit Risk Management* (2013) p. 322.

[5] Paul Glasserman, *Monte Carlo Methods in Financial Engineering* (Springer-Verlag New York, 2003) p. 596.

[6] Paul Glasserman and Jingyi Li, "Importance sampling for portfolio credit risk," Manag. Sci. **51**, 11 (2005).

[7] Michael A. Nielsen and Isaac L. Chuang, *Cambridge University Press* (2010) p. 702.

[8] N. Moll, P. Barkoutsos, L. S. Bishop, J. M. Chow, A. Cross, D. J. Egger, S. Filipp, A. Fuhrer, J. M. Gambetta, M. Ganzhorn, A. Kandala, A. Mezzacapo, P. Müller, W. Riess, G. Salis, J. Smolin, I. Tavernelli, and K. Temme, "Quantum optimization using variational algorithms on near-term quantum devices," Quantum Science and Technology **3**, 030503 (2018).

[9] Abhinav Kandala, Kristan Temme, Antonio D. Corcoles, Antonio Mezzacapo, Jerry M. Chow, and Jay M. Gambetta, "Error mitigation extends the computational reach of a noisy quantum processor," Nature **567**, 491–495 (2018).

[10] Vojtech Havlicek, Antonio D. Corcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow, and Jay M. Gambetta, "Supervised learning with quantum-enhanced feature spaces," Nature **567**, 209 – 212 (2019).

[11] Stefan Woerner and Daniel J. Egger, "Quantum risk analysis," npj Quantum Information **5**, 15 (2019).

[12] Patrick Rebentrost, Brajesh Gupt, and Thomas R. Bromley, "Quantum computational finance: Monte carlo pricing of financial derivatives," Phys. Rev. A **98**, 022321 (2018).

[13] Ana Martin, Bruno Candelas, Angel Rodriguez-Rozas, Jose D. Martin-Guerrero, Xi Chen, Lucas Lamata, Roman Orus, Enrique Solano, and Mikel Sanz, "Towards pricing financial derivatives with an ibm quantum computer," arXiv:1904.05803 .

[14] Roman Orus, Samuel Mugel, and Enrique Lizaso, "Quantum computing for finance: Overview and prospects," Reviews in Physics **4**, 100028 (2019).

[15] Nikitas Stamatopoulos, Daniel J. Egger, Yue Sun, Christa Zoufal, Raban Iten, Ning Shen, and Stefan Woerner, "Option Pricing using Quantum Computers," arXiv:1905.02666 .

[16] Basel Committee on Banking Supervision, "Basel III: A global regulatory framework for more resilient banks and banking systems," (2010).

[17] Marek Rutkowski and Silvio Tarca, "Regulatory capital modelling for credit risk," International Journal of Theoretical and Applied Finance **18**, 1550034 (2015).

[18] Basel Committee on Banking Supervision, "International convergence of capital measurement and capital standards," (2006).

[19] Basel Committee on Banking Supervision, "Revisions to the Basel II market risk framework," (2009).

[20] Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp, "Quantum Amplitude Amplification and Estimation," Contemporary Mathematics **305**, 53–74 (2002).

[21] Lov Grover and Terry Rudolph, "Creating superpositions that correspond to efficiently integrable probability distributions," (2002), arXiv:0208112.

[22] Gadi Aleksandrowicz, Thomas Alexander, Panagiotis Barkoutsos, Luciano Bello, Yael Ben-Haim, David Bucher, Francisco Jose Cabrera-Hernádez, Jorge Carballo-Franquis, Adrian Chen, Chun-Fu Chen, Jerry M. Chow, Antonio D. Córcoles-Gonzales, Abigail J. Cross, Andrew Cross, Juan Cruz-Benito, Chris Culver, Salvador De La Puente González, Enrique De La Torre, Delton Ding, Eugene Dumitrescu, Ivan Duran, Pieter Eendebak, Mark Everitt, Ismael Faro Sertage, Albert Frisch, Andreas Fuhrer, Jay Gambetta, Borja Godoy Gago, Juan Gomez-Mosquera, Donny Greenberg, Ikko Hamamura, Vojtech Havlicek, Joe Hellmers, Łukasz Herok, Hiroshi Horii, Shaohan Hu, Takashi Imamichi, Toshinari Itoko, Ali Javadi-Abhari, Naoki Kanazawa, Anton Karazeev, Kevin Krsulich, Peng Liu, Yang Luh, Yunho Maeng, Manoel Marques, Francisco Jose Martín-Fernández, Douglas T. McClure, David McKay, Srujan Meesala, Antonio Mezzacapo, Nikolaj Moll,

Diego Moreda Rodríguez, Giacomo Nannicini, Paul Nation, Pauline Ollitrault, Lee James O'Riordan, Hanhee Paik, Jesús Pérez, Anna Phan, Marco Pistoia, Viktor Prutyanov, Max Reuter, Julia Rice, Abdón Rodríguez Davila, Raymond Harry Putra Rudy, Mingi Ryu, Ninad Sathaye, Chris Schnabel, Eddie Schoute, Kanav Setia, Yunong Shi, Adenilton Silva, Yukio Siraichi, Seyon Sivarajah, John A. Smolin, Mathias Soeken, Hitomi Takahashi, Ivano Tavernelli, Charles Taylor, Pete Taylour, Kenso Trabing, Matthew Treinish, Wes Turner, Desiree Vogt-Lee, Christophe Vuillot, Jonathan A. Wildstrom, Jessica Wilson, Erick Winston, Christopher Wood, Stephen Wood, Stefan Wörner, Ismail Yunus Akhalwaya, and Christa Zoufal, "Qiskit: An open-source framework for quantum computing," (2019).

[23] Peter W. Shor, "Fault-tolerant quantum computation," in *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, FOCS '96 (IEEE Computer Society, Washington, DC, USA, 1996) pp. 56–.

[24] A.Yu. Kitaev, "Fault-tolerant quantum computation by anyons," Annals of Physics **303**, 2 – 30 (2003).

[25] Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland, "Surface codes: Towards practical large-scale quantum computation," Phys. Rev. A **86**, 032324 (2012).

[26] Sergey Bravyi and Jeongwan Haah, "Magic-state distillation with low overhead," Phys. Rev. A **86**, 052329 (2012).

[27] Peter Selinger, "Quantum circuits of t-depth one," Phys. Rev. A **87**, 042302 (2013).

[28] Austin G. Fowler and Craig Gidney, "Low overhead quantum computation using lattice surgery," (2018), arXiv:1808.06709.

[29] Yunseong Nam, Yuan Su, and Dmitri Maslov, "Approximate quantum fourier transform with o(nlog(n)) t gates," (2018), arXiv:1803.04933.

[30] Yohichi Suzuki, Shumpei Uno, Rudy Raymond, Tomoki Tanaka, Tamiya Onodera, and Naoki Yamamoto, "Amplitude estimation without phase estimation," (2019), arXiv:1904.10246.

[31] Dmitri Maslov, "On the advantages of using relative phase toffolis with an application to multiple control toffoli optimization," Phys. Rev. A **93**, 022311 (2016).

[32] F. Motzoi, M. P. Kaicher, and F. K. Wilhelm, "Linear and logarithmic time compositions of quantum many-body operators," Phys. Rev. Lett. **119**, 160503 (2017).

[33] Hai Lan, Barry L. Nelson, and Jeremy Staum, "A confidence interval procedure for expected shortfall risk measurement via two-level simulation," Operations Research **58**, 1481–1490 (2010).

[34] Sascha Desmettre, Ralf Korn, Javier Alejandro Varela, and Norbert Wehn, "Nested mc-based risk measurement of complex portfolios: Acceleration and energy efficiency," Risks **4**, 36 (2016).

[35] Kurt Stockinger, Jonas Heitz, Nils Andri Bundi, and Wolfgang Breymann, "Large-scale data-driven financial risk modeling using big data technology," International conference on Big Data computing, applications and technologies (2018).

[36] Daniel S Abrams and Colin P Williams, "Fast quantum algorithms for numerical integrals and stochastic processes," (1999), arXiv:9908083.

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

[38] A. Yu. Kitaev, "Quantum measurements and the Abelian Stabilizer Problem," (1995), arXiv:9511026.

[39] Paul Glasserman, Philip Heidelberger, and Perwez Shahabuddin, "Efficient Monte Carlo Methods for Value-at-Risk," in *Mastering Risk*, Vol. 2 (2000) pp. 5–18.

[40] Matthew Amy, Dmitri Maslov, Michele Mosca, and Martin Roetteler, "A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **32**, 818–830 (2013).

[41] Vadym Kliuchnikov, Dmitri Maslov, and Michele Mosca, "Practical approximation of single-qubit unitaries by single-qubit quantum Clifford and T circuits," IEEE Transactions on Computers **65**, 161–172 (2012).

[42] Steven A Cuccaro, Thomas G Draper, Samuel A Kutin, and David Petrie Moulton, "A new quantum ripple-carry addition circuit," (2004), arXiv:0410184.

[43] Thomas G. Draper, Samuel A. Kutin, Eric M. Rains, and Krysta M. Svore, "A logarithmic-depth quantum carry-lookahead adder," Quantum Information and Computation **6**, 351–369 (2006).