NISQ-HHL: Portfolio Optimization for Near-Term Quantum Hardware

Romina Yalovetzky, Pierre Minssen, Dylan Herman, Marco Pistoia
Future Lab for Applied Research and Engineering, JPMorgan Chase Bank, N.A.

***Abstract*—Portfolio optimization is an essential use case in Finance, but its computational complexity forces financial institutions to resort to approximated solutions, which are still time consuming. Thus, the scientific community is looking at how Quantum Computing can be used for efficient and accurate portfolio optimization.**

Portfolio optimization can be formulated as a quadratic program, with the cost function enforcing risk minimization for a targeted return. Of particular interest is the mean-variance portfolio optimization problem. Using the method of Lagrange multipliers, the program can be converted into a system of linear equations and potentially benefit from the exponential speedup provided by the HHL quantum algorithm. However, multiple components in HHL are unsuitable for execution on Noisy Intermediate Scale Quantum (NISQ) hardware.

This paper introduces NISQ-HHL, the first hybrid formulation of HHL suitable for the end-to-end execution of small-scale portfolio-optimization problems on NISQ devices. NISQ-HHL extends the hybrid HHL variant with newly available quantum-hardware features: mid-circuit measurement, Quantum Conditional Logic (QCL), and qubit reset and reuse. To best of our knowledge, NISQ-HHL is the first algorithm incorporating a QCL-enhanced version of Phase Estimation that was executed on real hardware. In addition, NISQ-HHL includes a novel method for choosing the optimal evolution time for the Hamiltonian simulation. Although this paper focuses on portfolio optimization, the techniques it proposes to make HHL more scalable are generally applicable to any problem that can be solved via HHL in the NISQ era.

We empirically demonstrate the effectiveness of NISQ-HHL by presenting the experimental results we obtained on a real quantum device, the trapped-ion Honeywell System Model H1.

Keywords. Portfolio Optimization, Quantum Linear Systems, HHL, Semi-Classical Quantum Fourier Transform, Mid-Circuit Measurement and Reuse, Quantum Conditional Logic.

I. Introduction

The HHL algorithm was introduced by Harrow, Hassidim and Lloyd [1] to solve the Quantum Linear Systems Problem (QLSP). This involves solving a linear system $A\vec{x} = \vec{b}$, where $A \in \mathbb{C}^{N\times N}$ and $\vec{x}, \vec{b} \in \mathbb{C}^N$, thereby returning the quantum state $|x\rangle$ corresponding, up to a normalization factor, to the solution of the linear system.

Rebentrost and Lloyd [2] applied HHL to *mean-variance portfolio optimization* [3]. This problem can be formulated as a linear quadratic-programming problem, and subsequently, be converted into a symmetric linear system, for which efficient classical algorithms exist [4].

Under the assumption that $\vec{b}$ (after normalization) can be efficiently loaded onto a quantum state and A is sparse and well-conditioned[1], HHL provides an exponential speedup in N over all known classical algorithms for QLSPs [1]. Since, for financial applications, time is of the essence, any speedup in the overall execution time can be extremely impactful. In the near term, sub-optimal algorithms might outperform asymptotically optimal ones. Therefore, we need to focus on the actual run-time complexity instead of the asymptotic one.

HHL is known to be cumbersome to deploy [7], especially on Noisy Intermediate Scale Quantum (NISQ) hardware [8]. One of the components of HHL that is particularly expensive on near-term devices is the eigenvalue inversion. Given enough qubits, this can be implemented with polynomial-depth circuits utilizing *quantum arithmetic* to approximate arcsine [9]. However, due to hidden constant factors behind the asymptotic notation, most near-term approaches fall back on an implementation based on the *uniformly controlled rotation* gate, which leads to n-qubit circuits that have $\Omega(2^n)$ circuit depth's [10].

This paper introduces NISQ-HHL, a novel enhancement of the classical/quantum hybrid HHL variant introduced by Lee *et al.* [11]. By leveraging newly available hardware features and optimizing existing components of HHL, our novel approach is suitable for the end-to-end execution of small-size mean-variance portfolio-optimization problems on NISQ devices. Although this paper focuses on portfolio optimization, the techniques it proposes to make HHL more scalable are generally applicable to any problem that can be solved via HHL in the NISQ era.

A. Novel Contributions of This Work

This paper makes the following novel contributions, all integrated into NISQ-HHL:

1) An enhanced formulation of the classical/quantum hybrid HHL algorithm [11] that integrates mid-circuit measurement, Quantum Conditional Logic (QCL), and qubit reset and reuse [12, 13] into the separate Quantum Phase Estimation (QPE) routine used for eigenvalue estimation. This has two main advantages, which make the resulting algorithm more suitable for NISQ computers:
   a) The number of ancillary qubits is reduced to just one. This single ancilla is measured, reset and reused as frequently as needed, thereby allowing for significantly fewer qubits for the computation.

[1] Methods for handling ill-conditioned matrices have been proposed [5, 6].

   b) While the standard QPE requires controlled gates between the various ancillary qubits, QCL enables applying gates conditioned on classical registers. This reduces the requirement for qubit connectivity, SWAP gates or qubit transport.

2) A new, efficient procedure for determining a value to scale A that allows for resolving the eigenvalues with significantly higher accuracy.
3) An empirical evaluation obtained by executing NISQ-HHL to solve portfolio-optimization problems end-to-end on real quantum hardware—the trapped-ion Honeywell System Model H1. The benchmark consists of a portfolio of S&P 500 assets. The evaluation includes a comparative analysis of the results.

B. Paper Organization

The remainder of this paper is organized as follows. Section II describes a known formulation of portfolio optimization as a QLSP, which makes it compatible with HHL. Section III introduces NISQ-HHL, focusing on its novelties. Section IV presents a detailed comparative analysis of the experimental results obtained by executing NISQ-HHL on the trapped-ion Honeywell System Model H1. Section V analyzes the circuits generated by NISQ-HHL and the results obtained on the Qiskit [14, 15] statevector simulator, indicating that NISQ-HHL can scale to portfolios potentially larger than those supported on current hardware. Section VI compares NISQ-HHL to state-of-the-art in quantum algorithms for portfolio optimization. Finally, Section VII summarizes the results of this work, and concludes the article.

II. Portfolio Optimization as a QLSP

HHL has been proposed as a possible solver for a specific portfolio-management problem [2], known as mean-variance portfolio optimization. Given a set of $\tilde{N}$ assets, this problem requires the following quantities as inputs: the *historical covariance matrix* $\Sigma \in \mathbb{R}^{\tilde{N}\times\tilde{N}}$, the *expected returns* $\vec{r} \in \mathbb{R}^{\tilde{N}}$, and the *prices* $\vec{p} \in \mathbb{R}^{\tilde{N}}$ of the assets. Its objective is to minimize the *risk*, represented by the quadratic form $\vec{w}^{\mathsf{T}}\Sigma\vec{w}$, subject to the desired *expected total return* $\mu \in \mathbb{R}$ and *budget* $\xi \in \mathbb{R}$. The solution $\vec{w} \in \mathbb{R}^{\tilde{N}}$ is the *allocation vector* that weighs each asset in the portfolio.[2]

This problem can be stated as a convex quadratic program:

$$\underset{\vec{w}\in\mathbb{R}^{\tilde{N}}}{\text{minimize}}\ \vec{w}^{\mathsf{T}}\Sigma\vec{w}\ :\ \xi = \vec{p}^{\mathsf{T}}\vec{w},\ \mu = \vec{r}^{\mathsf{T}}\vec{w}$$

This quadratic program can be reformulated as a linear system by using the method of Lagrange multipliers, resulting in the following equation:

$$\begin{bmatrix} 0 & 0 & \vec{r}^{\mathsf{T}} \\ 0 & 0 & \vec{p}^{\mathsf{T}} \\ \vec{r} & \vec{p} & \Sigma \end{bmatrix} \begin{bmatrix} \eta \\ \theta \\ \vec{w} \end{bmatrix} = \begin{bmatrix} \mu \\ \xi \\ \vec{0} \end{bmatrix} \tag{1}$$

where $\eta, \theta \in \mathbb{R}$ are the Lagrange multipliers. We will denote this linear system by $A\vec{x} = \vec{b}$, with $A \in \mathbb{R}^{N\times N}$ and $\vec{x}, \vec{b} \in \mathbb{R}^{N}$, where $N = \tilde{N} + 2$. A quantum state representing the solution, up to a normalization constant, can be obtained by solving the corresponding QLSP using HHL. This can be done because the covariance matrix, Σ, is Hermitian, and so A is Hermitian too. The resulting quantum state $|x\rangle = |\eta, \theta, w\rangle$ allows us to recover $|w\rangle$.

Following this approach, Rebentrost and Lloyd [2] have shown how to use the quantum state produced by HHL to make calculations that are of interest to the financial industry. For instance, given the optimal portfolio state, one can measure the portfolio's risk, or compare, through a controlled-SWAP test [16], the optimal portfolio to another candidate portfolio (e.g., one offered by a third party) that has been loaded onto a quantum state. The result of this comparison can then be used to decide which portfolio to invest in.

Formulating a mean-variance portfolio optimization problem as a QLSP may allow us to harness the theoretically proven exponential speedup provided by HHL. This is in contrast to general-purpose variational quantum algorithms [17–19] and, more specifically, variational quantum linear solvers [20, 21], which utilize heuristics and often have unproven speedups. However, there are important conditions that need to be met in order for HHL to have an exponential speedup.

For example, from Equation (1), it can be seen that $\vec{b}$ has only two non-zero entries, corresponding to the two constraints μ and ξ. Therefore, the number of non-zero entries in $\vec{b}$ is independent of N. As a consequence, $\vec{b}$ can be efficiently loaded onto a quantum state $|b\rangle$ with a single $\mathsf{R_y}$ gate. Conversely, A has only four guaranteed zero entries. The potential lack of sparsity of A may negate HHL's exponential speedup when performing the Hamiltonian simulation required by QPE [7].

Another challenge is how to access the allocation weights in the quantum solution state. One approach is to use quantum-state tomography. The complexity of this procedure for a real-valued state vector is $O(\tau N\log(N)/\epsilon^2)$ where ϵ is the accuracy and τ is the time to produce the HHL solution [22]. Therefore, the use of this technique would eliminate the possibility of an exponential speedup. Another approach is quantum-state sampling [2], because a good approximation of the allocation vector could be obtained by sampling from it. The signs of all the entries can be approximately recovered via what Rebentrost and Lloyd call the *long/short assumption*. This approach potentially allows us to maintain the same exponentially-reduced dependence on N enabled by HHL.

In Section III, we present the full end-to-end flow of NISQ-

[2] Since the solution is a weight vector, the budget is only a scaling parameter and can be set to 1. Moreover, in order to compare portfolio performances, the return of a portfolio is usually expressed as a percentage instead of a monetary amount. With a simple change of variables, the problem can be reformulated in these terms.

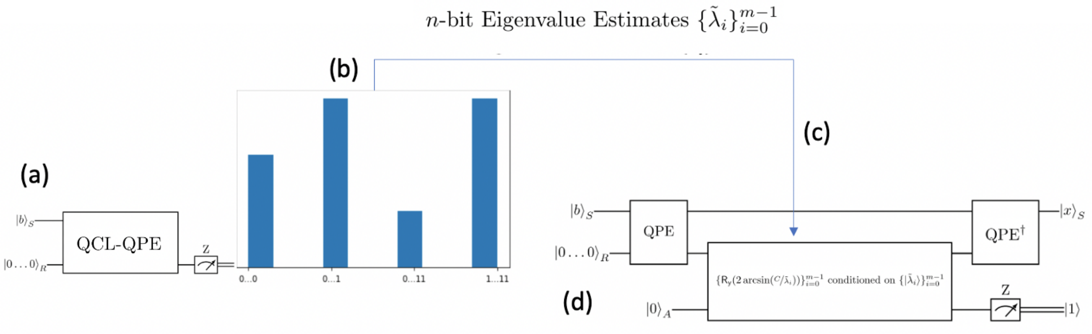

n-bit Eigenvalue Estimates $\{\tilde{\lambda}_i\}_{i=0}^{m-1}$
(b)
(c)
(a)
$|b\rangle_S$
$|0\dots0\rangle_R$
QCL-QPE
Z
0...0
0...1
0...11
1...11
$|b\rangle_S$
$|0\dots0\rangle_R$
QPE
QPE†
$|x\rangle_S$
$\{\mathsf{R_y}(2\arcsin(C/\tilde{\lambda}_i))\}_{i=0}^{m-1}$ conditioned on $\{|\tilde{\lambda}_i\rangle\}_{i=0}^{m-1}$
(d)
$|0\rangle_A$
Z
$|1\rangle$

Fig. 1. NISQ-HHL End-to-End Flow

HHL, illustrating how its characteristics make it suitable for executiong in NISQ devices.

III. NISQ-HHL

This section presents NISQ-HHL and all of its components, emphasizing the features that makes it scale on NISQ devices. Section IV shows an evaluation of NISQ-HHL on real hardware for a portfolio-optimization problem.

Given a Hermitian matrix $A \in \mathbb{C}^{N\times N}$, the n-bit estimation of an eigenvalue λ_i of A, with $i \in \{0,1,\dots,N-1\}$, can be represented by a binary string of length n, of the form $\tilde{\lambda}_i := \lambda_i^{(n-1)}\dots\lambda_i^{(1)}\lambda_i^{(0)}$. Such a binary string can be encoded in a computational basis state as follows: $|\tilde{\lambda}_i\rangle_T := |\lambda_i^{(n-1)}\dots\lambda_i^{(1)}\lambda_i^{(0)}\rangle_T$, where T is an n-qubit register.

Since A is Hermitian, $|b\rangle_S$ can be decomposed as $|b\rangle_S = \sum_{i=0}^{N-1}\beta_i\,|u_i\rangle_S$, where $\{\vec{u}_i\}_{i=0}^{N-1}$ is an orthonormal basis of $\mathbb{C}^N$ consisting of only eigenvectors of A, and S is a quantum register. Given that we are interested in problems that have unique solutions, from now on we will assume A is invertible.

The solution to the QLSP in Equation (1), ignoring a normalization constant, can be expressed as follows:

$$|x\rangle_S = \left|A^{-1}b\right\rangle_S = \sum_{i=0}^{N-1}\frac{\beta_i}{\lambda_i}\,|u_i\rangle_S$$

We define the set of the *relevant eigenvalues* of A as $\Lambda_b := \{\lambda_i : |{}^{\beta_i}/{}_{\lambda_i}| > \epsilon\}$, where $\epsilon \geq 0$ is a configurable threshold. Essentially, Λ_b contains the m distinct eigenvalues of A whose amplitudes in the solution, in absolute value, are sufficiently large.

One of the contributions of NISQ-HHL is its ability to enhance the hybrid HHL algorithm introduced by Lee *et al.* by replacing the standard QPE with a version of QPE that uses Quantum Conditional Logic (QCL) for eigenvalue estimation. These estimates are employed to condition the rotations in the eigenvalue inversion component. We refer to this QCL-enhanced QPE as QCL-QPE. In NISQ-HHL, QCL-QPE is used to estimate the relevant eigenvalues. In order to do so, this requires evolving the propagator $U := e^{iA2\pi\gamma}$ of A, for time $2\pi\gamma$, via Hamiltonian simulation. We define γ as the *scaling parameter* of the matrix A, for reasons that will be made clear in Section III-B. Another contribution of NISQ-HHL is integrating a novel verifiable procedure for selecting the optimal scaling parameter.

Figure 1 illustrates the end-to-end flow of NISQ-HHL. Our aim is to execute and validate NISQ-HHL on real quantum hardware supporting mid-circuit measurement, qubit reset and reuse, and QCL. NISQ-HHL consists of four steps, indicated as (a), (b), (c) and (d) in Figure 1:

(a) The QCL-QPE procedure is used to construct a distribution over the estimates of the relevant eigenvalues with n-bit precision. For this to work efficiently, we need to scale A by the optimal parameter γ. To do so, we first run Algorithms 1 and 2, introduced in Section III-B, to select the optimal value of γ.
(b) Classical post-processing is performed on the resulting histogram to obtain the estimates of the m relevant eigenvalues, i.e., Λ_b.
(c) The n-bit estimates, $\{\tilde{\lambda}_i\}_{i=0}^{m-1}$, obtained in (b), are used to determine rotation angles $\{2\arcsin({}^{C}/{}_{\tilde{\lambda}_i})\}_{i=0}^{m-1}$ for the eigenvalue inversion circuit. The estimates are also mapped to a smaller number of bits, r. Each rotation is conditioned on its corresponding r-bit estimate.
(d) The standard HHL procedure is executed, but it uses the circuit constructed in (c) for the eigenvalue inversion step.

Note that the implementation of the Hamiltonian simulation routine is not in the scope of this work. It would require addressing the challenges of performing it on NISQ devices [23]. In order to perform experiments with NISQ-HHL on quantum hardware, instead of using quantum algorithms for Hamiltonian simulation, we classically calculate U. Then, we pass it to the Qiskit transpiler [14], which decomposes it into basis gates.

In the following subsections we illustrate NISQ-HHL in detail.

A. Eigenvalue Estimation and Inversion with Quantum Conditional Logic

The eigenvalue inversion component of HHL involves controlled rotations conditioned on the n-bit estimations of the eigenvalues, i.e., multiset $\{\tilde{\lambda}_i\}_{i=0}^{N-1}$. This algorithmic component can thus be represented by the following mapping:

$$\sum_{i=0}^{N-1} |0\rangle \otimes \beta_i |u_i\rangle_S \otimes |\tilde{\lambda}_i\rangle_T \to$$
$$\sum_{i=0}^{N-1} \left(\sqrt{1 - \frac{C^2}{\tilde{\lambda}_i^2}} |0\rangle + \frac{C}{\tilde{\lambda}_i} |1\rangle \right) \otimes \beta_i |u_i\rangle_S \otimes |\tilde{\lambda}_i\rangle_T \quad (2)$$

where C is a normalization constant chosen to be in $O(1/\kappa)$, and κ is the condition number of A [1].

This transformation is accomplished by applying $\mathsf{R_y}$ rotation gates to the ancillary qubit $|0\rangle$ controlled by the T register containing the eigenvalue estimates. A rotation will have to be applied for each of the distinct elements of $\{\tilde{\lambda}_i\}_{i=0}^{N-1}$. The angle for the i^{th} rotation is $\theta_i = 2\arcsin(C/\tilde{\lambda}_i)$. This computation requires either some prior information about the eigenvalues or a coherent computation of the arcsine function using quantum arithmetic [24]. However, quantum arithmetic has not yet been shown to be feasible on NISQ devices. The asymptotically efficient implementation of arcsine made by Häner *et al.* [24] requires over one-thousand CNOT gates even for a small register T of size two.

Another approach would be to perform an exhaustive search of the basis states of the eigenvalue estimation register. This can be accomplished by a uniformly controlled rotation gate [10]. A uniformly controlled rotation on n qubits decomposes into $2^n - 1$ n-qubit controlled rotations, as shown in Figure 2.

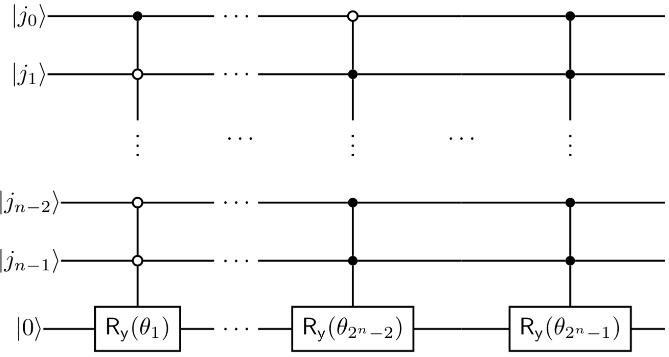


Fig. 2. Decomposition of a uniformly controlled rotation gate into $2^n - 1$ n-qubit controlled rotations. QPE loads the k^{th} bit of the n-bit approximation of each eigenvalue onto qubit j_k, for $k = 0, \ldots, n-2, n-1$. For eigenvalue inversion, $\theta_i = 2\arcsin(2^n C/i)$, for $i = 1, \ldots, 2^n - 2, 2^n - 1$.

This would effectively control on all of the eigenvalue approximations that can be made with n bits, which exclude the zero state because A is assumed to be invertible. However, this approach can quickly become infeasible since n is the bit precision required to approximate the eigenvalues, and the circuit depth's dependence on it is exponential.[3]

Yet another approach is the classical/quantum hybrid solution for HHL introduced by Lee *et al.*. Before running HHL, they apply QPE to the propagator U of A with input state $|b\rangle$. The output probability distribution is used to obtain estimates of the eigenvalues before performing the eigenvalue inversion. These estimates are employed to compute the angles and controls for the required rotations to perform the eigenvalue inversion.

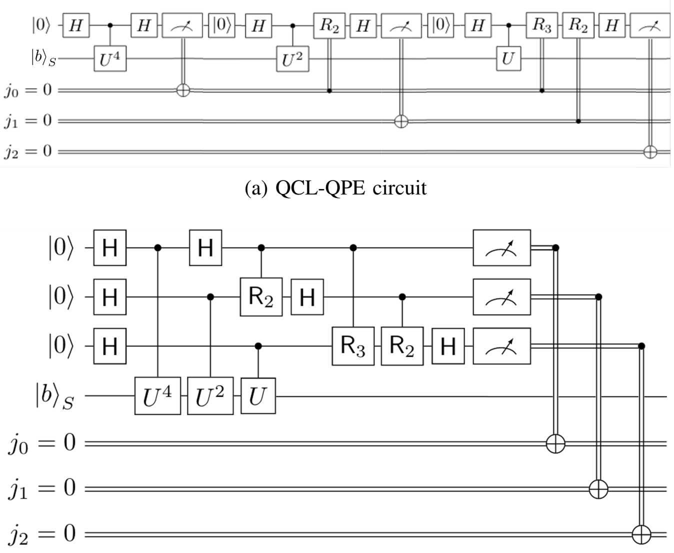
NISQ-HHL aims at estimating only the m relevant eigenvalues of A. The reason is that their amplitudes in the solution of the QLSP, in absolute value, are sufficiently large. Moreover, by controlling the rotations on the estimates of the relevant eigenvalues, we are reducing the number of controlled rotations in comparison to the uniformly controlled rotation gate. In the general case, $m \leq N$, where $m := |\Lambda_b|$ and N is the size of the linear system, as introduced previously. In the uniformly controlled rotation approach, the rotations are conditioned on each of the possible n-bit strings. Since the elements of Λ_b are distinct and, in the worst case, $m = N$, it follows that $m \leq N \leq 2^n - 1$. For example, given a 5×5 Hermitian matrix A, three bits are required to represent all the possible elements in Λ_b. This implies that there are seven rotations in the uniformly controlled rotation gate. Since n is the number of bits used for estimating the eigenvalues, potentially requiring high precision, it is usually the case that $m \leq N \ll 2^n - 1$. Furthermore, as we increase this precision, the number of rotations in the uniformly controlled rotation gate increases exponentially. This enables the NISQ-HHL eigenvalue inversion implementation to use significantly fewer controlled rotations than the uniformly controlled rotation approach.

To estimate the relevant eigenvalues, one could utilize the aforementioned QPE approach of Lee *et al.*. However, QPE is still difficult to implement on NISQ hardware because the number of ancillary qubits grows as the desired bit precision, and QPE relies on many controlled gates [26]. Moreover, QPE fully entangles qubits. Therefore, an all-to-all connection topology is preferred to limit the use of SWAP gates.

At least at a theoretical level, a QPE variant has been identified that better lends itself to NISQ hardware by utilizing the semi-classical inverse Quantum Fourier Transform (QFT) [27–29]. This is a non-unitary version of the inverse QFT that estimates each bit of the eigenvalue sequentially. A diagram of this variant is displayed in Figure 3a, which shows how to efficiently estimate the eigenvalues of the unitary operator U to three-bit precision by leveraging mid-circuit measurements, ground-state resets, QCL, and qubit reuse.

QCL-QPE is mathematically equivalent to performing the original inverse QFT and measuring the eigenvalue register. It is also similar to iterative Quantum Phase Estimation (iQPE) [30] to the extent that it only requires one ancillary qubit to achieve arbitrary bit precision of the eigenvalues. A limitation of iQPE, however, is that it requires the initial state to be an eigenvector of U in order to estimate its corresponding eigenvalue. This prior knowledge of the eigenvectors could be used to directly solve the QLSP. Conversely, by leveraging the mid-circuit measurement, qubit reset and reuse, and QCL hardware technology, QCL-QPE can estimate eigenvalues without prior information of the eigenvectors.

[3] The *Gray code* [25] can be used to reduce the number of basis gates required for this exponentially long sequence of n-qubit controlled rotations. However, it comes with the cost of computationally intensive classical operations, and the depth of the circuit is still in $\Theta(2^n)$.



(a) QCL-QPE circuit

(b) QPE circuit

Fig. 3. Circuits for estimating the eigenvalues of the unitary operator U to three bits using QCL-QPE (a) or QPE (b). S is the register that U is applied to, and j is a classical register. H refers to the Hadamard gate and $\mathsf{R_k}$, for $\mathsf{k} = 2, 3$, are the phase gates.

QCL-QPE has two properties that make its circuits more suitable for near-term devices than the standard version of QPE:

1) QCL-QPE requires only one ancillary qubit for an arbitrary bit precision, and
2) QCL-QPE replaces two-qubit gates with one-qubit gates controlled by classical bits as it can be seen by comparing Figures 3a and 3b.

The relevant eigenvalues of A are estimated in a separate procedure using QCL-QPE, represented as Step (a) in Figure 1. Therefore, in Step (c), we can classically map the n-bit eigenvalue estimates to r bits, such that $r < n$. The rotation gates are conditioned on these r-bit estimates, while the rotation angles are computed using the higher-precision, n-bit estimates. Following this, in Step (d), we are able to run the standard HHL using r ancillas, while performing matrix inversion with n-bit estimations of the eigenvalues. Lowering the number of qubits and circuit depth reduces the exposure to noise.

Note that, apart from the eigenvalue estimation component, where the standard QPE is replaced by QCL-QPE, as shown in Section III-A, NISQ-HHL uses the standard QPE. This is because QCL-QPE repeatedly collapses the quantum state of the ancillary qubit through mid-circuit measurements. Therefore, QCL-QPE cannot be incorporated as part of a deeper circuit that relies on the quantum states encoded in this ancillary register.

It is important to emphasize that we do not claim asymptotic efficiency. However, we do assert that the total complexity of NISQ-HHL is lower than the ones achieved with the previously mentioned methods (quantum arithmetic and uniformly controlled rotation gate) for a low-qubit count. Therefore, the presented methodology brings HHL closer to being realizable on current quantum hardware.

Having explained QCL-QPE and how it is used to construct a distribution over the estimates of the relevant eigenvalues, in Section III-B, we introduce the algorithms to find the optimal scaling parameter γ. This parameter is used to scale the matrix A, which enables resolving the eigenvalues in the QCL-QPE output distribution with significantly higher accuracy.

B. Optimizing the Selection of the Scaling Parameter

In the foundational HHL article [1], A is assumed to have positive eigenvalues in $[{}^1/_\kappa, 1]$, where κ is the condition number of A. The eigenvalues are restricted to this range to account for the periodicity of the imaginary exponential and ensured well-conditioning. In practice, it is necessary to scale A to have a spectrum in this range. However, even under this assumption, we could be wasting qubits to unnecessarily encode values between the largest eigenvalue, λ_{max}, and 1. We pursue a more efficient approach, consisting of estimating λ_{max} first, and then scaling A by $\gamma = \tilde{\lambda}_{\text{max}}^{-1}$, so that the maximum eigenvalue of γA is 1.

A benefit of our approach is that it only considers $\lambda_{\text{max},b}$, the largest eigenvalue in Λ_b, instead of λ_{max}. For now on, we will only take into account the eigenvalues in Λ_b and will not, for example, make a distinction between $\lambda_{\text{max},b}$ and λ_{max}.

Algorithm 1: Optimize the selection of γ using n-bit estimations of eigenvalues

```
Guess an overapproximation $\alpha$ of $\lambda_{\text{max}}$
$\gamma := {}^1/_\alpha$ // Initialize scaling parameter
$x := 0$
// At each step, $\gamma * \lambda_{\text{max}} \leq 1$
while $x \neq 2^n - 1$ do
    $p :=$ $n$-bit output distribution of QCL-QPE using
      unitary $e^{iA2\pi\gamma}$ and input state $|b\rangle$
    $x := \max\{j \in \{0, ..., 2^n - 1\} |\ p_j > 0,\ p_j \in p\}$
    // $x$ is an $n$-bit estimation of
    // $2^n * \gamma * \lambda_{\text{max}}$
    if $x = 0$ then
        $\gamma := \gamma * 2^n$
    else
        $\gamma := \gamma * {}^{(2^n - 1)}/_x$
    end
end
Result: $\gamma = \tilde{\lambda}_{\text{max}}^{-1}$, with $\tilde{\lambda}_{\text{max}}$ $n$-bit estimation of $\lambda_{\text{max}}$
```

Algorithm 1 shows how to optimize the selection of γ. Given n-bits, the optimal value of γ returned by the algorithm helps to encode the eigenvalues using all available bits in the output distribution of QCL-QPE applied to $U := e^{iA2\pi\gamma}$. As a result, this makes it easier to distinguish the eigenvalues from each other and estimate the relevant eigenvalues, i.e., Λ_b, accurately. Without loss of generality, Algorithm 1 assumes all the eigenvalues to be positive. We will show shortly how to account for negative eigenvalues.

Algorithm 1 starts by guessing an overapproximation of λ_{max}. We will discuss later how to make this operation rigorous. The next step consists of iteratively updating the value of γ until γ converges to the optimal value. During each iteration, Algorithm 1 runs QCL-QPE to compute the n-bit estimates of the eigenvalues of the unitary U using gamma computed during the previous iteration. Specifically, Algorithm 1 post-processes the output distribution to get a new n-bit estimation of $2^n\gamma\lambda_{\text{max}}$ in order to update γ for the next iteration.[4]

The number of iterations to find the optimal γ using Algorithm 1 is in $\Theta\left({}^1\!/\!{}_n \log_2\left({}^\alpha\!/\!{}_{\lambda_{\text{max}}}\right)\right)$. To ensure we do not overestimate γ in the process, we could take a conservative approach, which consists of overestimating x while computing it, thus lowering ${}^{(2^n-1)}\!/\!{}_x$. In practice, we also want to underestimate γ in order to prevent amplitudes of basis states near 2^n from dispersing and mixing with basis states near 0 and causing overflowing.

Regarding the bit precision, to represent both the largest and smallest eigenvalues, in theory, we need $n \geq \log_2({}^{\lambda_{\text{max}}}\!/\!{}_{\lambda_{\text{min}}}) = \log_2(\kappa)$. However, this requires prior knowledge of κ. In practice, we first run Algorithm 1 with an initial precision of n bits. Then, if λ_{min}, estimated via QLC-QPE, turns out to be 0, this indicates that we need higher bit precision to prevent the 0 state from having significant probability. This is because A is assumed invertible and consequently, no eigenvalue can be 0. Thus, we increase n just enough to guarantee that the estimation of λ_{min} is different from 0, at which point $n \geq \log_2(\kappa)$.

The correctness of Algorithm 1 relies on the fact that α is an overapproximation of λ_{max}. One way to achieve this result is to use the Frobenius norm of A, defined as $||A||_{\text{F}} := \sqrt{\text{Tr}(A^\dagger A)}$. Indeed, since A is Hermitian, $||A||_{\text{F}} = \sqrt{\sum_{i=0}^{N-1} \lambda_i^2}$, and so the Frobenius norm of A is a valid overapproximation of λ_{max}. However, the computation of the Frobenius norm has quadratic complexity in N. Therefore, it is desirable to find a faster approach for finding a value for α that overapproximates λ_{max}. For this reason, we propose Algorithm 2, which, using one execution of QCL-QPE, tests the validity of any initial guess of α. This bypasses the need to compute $||A||_{\text{F}}$. If the initial guess of α does not satisfy $\alpha \geq \lambda_{\text{max}}$, we can retry with a, potentially significantly, larger α. In fact, given the logarithmic complexity, in terms of number of iterations, of Algorithm 1, even with low bit precision, such as $n = 4$, overestimating λ_{max} by a factor of a billion would only require eight iterations before returning the optimal γ. In practice, we expect to find γ significantly more efficiently by starting with a guess, α, than by computing $||A||_{\text{F}}$.

The idea of Algorithm 2 is the following. If α is a valid guess, there will be no overflow in the output distribution of QCL-QPE applied to $e^{iA2\pi\gamma}$ with $\gamma = {}^1\!/\!{}_\alpha$. Then, performing $n+1$ right bit shifts should reduce all n-bit eigenvalue estimates to 0. To test this, Algorithm 2 executes QCL-QPE using $e^{iA2\pi\Gamma}$ with $\Gamma = {}^1\!/\!{}_{2^{n+1}\alpha}$. On the contrary, if α is not a valid guess, the $n+1$ right bit shifts would not be sufficient to reduce all estimations to 0. Note, γ and Γ have the same role in the definition of U and just help differentiate the two algorithms.

Algorithm 2: Verify if α is a n-bit overestimation of λ_{max}

Assumption: at each step, there is at least one eigenvalue of γA not in $\bigcup_{j\in\mathbb{Z}} \left[j - 2^{-(n+1)},\ j + 2^{-(n+1)}\right]$

```
// Initialize scaling parameter
$\Gamma := {}^1\!/\!{}_{2^{n+1}\alpha}$
$p :=$ $n$-bit output distribution of QCL-QPE using
  unitary $e^{iA2\pi\Gamma}$ and input state $|b\rangle$
if $p_0 \neq 1$ then
    Return $\alpha$ is not valid
    // Otherwise all eigenvalues
    // estimations would have been 0
else
    Return $\alpha$ is valid
end
Result: Return if $\alpha$ is an overestimation of $\lambda_{\text{max}}$
```

As mentioned before, Algorithm 1 assumes positive eigenvalues. To take into account negative eigenvalues, one can define the maximum eigenvalue using the absolute value, encode negative eigenvalues using two's complement and then replace 2^n by 2^{n-1} in the update of γ. Similarly, in Algorithm 2, replacing ${}^1\!/\!{}_{2^{n+1}}$ by ${}^1\!/\!{}_{2^n}$ in the definition of Γ suffices to support negative eigenvalues. Accounting for negative eigenvalues is crucial, since A, in the QLSP represented by Equation (1), may be indefinite.

We note that Kerenidis and Prakash [31] have developed an algorithm to ϵ-approximate $\eta := {}^{||A||_2}\!/\!{}_{||A||_{\text{F}}}$, where $||A||_2 = |\lambda_{\text{max}}|$ is the spectral norm of A. Thus, their algorithm can also be used to find $|\lambda_{\text{max}}|$ given that $||A||_{\text{F}}$ has been previously computed, which is not required for Algorithm 1. As mentioned before, the optimal γ is $|\lambda_{\text{max},b}|^{-1}$ and not $|\lambda_{\text{max}}|^{-1}$, when given $|b\rangle$ as an initial state. Since Algorithm 1 only computes $|\lambda_{\text{max},b}|^{-1}$, this makes it more suitable for NISQ-HHLbased on the discussions in Section III-A.

In addition, their algorithm executes standard QPE, which is not exchangeable for QCL-QPE because of the coherence requirement. In contrast, Algorithm 1 runs QCL-QPE, which for the reasons explained above, is more suitable for NISQ computers.

Having introduced the theoretical formulation of NISQ-HHL, in Section IV we present experimental results obtained from running the components of NISQ-HHL separately, as well as collectively in an end-to-end flow.

IV. Experimental Results

We considered a portfolio-optimization problem with two S&P 500 assets for demonstrating the validity of NISQ-HHL. After considering the two constraints, the matrix A of the

[4] Any algorithm outputting an n-bit estimation of $2^n\gamma\lambda_{\text{max}}$ could be used in place of QCL-QPE. Nevertheless, we chose to use QCL-QPE, since it is more compatible with NISQ devices, as explained in Section III-A.

linear system is of size 4×4. The experimental results presented in this section were obtained on the trapped-ion Honeywell System Model H1 given its support for mid-circuit measurements, qubit resets and reuses, and QCL [12]. In order to do so, we transpiled and optimized the circuits from Qiskit to H1's native gates using Cambridge Quantum's pytket package [32].

Note that, for the reasons explained in Section III, in the components where Hamiltonian simulation is required, we classically computed $U := e^{iA2\pi\gamma}$, where γ is the scaling parameter. Then, we passed it to Qiskit, which decomposed it into basis gates.

As the error rates of two-qubit gates are an order of magnitude larger than those of one-qubit gates [12], and the numbers of both gate types are similar in the circuits used, we will only present the H1 two-qubit gate ZZMax counts for the circuits. The ZZMax is equivalent to $\mathsf{R}_{\mathsf{ZZ}}(\pi/2)$, and, up to one-qubit gates, it is realized via the Mølmer-Sørensen interaction [12].

In the following subsections, we present data collected from running some of the components of NISQ-HHL separately. We used these results to calibrate the precision to use at each step of an end-to-end run of NISQ-HHL for this particular portfolio-optimization problem. Then, in Section IV-D, we show results from the end-to-end execution. The circuits were also ran on the Qiskit Aer statevector and QASM simulators to compare with the hardware runs.

A. Standard QPE and QCL-QPE Benchmark

We benchmarked the performance of both the standard QPE and QCL-QPE for estimating the eigenvalues of A. We used as the initial state $|b\rangle$ from the portfolio-optimization problem with two assets that was introduced before. We set $\gamma = 100$ for both implementations.

In Table I, we compare the number of gates and qubits required for both QPE implementations for estimating the eigenvalues of A to different precisions: three, four and five. We can see that QCL-QPE employs fewer qubits and gates.

		3-bit	4-bit	5-bit
Standard QPE	Gates	63	88	115
	Qubits	5	6	7
QCL-QPE	Gates	57	76	95
	Qubits	3	3	3

TABLE I
COMPARISON OF THE NUMBER OF TWO-QUBIT ZZMAX GATES AND QUBITS IN BOTH QPE IMPLEMENTATIONS FOR ESTIMATING EIGENVALUES TO DIFFERENT PRECISIONS.

As the precision grows, the number of two-qubit gates increases for both implementations. However, the number of two-qubit gates saved using QCL-QPE, instead of the standard implementation, grows quadratically as $n(n-1)$, with n being the bit precision. Moreover, even though the precision in bits increases, the number of qubits in the QCL implementation does not change. This contrasts with the linear growth in the standard QPE.

In order to quantify the performance of both implementations, we compared the empirical distribution of measurement results from the circuit execution on the Honeywell System Model H1 to the distribution obtained from the Qiskit QASM simulator. One way to compare two probability mass functions, p and q, is to use the fidelity metric [33]: $F(p,q) = (\sum_i \sqrt{p_i q_i})^2, F(p,q) \in [0,1]$.

We compare the achieved fidelity in both implementations for the three precisions, in Table II. It can be seen that the computed fidelity metrics for the two implementations are similar, for three-bit estimations. Here the number of saved two-qubit gates using QCL-QPE is small. In addition, reducing the number of qubits does not overcome potential errors due to mid-circuit measurements and resets.

	3-bit	4-bit	5-bit
Standard QPE	98.6	90.4	42.6
QCL-QPE	98.1	95.0	43.2

TABLE II
FIDELITY EXPRESSED IN % BETWEEN THE PROBABILITY DISTRIBUTIONS FROM THE QPE EXPERIMENTS RAN ON THE HONEYWELL SYSTEM MODEL H1 AND IN THE QISKIT QASM SIMULATION, WITH 2000 SHOTS EACH.

When we increase the precision to four and five bits, the circuits in both implementations deepen, and therefore, we see a drop in fidelity. Nevertheless, as shown in Table I, QCL, mid-circuit measurement, and qubit reset and reuse, result in the QCL-QPE circuit being shallower than the standard implementation. As a consequence, the achieved fidelity with QCL-QPE is still higher than the standard QPE. In both implementations, the decay of the fidelity for five-bit precision can be explained by the number of gates approaching the limit supported by current devices.

Given these results, in order to optimize the portfolio with NISQ-HHL, we will run the separate QCL-QPE procedure, Step (a), for estimating the eigenvalues to four bits. Apart from this separate component, NISQ-HHL will use the standard QPE circuit in Step (d) of the algorithm with three ancillas. In the eigenvalue inversion circuit, we will map the four-bit estimates to three-bit estimates to be represented by three ancillas in the HHL circuit, as explained in Section III-A.

Now that we have determined the bit precision for both the standard QPE and the QCL-QPE components, in Section IV-B we show how to select the set of estimates of the relevant eigenvalues. This is accomplished, in Step (b), by classically post-processing the output probability distribution of QCL-QPE executed on hardware.

B. QCL-QPE for Estimating the Eigenvalues

For QCL-QPE to effectively estimate the relevant eigenvalues of A, we need to optimize the scaling parameter γ. We started with an initial guess of 0.02 for λ_{max}, corresponding to $\gamma = 50$. We verified with Algorithm 2 that $\alpha = 0.02$ was indeed an overestimation of λ_{max}, as required by Algorithm 1.

Following Algorithm 2, in the case of negative eigenvalues, we tested the validity of the initial guess $\gamma = 50$ with precision

$n = 4$ by running QCL-QPE with $\Gamma = \gamma \cdot 2^{-4} = 50 \cdot 2^{-4}$. The output probability distribution, shown in Figure 4 (a), is concentrated around zero, thus $\gamma = 50$ is a valid guess. An example for an invalid input would be $\gamma = 3200$. The output distribution of QCL-QPE with $\Gamma = \gamma \cdot 2^{-4} = 200$ is plotted on Figure 4 (b). As this distribution is not concentrated around zero, $\gamma = 1600$ is not a valid guess.

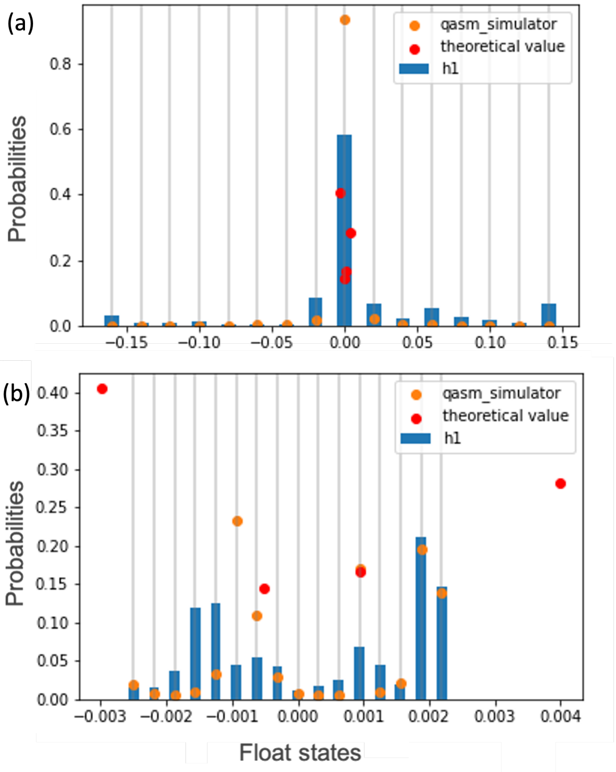


Fig. 4. Probability distributions over the four-bit eigenvalue estimates from the QCL-QPE executions using $e^{iA2\pi\Gamma}$ for $\Gamma = 50 \cdot 2^{-4}$ (a) and $\Gamma = 200$ (b). The blue bars represent the experimental results on the H1 machine - 2000 shots (a) and 2000 shots (b). The theoretical (classically calculated) eigenvalues are represented with the red dots and the results from the Qiskit QASM simulator are represented with the orange dots. On (b) as the theoretical eigenvalues exceed the values we can encode, the distribution we observed shows that values overflowed.

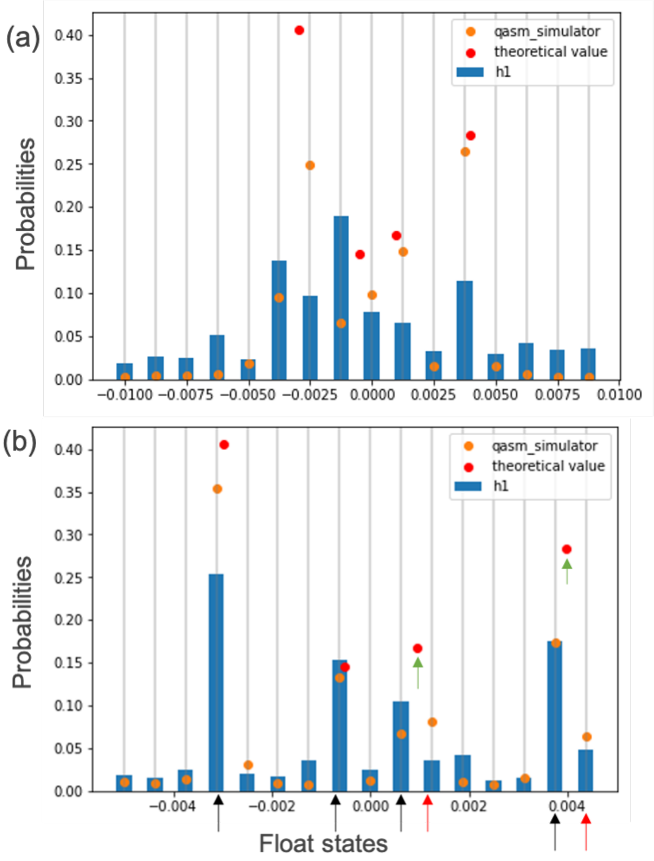


Fig. 5. Probability distributions over the four-bit eigenvalue estimates from the QCL-QPE run using $e^{iA2\pi\gamma}$ for $\gamma = 50$ (a) and $\gamma = 100$.The blue bars represent the experimental results on the H1 machine - 2000 shots (a), 1000 shots (b). The theoretical (classically calculated) eigenvalues are represented with the red dots and the results from the Qiskit QASM simulator are represented with the orange dots.

Now that we confirmed $\gamma = 50$ is a valid guess, we can execute Algorithm 1. We ran QCL-QPE for estimating the eigenvalues of A with this value of γ, and we used $|b\rangle$ as the initial state. The output probability distribution is displayed in Figure 5 (a). The x-axis is binned into 16 values, which are all of the possible four-bit estimates in decimal. They are represented by the grey, vertical lines. In the experiment, we only observed significant probabilities (blue bars) for states within the range $[-0.005, 0.005]$.

Thus, in order to better distinguish the eigenvalues, we decreased the distance between bins by increasing the scaling factor γ. We did so by using our scale optimization algorithm, Algorithm 1, which increased γ to 100. As explained before, we overestimated $\lambda_{\max}$ and hence underestimated γ to avoid overflow.

We can see in Figure 5 (b) that using $\gamma = 100$ makes the x-axis range smaller than in (a), while keeping the same number of bins. As a consequence, the bin intervals are smaller, and there is a better agreement between the theoretical eigenvalues, which are classically calculated, and the experimental probabilities. Moreover, we can see close concordance between the experimental and the simulation results (orange dots). The ability to better distinguish eigenvalues in (b) over (a) shows the importance of the scale optimization procedure. For clarification, the distribution of classically calculated, theoretical, values in the plots are distributed according to $\{|\beta_i|^2\}_{i=0}^{N-1}$, as mentioned at the beginning of Section III.

Once we have obtained the probability distribution over the eigenvalue estimates with the optimal γ, we classically post-processed this distribution, displayed in Figure 5 (b), to select the estimates of the elements in Λ_b. These values were used to construct the rotations for the eigenvalue inversion circuit. As QPE uses the QFT, which is a discrete Fourier transform, we expect non-zero probability of non-relevant states even with a noiseless quantum computer. Since these eigenvalues would need an infinite precision to be fully represented, we expect probability mass to disperse to neighboring states.

In the general case, NISQ-HHL uses the the following technique to select the states that best represent the relevant eigenvalues (i.e., Λ_b). Given the definition of Λ_b (refer to Section III), we are looking for each eigenvalue, λ_i, that satisfies $|\beta_i/\lambda_i| > \epsilon$. In this particular case, given the four-bit precision, with one bit encoding the sign, the factor $1/\lambda_i$ will be at most 2^3, and hence it is not significant enough to consider using the mentioned technique. However, a noise-threshold can be used to remove states with close-to-zero probability in the distribution displayed in Figure 5 (b). We then picked states whose probabilities are significantly higher than this threshold. It is straightforward to see in the distribution that the four states indicated with black arrows have probabilities that are higher than the noise-threshold.

We can see that the two largest eigenvalues, computed classically and indicated with the red dots and the green arrows in Figure 5 (b) are in the middle of two vertical lines, which represent possible eigenvalue estimates. As a consequence, we can state that the probabilities corresponding to each of these eigenvalues are split into their neighboring states. As such, we can also consider the states with higher probability indicated with the red arrows as estimates of the elements in Λ_b.

Note that we have detected this probability splitting in neighboring states by looking at the classically calculated eigenvalues. As the eigenvalues might have an infinite binary decomposition, thus we expect this probability splitting to exist regardless of the precision. Therefore, we also used the Qiskit QASM quantum simulator. The probabilities of the neighboring states calculated on simulator (indicated with the red arrows) are significantly higher than the noise-threshold for the simulation results. This allows us to consider these states as estimates of the elements in Λ_b. In addition, more complex resolution methods, such as multiple scaling factors, might be used in future work.

To sum up, we have built two possible sets of four-bit estimates to represent the elements in Λ_b. These are: the four estimates indicated with the black arrows and the six estimates indicated with both the black and the red arrows. Later on in the article we will compare the performance of NISQ-HHL circuits using these two sets.

Before implementing the end-to-end NISQ-HHL circuit, we have to decide the number of ancillary qubits. As we have already benchmarked the QPE component in Section IV-C, we will look at the eigenvalue inversion component in the next section.

C. NISQ-HHL Eigenvalue Inversion Circuit Performance

If we run the HHL circuit up to the eigenvalue inversion component, ideally a measurement of the register T will return an n-bit estimate of one of the relevant eigenvalues, and S will return a superposition of the corresponding eigenvectors, as shown in the final state in (2). Unfortunately, this is not necessarily true, mainly due to hardware noise. To quantify this noise as a function of the number of ancillas and determine the size of the eigenvalue register in NISQ-HHL, we tested this component.

To do so, we ran circuits that perform the transformation represented by (3), with the register T consisting of two, three or four qubits. For these experiments, we used one of the eigenvalues of A, λ, that was estimated to four bits in Section IV-B. The estimate was truncated in the cases where T consisted of two or three qubits. We took $C := \min\{|\tilde{\lambda}_i|\}_{i=0}^{m-1}$, and defined the corresponding rotation angle to be $2\arcsin(C/\tilde{\lambda})$.

In addition, the register S was initialized with the eigenstate $|u\rangle$ corresponding to λ. The first term of the final state in the mapping in (3) represents the desired output state. Whereas, the second term represents the presence of noise.

$$\begin{aligned} |0\rangle \otimes |\tilde{\lambda}\rangle_T |u\rangle_S \to & \\ & \left(\sqrt{1-\frac{C^2}{\tilde{\lambda}^2}}\,|0\rangle + \frac{C}{\tilde{\lambda}}\,|1\rangle\right) \otimes |\tilde{\lambda}\rangle_T |u\rangle_S \\ & + \sum_{j=0}^{M} (\alpha_j |0\rangle + \beta_j |1\rangle) \otimes |\eta_j\rangle_T |\sigma_j\rangle_S \end{aligned} \quad (3)$$

We ran the circuits on the Honeywell H1 system, and we measured all registers. Figure 6 displays the probability distributions of certain events that occur after measuring all of the registers for each of the three circuits executed. The events “0” (success) and “1” (failure) correspond to the post-measurement states $|0\rangle |\tilde{\lambda}\rangle_T |u\rangle_S$ and $|1\rangle |\tilde{\lambda}\rangle_T |u\rangle_S$ respectively (first term in the final state mapping in (3)). Tracing out the rotation ancilla, the event “other” corresponds to the post-measurement state of registers T and S not being the product state $|\tilde{\lambda}\rangle_T |u\rangle_S$ (second term of the final state in (3)).

We conclude that increasing the number of ancillary qubits, which are the ancillas in the end-to-end HHL, raises the probability of “other”. This represents the probability of measuring noise in both the T and S registers. The noise is represented by states that do not correspond to the estimated eigenvalue and its eigenstate. Moreover, we see that for four ancillas the experimental probabilities and the theoretical values represented by the dotted lines disagree.

Even though it seems that the best choice is to use two ancillas, it will not provide enough precision for the QPE to separate the eigenvalues for a proper eigenvalue inversion. Therefore, we decided to employ three ancillas. This result is in agreement with the conclusion reached from the results of the standard QPE benchmarks in Section IV-A.

Now that we have determined the number of ancillas for the NISQ-HHL circuit, we implemented it and we studied the number of controlled rotations and the circuit depth’s, in terms of ZZMax, of the eigenvalue inversion circuit. In Table III, we show a comparison for the three implementations: the uniformly controlled rotation gate and the proposed NISQ-HHL eigenvalue inversion circuit, in which the rotations are conditioned on four and six estimated eigenvalues respectively

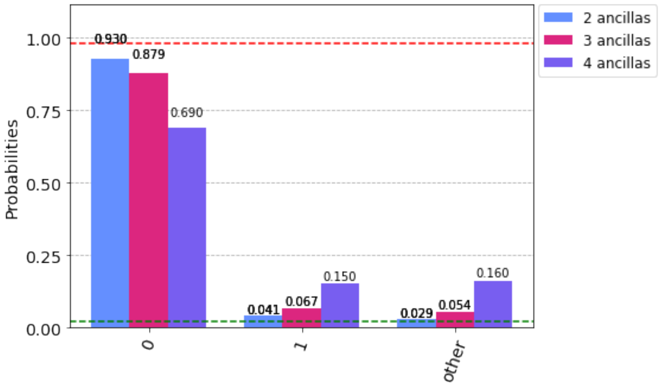


Fig. 6. Probability distributions of previously mentioned events that occur after measuring all of the registers after the eigenvalue inversion on the Honeywell H1 device with 1000 shots. The red dotted line represents the theoretical probability of the failure ("0") event and the green dotted line that of success ("1") event.

(refer to Section IV-B). In all the cases, the circuits consisted of three control ancillary qubits and one rotation ancillary qubit.

	Uniformly	NISQ-HHL	NISQ-HHL
Rotations	7	4	6
Depth	138	80	120

TABLE III
COMPARISON OF THE NUMBER OF ROTATIONS AND ZZMAX DEPTH OF THE DIFFERENT EIGENVALUE INVERSION CIRCUITS.

The number of rotations in each approach was determined as follows. According to the classical post-processing, we implemented two NISQ-HHL circuits conditioning the rotations on two sets of estimates of Λ_b respectively, one with four estimates and another one with six. In this approach, one could rotate on $|0\rangle$ as it can represent an eigenvalue that got rounded to 0 when reducing the bit precision from n to r bits. For example, in Figure 5 (b) the first black arrow to the right of the x-axis origin, corresponds to the $|000\rangle$ state when represented with three bits. The rotation angle is determined based on the four-bit representation $|1111\rangle$, i.e., -1 in two's complement representation. In contrast, the uniformly controlled rotation approach excludes the zero state, for the reasons explained in Section III. As a consequence, in this approach the rotations employing three control ancillas are conditioned on seven eigenvalue estimates.

We see in Table III that the NISQ-HHL eigenvalue inversion circuits are significantly shallower: a reduction of 42% and 13% in depth when conditioning on four and six estimates respectively. What is more, the angles for the controlled rotations $\{\theta_i\}_{i=0}^{m-1}$ are estimated from eigenvalue estimates of Λ_b, which utilized more precision in our approach. Therefore, for these two reasons, we expect to obtain a more accurate estimation of the solution to the linear system by executing NISQ-HHL on NISQ devices.

A way of comparing the NISQ-HHL estimation of the best portfolio to the classical calculated solution is by using a controlled-SWAP test [16] between the quantum states that represent them.

D. SWAP Test Between the Portfolio State and Classical Solution

The controlled-SWAP test can be used to compute the magnitude of the inner product between the quantum state that represents the allocation vector produced by NISQ-HHL and the classical solution loaded onto a quantum state [34].

We added a new qubit to the NISQ-HHL circuit called the *swap ancillary qubit*, and we loaded the normalized classically calculated solution to the linear system, i.e., $\vec{x_c}$, onto this qubit. Then we ran the NISQ-HHL algorithm and the controlled-SWAP test between the HHL output state $|x\rangle$ and the quantum state encoding the classical solution $|x_c\rangle$.

We executed the circuits in the Qiskit statevector simulator and on the Honeywell H1 hardware for the three different approaches discussed before: the uniformly controlled rotation gate and NISQ-HHL conditioning on four and six estimates respectively. We measured the rotation and swap ancillary qubits. With the results from the measurements of these two qubits using 3000 shots, we calculated the inner product between the quantum states as: $\sqrt{2\frac{P(|10\rangle)}{P(|10\rangle)+P(|11\rangle)}} - 1$, where P represents the probability of measuring the respective quantum state. The most-significant qubit corresponds to the rotation ancilla, where a post-measurement state of $|1\rangle$ corresponds to success.

We show the inner products as well as the circuit depth's and the number of controlled rotations for the three approaches, in Table IV. Regarding the simulation results, the NISQ-HHL eigenvalue inversion rotations conditioned on six eigenvalues increased the inner product by 70% over the original method with a uniformly controlled rotation. This is due to the four-bit estimation used in our method for computing the rotation angles, compared to the three-bit precision in the previously mentioned method.

	Uniformly	NISQ-HHL	NISQ-HHL
Rotations	7	4	6
Depth	272	214	254
Simulation	0.59	0.49	0.83
H1	0.42	-	0.46

TABLE IV
COMPARISON OF THE NUMBER OF ROTATIONS IN THE EIGENVALUE INVERSION, ZZMAX DEPTH OF THE HHL PLUS THE CONTROLLED-SWAP TEST CIRCUITS AND THE INNER PRODUCT CALCULATED FROM THE QISKIT STATEVECTOR SIMULATOR AND THE HONEYWELL H1 RESULTS WITH 3000 SHOTS.

We achieved a smaller inner product for NISQ-HHL when conditioning the rotations on four eigenvalue estimates, instead of six, on simulator. The reason for this is that the set of four eigenvalue estimates is an incomplete representation of the elements of Λ_b due to an inefficient classical post-processing over the probability distribution of the eigenvalue estimates displayed in Figure 5 (b). As discussed, not all of the classically calculated values coincide with the experimental results. In order to reach a better inner product, we had to detect that, in some cases, the probability mass was split between the closest approximations to the eigenvalues. This was possible by analyzing the results of the Qiskit QASM simulator together with experimental results. This shows the importance of an efficient classical post-processing that allows for determining the controls and angles for the rotations in the eigenvalue inversion step.

We can see from the results of the experiments on the Honeywell System Model H1 that the calculated inner products are significantly smaller than the ones calculated using the Qiskit statevector simulator. This stems from the fact that the circuits executed are very deep for NISQ devices. Nevertheless, NISQ-HHL, controlling on the six estimates, performed better than the uniformly controlled rotation version. This is because our novel approach employs more precision in bits for the rotation angles, and the shallower circuits used are less prone to hardware noise.

V. Scaling up NISQ-HHL

The aim of this section is to present how NISQ-HHL can be easily applied to a portfolio of any given size. Particularly, we show how the algorithm works for more assets. We considered two portfolio-optimization problems consisting of 6 and 14 S&P 500 assets respectively. We ran NISQ-HHL for these two problems followed by the controlled-SWAP test on the Qiskit statevector simulator. The results from the controlled-SWAP test were used to calculate the inner product between the quantum state produced by HHL and the optimal quantum state computed classically.

We ran Step (a) and (b) in Figure 1, for both problems. The rotations in the eigenvalue inversion circuits were condition on these estimates respectively: four estimates for the 6 assets problem and five for the 14 assets problem. In both cases we used six ancillary qubits.

Even though we are running fewer controlled rotations in NISQ-HHL, the number of qubits required is still significant: 14 for 6 assets and 16 for 14 assets. The circuits are also still very deep. These characteristics prevent us from running these experiments on real hardware. Nevertheless, we can calculate the inner products using the Qiskit statevector simulator. We compare the number of controlled rotations, the circuit depth's and the inner products, in Table V.

We can see that for both sets of assets, the inner products calculated with NISQ-HHL are significantly high (very close to one). The inner products obtained for the 14 assets problem are slightly higher than for the 6 assets one. A reason for this is that the eigenvalue inversion implementation depends on the classical post-processing of the probability distribution obtained with the separate QCL-QPE routine. And in some cases, this processing may estimate the elements of Λ_b more accurately than in others. A future research project could be improving the post-processing technique to better identify the elements of Λ_b.

Moreover, the number of rotations, and as a result the circuit depth's, are an order of magnitude less in the NISQ-HHL implementation in comparison to the uniformly controlled rotation circuit. These results shows that when the hardware will be able to support the specified numbers of qubits and circuit depth's, NISQ-HHL will achieve very good results.

	NISQ-HHL	Uniform	NISQ-HHL	Uniform
Assets	6	6	14	14
Rots	4	64	5	64
Depth	1877	12911	6514	11786
Prod	0.86	0.92	0.98	0.95

TABLE V

Comparison of HHL plus controlled-SWAP test circuit characteristics using NISQ-HHL and the uniformly controlled rotation gate ("Uniform") for the 6 and 14 assets problems. We compare the number of controlled rotations ("Rots"), the ZZMax depth and the inner product ("Prod") calculated using measurements collected from the Qiskit statevector simulator.

This statement holds under the assumption that the vector $\vec{b}$ can be efficiently loaded onto a quantum state, and the matrix A is sparse and well-conditioned. We studied how the condition number (κ) of A scales with the number of assets in the portfolio (N).

We first used daily prices from S&P 500 stocks starting from 2019 to August 2021 to build A. Using linear regression and discarding the first elements, we obtained an exponential fitting: $\kappa = 10^{0.011N+6.439}$ with $R^2 > 0.988$. This result is very dependent on the time interval considered. When considering a bigger time-frame, from 2018 to August 2021, we found a quadratic fitting: $\kappa = (0.315N + 1.4)^2$ with $R^2 > 0.987$. The condition number as a function of the number of assets for this time-frame is shown in Figure 7. It is also quadratic if we consider prior years as the starting point of the interval. Additionally, the fitting coefficient is even lower if we keep on extending to the past. This was tested up to 2015, and in this case, the condition number is halved for 400 assets. For all of these scenarios considered, conditioning methods [5, 6] will be needed to harness the theoretically proven speedup. Note, we observed that, in practice, scaling each component of $A : (\vec{r}, \vec{p}, \Sigma)$ to the same order of magnitude, reduces the condition number of A. This transformation is possible as it only impacts the Lagrange multipliers, which are not relevant to the resulting portfolio. Notice that $\vec{b}$ will have to be scaled in the same way.

VI. Related Work

Variational algorithms have been developed in the last couple of years for optimization problems, such as the Variational Quantum Eigensolver (VQE) [35] and Quantum Approximation Optimization Algorithm (QAOA) [18]. They rely on heuristics. Although their speedup is unproven, they have shown great results when obtaining the ground-state of molecules on NISQ devices [36]. More specifically, a variational quantum linear solver [20] has been introduced last year.

Regarding quantum annealing computation, a hybrid algorithm for *dynamic portfolio optimization with minimal holding*

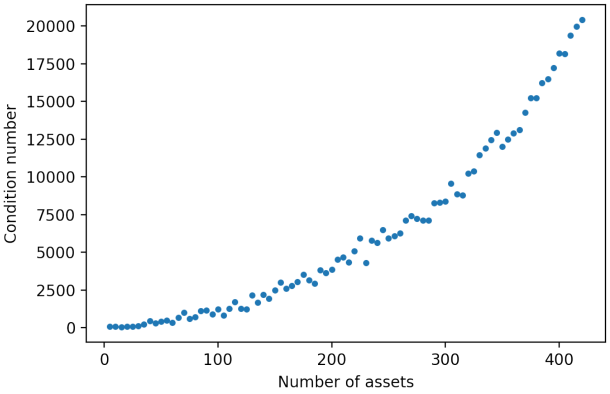


Fig. 7. Condition number of the matrix A as a function of the number of the S&P 500 assets considered. A is built with the assets' historical data from 2018 to August 2021.

period was executed on the D-Wave quantum annealer processor [37]. However, they are solving a different objective function structure to the one introduced in this article. Like variational algorithms, there is no guaranteed speedup.

Another approach is Grover Adaptive Search (GAS) that has been used for global optimization problems in which Grover's algorithm [38] is employed to efficiently implement Pure Adaptive Search (PAS) [39]. Its speedup has been proven to be quadratic.

Contrary to these algorithms, HHL, assuming the normalized $\vec{b}$ can be efficiently loaded onto a quantum state and A is sparse and well-conditioned, has been proven to bring about exponential speedup in N, i.e., the dimension of the system, over the best classical linear solver. It is important to mention that in order to achieve this exponential speedup, we can not access the solution directly using exhaustive techniques such as quantum state tomography. Instead, we could sample from the quantum state to get a good approximation to the allocation vector. Another approach [2] is to use the quantum portfolio state to make calculations that are of interest to the financial industry without classically accessing it. However, HHL is still cumbersome to deploy on NISQ devices. NISQ-HHL, introduced in this article, intends to bring it closer to implementation.

VII. Conclusions

In this work, we introduced NISQ-HHL, a novel extension of the HHL algorithm that incorporates an eigenvalue inversion component suitable for NISQ devices. We executed it on a NISQ device, the trapped-ion Honeywell System Model H1, to optimize a S&P 500 portfolio by casting this problem as a QLSP.

The first step of NISQ-HHL consists of using QCL-QPE to obtain estimates of the relevant eigenvalues Λ_b. This variant of QPE uses mid-circuit measurement, qubit reset and reuse, and Quantum Conditional Logic (QCL). Contrary to the standard implementation, QCL-QPE reduces the number of ancillas to just one for an arbitrary bit precision. As a consequence, we can reduce the number of qubits required dramatically, which is crucial for near-term hardware. Moreover, QCL-QPE replaces two-qubit gates with one-qubit gates controlled by classical bits.

We experimentally showed that QCL-QPE achieves high fidelity, between experimental and simulated measurement distributions, for three-bit precision. Particularly, for four-bit estimations, QLC-QPE achieved a higher fidelity than the standard QPE. What is more, we developed an algorithm that optimizes the scaling parameter γ for the Hamiltonian simulation required by QCL-QPE. We showed that the scaling of A by the γ, obtained with the algorithm, enabled resolving the relevant eigenvalues in the output distribution of QCL-QPE more accurately. These estimates of the relevant eigenvalues are used to implement a near-term efficient eigenvalue inversion circuit, in which the rotations are conditioned on them. In comparison to the uniformly controlled rotation approach, the number of rotations in the NISQ-HHL implementation is smaller, and as a consequence, the circuit is significantly shallower.

We empirically demonstrated the validity of NISQ-HHL. We obtained, with great fidelity, the optimal allocation vector represented as a quantum state for a portfolio-optimization problem with two S&P 500 assets by executing NISQ-HHL on the Honeywell H1 system. We also showed that the NISQ-HHL eigenvalue inversion circuit is significantly more efficient than the uniformly controlled rotation gate method. This is because we reduced the number of controlled rotations, and as a consequence, the circuit depth's.

Moreover, we calculated the inner product between the quantum portfolio state and the classically calculated solution loaded as a quantum state. Using NISQ-HHL we obtained a higher inner product than the uniformly controlled rotations method when executing on hardware and in the Qiskit statevector simulator. Finally, we showed that NISQ-HHL can be easily applied to a portfolio of any given size.

Acknowledgments

We thank Tony Uttley, Brian Neyenhuis and the rest of the Honeywell Quantum Solutions team for assisting us on the execution of the experiments on the Honeywell System Model H1. We also thank Michele Mosca and his team at University of Waterloo for their insights, and Aram Harrow from Massachusetts Institute of Technology for his precious feedback.

Disclaimer

This paper was prepared for information purposes by the Future Lab for Applied Research and Engineering (FLARE) group of JPMorgan Chase Bank, N.A.. This paper is not a product of the Research Department of JPMorgan Chase & Co. or its affiliates. Neither JPMorgan Chase & Co. nor any of its affiliates make any explicit or implied representation or warranty and none of them accept any liability in connection with this paper, including, but limited to, the completeness, accuracy, reliability of information contained herein and the potential legal, compliance, tax or accounting effects thereof. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction.

References

[1] A. W. Harrow *et al.*, "Quantum algorithm for linear systems of equations," *Physical Review Letters, 103(15):150502, Oct 2009*,

[2] P. Rebentrost *et al.*, "Quantum computational finance: Quantum algorithm for portfolio optimization," *arXiv:1811.03975 [quant-ph] 2018*,

[3] H. Markowitz, "Portfolio selection," *The Journal of Finance, 7(1):77-91, 1952*,

[4] J. R. Bunch *et al.*, "Some stable methods for calculating inertia and solving symmetric linear systems," *Mathematics of computation, 1977*,

[5] B. D. Clader *et al.*, "Preconditioned quantum linear system algorithm," *Physical Review Letters, 110(25):250504, Jun 2013*,

[6] C. Shao *et al.*, "Quantum circulant preconditioner for a linear system of equations," *Physical Review A, 98(6):062321, 2018*,

[7] S. Aaronson, "Read the fine print," *Nature Physics, 11(4):291–293, 2015*,

[8] J. Preskill, "Quantum computing in the nisq era and beyond," *Quantum, 2:79, 2018*,

[9] L. Ruiz-Perez *et al.*, "Quantum arithmetic with the quantum fourier transform," *Quantum Information Processing, 16(6):152, 2017*,

[10] M. Möttönen *et al.*, "Transformation of quantum states using uniformly controlled rotations," *Quantum Info. Comput., 5(6):467–473,September 2005*,

[11] Y. Lee *et al.*, "Hybrid quantum linear equation algorithm and its experimental test on ibm quantum experience," *Scientific reports, 9(1):1–12, 2019*,

[12] J. Pino *et al.*, "Demonstration of the trapped-ion quantum ccd computer architecture," *Nature, 592(7853):209–213, Apr 2021*,

[13] A. G. Rattew *et al.*, "The Efficient Preparation of Normal Distributions in Quantum Registers," *arXiv:2009.06601*, 2020.

[14] G. Aleksandrowicz *et al.*, "Qiskit: An open-source framework for quantum computing," *10.5281/zenodo.2562111*, 2019.

[15] A. Asfaw *et al.* (2020). "Learn Quantum Computation Using Qiskit," [Online]. Available: http://community.qiskit.org/textbook.

[16] R. G. Beausoleil *et al.*, "Tests of quantum information," *US Patent 7,559,101 B2*, 2008.

[17] N. Moll *et al.*, "Quantum optimization using variational algorithms on near-term quantum devices," *Quantum Science and Technology, 3(3):030503, 2018*,

[18] E. Farhi *et al.*, "A quantum approximate optimization algorithm," *arXiv:1411.4028 [quant-ph] 2014*,

[19] A. G. Rattew *et al.*, "A Domain-agnostic, Noise-resistant, Hardware-efficient Evolutionary Variational Quantum Eigensolver," *arXiv*, arXiv–1910, 2019.

[20] C. Bravo-Prieto *et al.*, *Variational quantum linear solver*, 2020. arXiv: 1909.05820.

[21] H.-Y. Huang *et al.*, "Near-term quantum algorithms for linear systems of equations," *arXiv preprint arXiv:1909.07344*, 2019.

[22] I. Kerenidis *et al.*, "A Quantum Interior Point Method for LPs and SDPs," *ACM Transactions on Quantum Computing*, vol. 1, no. 1, pp. 1–32, 2020.

[23] A. Gilyén *et al.*, "Quantum singular value transformation and beyond: Exponential improvements for quantum matrix arithmetics," in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, 2019, pp. 193–204.

[24] T. Häner *et al.*, "Optimizing quantum circuits for arithmetic," *arXiv:1805.12445 [quant-ph] 2018*,

[25] M. Möttönen *et al.*, "Quantum circuits for general multiqubit gates," *Physical review letters*, vol. 93, no. 13, p. 130 502, 2004.

[26] H. Mohammadbagherpoor *et al.*, "Experimental challenges of implementing quantum phase estimation algorithms on ibm quantum computer," *arXiv:1903.07605 [quant-ph] 2019*,

[27] R. B. Griffiths *et al.*, "Semiclassical fourier transform for quantum computation," *Phys. Rev. Lett., 76(17):3228–3231, Apr 1996*,

[28] S. Beauregard, "Circuit for shor's algorithm using 2n+3 qubits," *Quantum Info. Comput., 3(2), March 2003*,

[29] M. Mosca *et al.*, "The hidden subgroup problem and eigenvalue estimation on a quantum computer," *In NASA International Conference on Quantum Computing and Quantum Communications, pages 174–188. Springer, 1998*,

[30] M. Dobšíček *et al.*, "Arbitrary accuracy iterative quantum phase estimation algorithm using a single ancillary qubit: A two-qubit benchmark," *Physical Review A, 76(3):030306, Sep 2007*,

[31] I. Kerenidis *et al.*, "Quantum gradient descent for linear systems and least squares," *Physical Review A*, vol. 101, no. 2, Feb. 2020, ISSN: 2469-9934. DOI: 10.1103/physreva.101.022316. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.101.022316.

[32] S. Sivarajah *et al.*, "T—ket⟩: A retargetable compiler for nisq devices," *Quantum Science and Technology, 6(1):014003, 2020*,

[33] M. A. Nielsen *et al.*, *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[34] H. Buhrman *et al.*, "Quantum fingerprinting," *Physical Review Letters, 87(16):167902, 2001*,

[35] A. Peruzzo *et al.*, "A variational eigenvalue solver on a photonic quantum processor," *Nature communications 2014*,

[36] Q. Gao *et al.*, “Computational investigations of the lithium superoxide dimer rearrangement on noisy quantum devices,” *The Journal of PhysicalChemistry A, 125(9):1827–1836, 2021*,

[37] S. Mugel *et al.*, “Hybrid quantum investment optimization with minimal holding period,” *arXiv preprint arXiv:2012.01091*, 2020.

[38] L. K. Grover, “A fast quantum mechanical algorithm for database search,” *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing, pages 212–219, 1996*,

[39] W. P. Baritompa *et al.*, “Grover’s quantum algorithm applied to global optimization,” *SIAM Journal on Optimization, 15(4):1170–1184, 2005*,