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

Romina Yalovetzky, Pierre Minssen, Dylan Herman, and Marco Pistoia

Future Lab for Applied Research and Engineering, JPMorgan Chase Bank, N.A.

Portfolio optimization is an essential use case in Finance, but its computational complexity forces financial institutions to resort to approximated solutions, which are time consuming. Using the method of Lagrange multipliers, the mean-variance portfolio optimization problem can be represented by 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, qubit reset and reuse, and Quantum Conditional Logic (QCL). To the best of our knowledge, NISQ-HHL is the first algorithm incorporating a QCL-enhanced version of Phase Estimation executed on real hardware, the trapped-ion Quantinuum System Model H1-1. In addition, NISQ-HHL includes a novel method for choosing the optimal evolution time for 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.

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

1 Introduction

The Quantum algorithm for solving linear systems of equations named HHL, introduced by Harrow, Hassidim and Lloyd [1], has been proposed as a solver for a specific portfolio-management problem [2] known as *mean-variance portfolio optimization*. Under some assumptions [3], HHL provides an exponential speedup in N over all known classical algorithms for Quantum Linear Systems Problems (QLSPs) of size N [1]. Since, for financial applications, time is of the essence, this speedup in the overall execution time can be extremely impactful.

However, HHL is known to be cumbersome to deploy [3], especially on NISQ hardware [4]. 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 [5]. 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*, whose n-qubit circuit depth is $\Omega(2^n)$ [6]. This paper introduces NISQ-HHL, a novel enhancement of the classical/quantum hybrid HHL variant introduced by Lee *et al.* [7]. By leveraging newly available hardware features and optimizing existing components of HHL, NISQ-HHL 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 suitable for near-term devices, are generally applicable to any problem that can be solved leveraging HHL including machine learning algorithms [8, 9].

Given a set of $\tilde{N}$ assets, mean-variance portfolio optimization 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. This problem can be stated as a convex quadratic program:

$$\underset{\vec{w}\in\mathbb{R}^{\tilde{N}}}{\text{minimize}}\ \vec{w}^{\mathsf{T}}\Sigma\vec{w}\ :\ \xi = \vec{p}^{\mathsf{T}}\vec{w},\ \mu = \vec{r}^{\mathsf{T}}\vec{w}, \quad (1)$$

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

$$\begin{bmatrix} 0 & 0 & \vec{r}^{\mathsf{T}} \\ 0 & 0 & \vec{p}^{\mathsf{T}} \\ \vec{r} & \vec{p} & \Sigma \end{bmatrix} \begin{bmatrix} \eta \\ \theta \\ \vec{w} \end{bmatrix} = \begin{bmatrix} \mu \\ \xi \\ \vec{0} \end{bmatrix}, \quad (2)$$

where $\eta, \theta \in \mathbb{R}$ are the Lagrange multipliers. We will denote equation (2) 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. The resulting quantum state $|x\rangle = |\eta, \theta, w\rangle$ allows for recovering the subsystem of qubits corresponding to the allocation vector $|w\rangle$.

2 Results

2.1 The NISQ-HHL Framework

One of the contributions of NISQ-HHL is its ability to replace the standard Quantum Phase Estimation (QPE) with a version 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 version of QPE as QCL-QPE.

Given a Hermitian matrix $A \in \mathbb{C}^{N\times N}$, the n-bit estimation of an eigenvalue λ_i of A, with $i \in \{0, 1, \ldots, N-1\}$, can be represented by a binary string of length n, of the form $\tilde{\lambda}_i := \lambda_i^{(n-1)} \ldots \lambda_i^{(1)}\lambda_i^{(0)}$, which can be encoded in a computational basis state as follows: $|\tilde{\lambda}_i\rangle_T := |\lambda_i^{(n-1)} \ldots \lambda_i^{(1)}\lambda_i^{(0)}\rangle_T$, 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 to be invertible. The solution to the QLSP: $A\vec{x} = \vec{b}$, 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. In NISQ-HHL, QCL-QPE is used to estimate the elements of Λ_b, and the rotations in the eigenvalue inversion component are conditioned on these estimates. Estimating the m elements of Λ_b, $\{\tilde{\lambda}_i\}_{i=0}^{m-1}$, requires evolving the propagator $U := e^{iA2\pi\gamma}$ of A, for time $2\pi\gamma$, via Hamiltonian simulation. We define γ as A's *scaling parameter*. Another contribution of NISQ-HHL is a novel verifiable procedure for selecting the optimal γ.

Fig. 1 illustrates the end-to-end flow of NISQ-HHL, with the objective of executing and validating it on a real quantum computer that supports mid-circuit measurement, qubit reset and reuse, and QCL. NISQ-HHL consists of four steps, indicated as (a), (b), (c) and (d) in Fig. 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 devised an algorithm that computes the optimal γ, which we then use to scale A.

(b) Classical post-processing is performed on the resulting histogram to obtain the n-bit estimates $\{\tilde{\lambda}_i\}_{i=0}^{m-1}$ of the m relevant eigenvalues.

(c) The estimates obtained in (b) are used to implement the eigenvalue inversion circuit. The angles for the rotation gates in the inversion circuit are $\{2\arcsin(C/\tilde{\lambda}_i)\}_{i=0}^{m-1}$, where C is a normalization constant. These estimates are also mapped to a smaller number, r, of bits. 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.

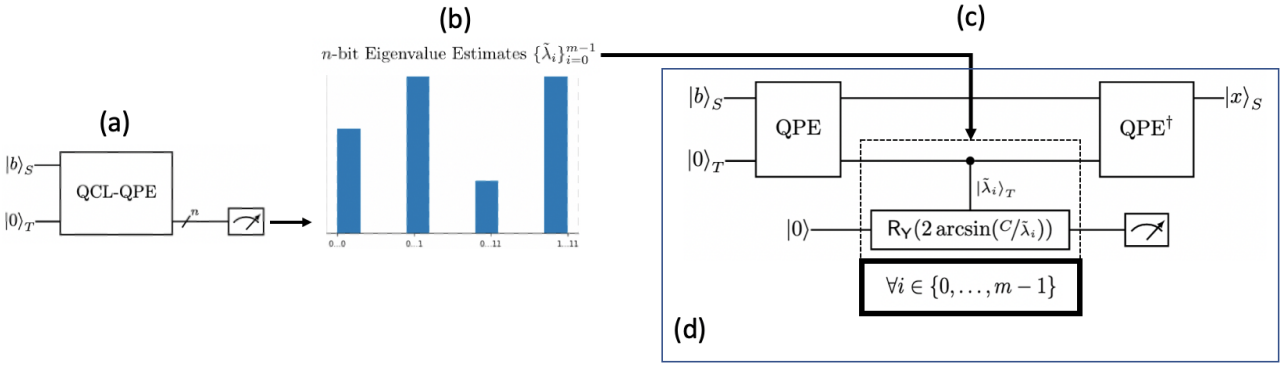

Figure 1: NISQ-HHL end-to-end flow

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 for a low-qubit count than what we would have achieved with some of the state-of-the-art methods, such as quantum arithmetic and uniformly controlled rotation. Therefore, the presented methodology brings HHL closer to being realizable on current quantum hardware.

2.2 Eigenvalue Estimation and Inversion with QCL

The eigenvalue inversion component of the standard HHL algorithm 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:

$$\sum_{i=0}^{N-1} |0\rangle \otimes \beta_i |u_i\rangle_S \otimes |\tilde{\lambda}_i\rangle_T \rightarrow \sum_{i=0}^{N-1} \left(\sqrt{1 - \frac{C^2}{\tilde{\lambda}_i^2}} |0\rangle + \frac{C}{\tilde{\lambda}_i} |1\rangle \right) \otimes \beta_i |u_i\rangle_S \otimes |\tilde{\lambda}_i\rangle_T \tag{3}$$

where C is a normalization constant chosen to be in $O(1/\kappa)$ [1], and κ is the condition number of A. This transformation is accomplished by applying 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 requires either some prior information about the eigenvalues or a coherent computation of the arcsine function using quantum arithmetic [10]. However, quantum arithmetic has not yet been shown to be feasible on NISQ devices. For example, the asymptotically efficient implementation of arcsine made by Häner *et al.* [10] 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 [6]. A uniformly controlled rotation on n qubits decomposes into $2^n - 1$ n-qubit controlled rotations. 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 n is exponential.[1]

Yet another approach is the classical/quantum hybrid solution for HHL introduced by Lee *et al.* that, before running HHL, applies 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, instead of all of the matrix's eigenvalues. The reason is that their am-

[1] The *Gray code* [11] 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 circuit depth is still in $\Theta(2^n)$.

plitudes in the solution of the QLSP, in absolute value, are sufficiently large. Moreover, by controlling the rotations on the estimates of only the relevant eigenvalues, we are reducing the number of controlled rotations in comparison to the uniformly controlled rotation gate approach. 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 $2^n - 1$ n-bit strings, where n is the number of bits used for estimating the eigenvalues. Since the eigenvalues potentially require high precision, it is usually the case that $m \leq N \ll 2^n - 1$. 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. However, QPE is still difficult to implement on NISQ hardware because the number of ancillary qubits grows as the desired bit precision. QPE relies on many controlled gates [12] and fully entangles qubits. Therefore, an all-to-all connected 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) [13–15]. 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 Fig. 2a, which shows how to efficiently estimate the eigenvalues of the unitary operator U to three-bit precision by leveraging mid-circuit measurement, ground-state reset and qubit reuse, and QCL.

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) [16] 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.

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

1. It requires only one ancillary qubit for an arbitrary bit precision.

2. It replaces two-qubit gates with one-qubit gates controlled by classical bits as it can be seen by comparing the two circuits in Fig. 2.

We experimentally demonstrated the benefits of QCL-QPE over the standard QPE on NISQ devices. We considered a portfolio-optimization problem with two S&P 500 assets formulated as a QLSP: $A|x\rangle = |b\rangle$. We benchmarked the performance of both QPE implementations for estimating the eigenvalues of A on the trapped-ion Quantinuum System Model H1-1. We used as the initial state $|b\rangle$, and we set $\gamma = 100$ for both implementations. We compared the empirical distribution of measurements from the execution on hardware to the distribution obtained from the Qiskit QASM simulator by using the fidelity metric [17]. Refer to Section 3.2 for more details on this metric and to Section 3.1 for the experimental set-up.

In Table 1, we compare the number of two-qubit ZZMax gates and qubits for estimating the eigenvalues to different precisions: three, four and five. We compare the ZZMax gates as this is the native two-qubit gate of the Quantinuum System Model H1-1. We also compare the fidelity between the experimental distribution of measurement results and the one obtained with the simulator.

		3-bit	4-bit	5-bit
Standard QPE	Gates	63	88	115
	Qubits	5	6	7
	Fidelity	98.6	90.4	42.6
QCL-QPE	Gates	57	76	95
	Qubits	3	3	3
	Fidelity	98.1	95.0	43.2

Table 1: Comparison of the number of two-qubit ZZMax gates, number of qubits and the fidelity in percentage in both QPE implementations for estimating eigenvalues to different precisions.

The number of qubits in the QCL-QPE does not change with the precision, as opposed to

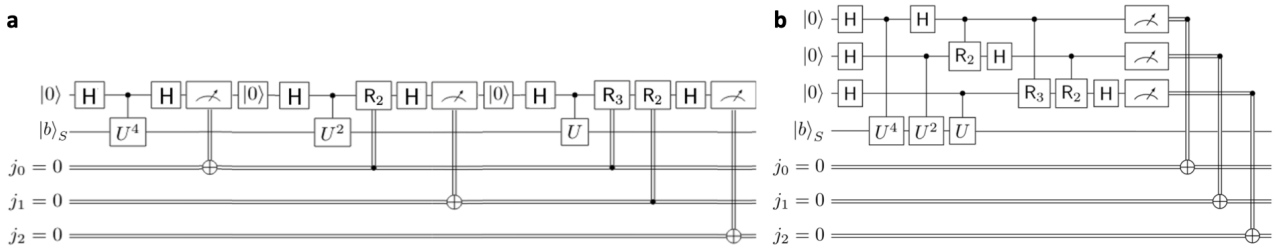

Figure 2: Circuits for estimating the eigenvalues of the unitary operator U to three bits using QCL-QPE (**a**) and standard QPE (**b**). S is the register that U is applied to, and j is a classical register. H refers to the Hadamard gate and $\mathsf{R_k}$, for $\mathsf{k} = 2, 3$, are the phase gates.

the linear growth in the standard implementation. What is more, QCL-QPE replaces the two-qubit gates required by the inverse QFT with classically-controlled one-qubit gates. On hardware, the number of two-qubit gates saved is $n(n-1)$.

It can be seen that the fidelities are similar for three-bit estimations because the number of saved two-qubit gates using QCL-QPE is small. 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 the QCL-QPE circuits are shallower, its fidelities remain consistently higher. In both implementations, the decay in fidelity for five-bit precision can be explained by the number of gates approaching the limit supported by current devices.

2.3 Optimizing the Scaling Parameter

Throughout this discussion, we will assume A only has positive eigenvalues. Thus the spectral norm is equal to the maximum eigenvalue: $\|A\|_2 = \lambda_{max}$. Before running QCL-QPE applied to the propagator $U := e^{iA2\pi\gamma}$ of A to estimate the relevant eigenvalues to n bits, the scaling factor γ is determined by executing a novel algorithm, the scale optimization algorithm, embedded in NISQ-HHL that consists of Algorithm 1 and 2. QPE only outputs the normalized phase angle of the eigenvalues of U, which lies in $(0, 1]$. As the matrix A may have eigenvalues not contained in this interval, the scale optimization algorithm's objective is to compute and validate a factor γ that ensures $\|\gamma A\|_2 \leq 1$. However, suppose $\lambda_{max} < 1$, with QPE, we will be wasting qubits to unnecessarily encode values in $(\lambda_{\max}, 1]$. We pursue a more efficient approach, consisting of estimating $\lambda_{\max}$ first, and then scaling A by $\gamma = \tilde{\lambda}_{\max}^{-1}$, so that the maximum eigenvalue of γA is 1. The phase estimation procedure does not estimate all the eigenvalues of A but only the subset of eigenvalues whose corresponding coefficients in the decomposition of $|b\rangle$ are nonzero. Therefore, by denoting $\lambda_{\max,b}$ the maximal value of this subset, the actual optimal value for γ is $\lambda_{\max,b}^{-1}$ instead of $\lambda_{\max}^{-1}$. Note that scaling by γ potentially results in a longer Hamiltonian simulation time.

Algorithm 1: Optimize the selection of γ using n-bit estimations of eigenvalues
Assumption: $\lambda_{\max}/\alpha \leq 1$

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

The algorithm gradually increases γ to minimize the error of estimation, as shown in Fig. 3. Algorithm 1 starts by guessing an overapproximation of $\lambda_{\max}$. We will discuss later how to make

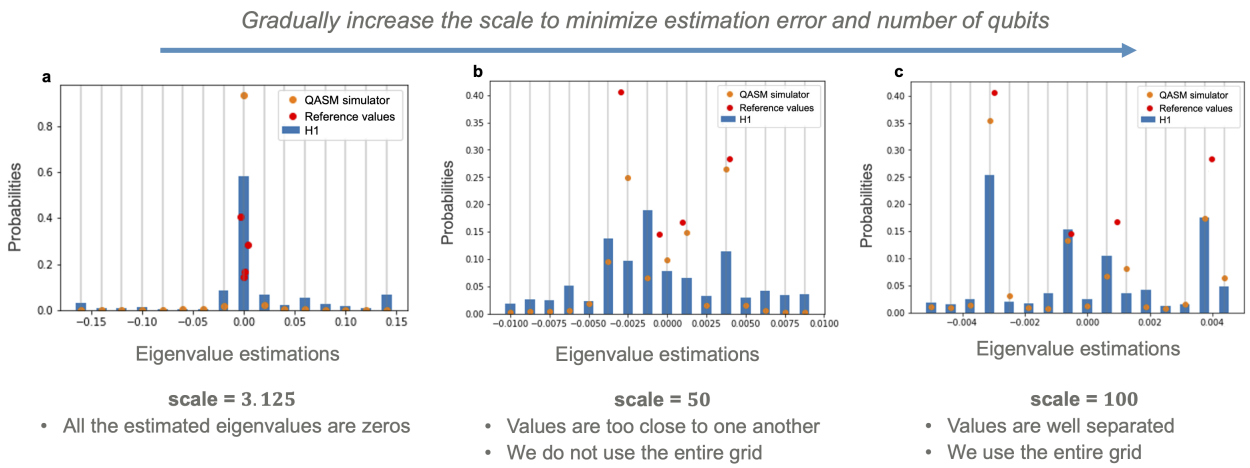

Figure 3: Scaling algorithm mechanism over the QCL-QPE output distributions. The red dots correspond to the classically calculated eigenvalues of A, the orange dots to the probability distributions obtained by the Qiskit QASM simulator, and the blue bars to the experimental results obtained on the Quantinuum System Model H1-1.

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 the γ that was computed during the previous iteration. Specifically, Algorithm 1 post-processes the output distribution to get a new n-bit estimation of $2^n\gamma\lambda_{\max}$ in order to update γ for the next iteration[2]. Fig. 3 displays the QCL-QPE output distributions corresponding to different iterations of Algorithm 1 in which the scale γ is increased until convergence. The number of iterations to find the optimal γ using Algorithm 1 is in $\Theta((1/n)\log_2(\alpha/\lambda_{\max}))$.

In practice we want to underestimate γ at each step in order to reduce the amount of interference that occurs between basis states $|i\rangle$ when i is near the extrema of $\{0,\ldots,2^n-1\}$. Note that n, the number of bits used for estimation, is an input parameter of the algorithm. To represent both the largest and smallest eigenvalues, in theory, we need $n \geq \log_2(\lambda_{\max}/\lambda_{\min}) = \log_2(\kappa)$. However, this requires prior knowledge of κ. In practice, we first run Algorithm 1 with an initial precision of n bits. Then, if $\lambda_{\min}$, 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 $\lambda_{\min}$ 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 $\lambda_{\max}$. One way to achieve this result is to use the Frobenius norm of A, defined as $||A||_{\mathrm{F}} := \sqrt{\mathrm{Tr}(A^\dagger A)}$. Indeed, since A is Hermitian, $||A||_{\mathrm{F}} = \sqrt{\sum_{i=0}^{N-1}\lambda_i^2}$, and so the Frobenius norm of A is a valid overapproximation of $\lambda_{\max}$. 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 $\lambda_{\max}$. 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 $||A||_{\mathrm{F}}$. If the initial guess of α does not satisfy $\alpha \geq \lambda_{\max}$, 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 $\lambda_{\max}$ 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 $||A||_{\mathrm{F}}$. Therefore combining the two algorithms allows us to efficiently find A's scaling parameter γ.

The idea of Algorithm 2 is the following. If α is a valid guess, there will be no overflow in

[2]Any algorithm outputting an n-bit estimation of $2^n\gamma\lambda_{\max}$ 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 the main text.

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 $\lambda_{\max}$

Assumption: at each step, there is at least one eigenvalue of ΓA *not in* $\bigcup_{j\in\mathbb{Z}/\{0\}} \left[j - 2^{-(n+1)},\ j + 2^{-(n+1)}\right]$

```
// Initialize scaling parameter
Γ := 1/(2^{n+1}α)
ℙ := n-bit output distribution of
  QCL-QPE using unitary e^{iA2πΓ} and
  input state |b⟩
if ℙ[0] ≠ 1 then
    Return α is not valid
    // Otherwise all eigenvalues
    // estimations would have been 0
else
    Return α is valid
end
```

Result: Return if α is an overestimation of $\lambda_{\max}$

As mentioned, 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, which tests the validity of the initial guess used in Algorithm 1, 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 may be indefinite.

We note that Kerenidis and Prakash [18] have developed an algorithm to ϵ-approximate $\eta := ||A||_2/||A||_{\mathrm{F}}$. Thus, their algorithm can also be used to find $\lambda_{\max}$ given that $||A||_{\mathrm{F}}$ has been previously computed, which is not required for Algorithm 1. As mentioned before, the optimal γ is $\lambda^{-1}_{\max,b}$ and not $\lambda^{-1}_{\max}$, when given $|b\rangle$ as an initial state. Since Algorithm 1 only computes $\lambda^{-1}_{\max,b}$, this makes it more suitable for NISQ-HHL. 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 previously, is more suitable for NISQ computers.

Lastly, as mentioned in the initial HHL article [1], for numerical stability reasons, we would like to only invert eigenvalues that are within the well-conditioned part of the spectrum of A: $\lambda > 1/\kappa'$, where $1/\kappa' < 1/\kappa$. The reason behind this is that if there is an eigenvalue λ such that $\lambda < 1/\kappa$, when we invert it, a small relative error in λ will cause $1/\lambda$ to deviate from the true value by an amount that is many times κ. This error would dominate all other terms in the sum $A^{-1} = \sum_{i=0}^{N-1} 1/\lambda_i |u_i\rangle \langle u_i|$ making the returned value of A^{-1} deviate from its true value to an unacceptable degree [19]. To deal with ill-conditioning, it has been proposed to use filter functions [1] that act to invert A only on its well-conditioned subspace [19]; a type of interpolation is perform for eigenvalue satisfying $1/\kappa' < \lambda < 1/\kappa$ for some κ', say $\kappa' = 2\kappa$. In our case, because we obtain classically accessible estimates of the eigenvalues before performing HHL, we are able to construct controlled rotations for the eigenvalue corresponding to the different cases of the filter functions. In our simple experiments, we do selectively invert certain eigenvalues but do not apply the filter functions.

2.4 Portfolio Optimization with NISQ-HHL

This section presents empirical results, validating NISQ-HHL by optimizing the previously introduced portfolio of two S&P 500 assets. This problem is formulated as a QLSP: $A|x\rangle = |b\rangle$. According to the benchmark of the QPE implementations discussed previously, we ran the separate QCL-QPE procedure, step (a) in Fig. 1, for estimating the eigenvalues to four bits. Apart from this separate component, NISQ-HHL uses the standard QPE circuit in step (d) of the algorithm. Its implementation for optimizing the portfolio employed three ancillas. In the eigenvalue inversion circuit, we mapped the four-bit estimates to three-bit estimates to be represented by three ancillas in the HHL circuit. By doing this, we managed to run HHL using fewer ancil-

	HHL	NISQ-HHL	Relative Change
Rotations	7	6	-14.29%
Eigenvalue Inversion Depth	138	120	-13.04%
Inner Product in Simulation	0.59	0.83	40.67%
Inner Product on H1-1	0.42	0.46	9.52%

Table 2: Comparison of the eigenvalue inversion circuit of NISQ-HHL and HHL using the uniformly controlled rotation gate and its performance. The magnitude of the inner product were calculated with the Qiskit statevector simulator and the Quantinuum System Model H1-1 with 3000 shots. The circuit depth is calculated in terms of the ZZMax gates. The "Relative Change" column shows the relative change of the value of that particular row obtained with NISQ-HHL in comparison to the one obtained with HHL.

las, while performing matrix inversion with eigenvalues estimates that have higher bit precision. Lowering the number of qubits and circuit depth reduces the exposure to noise. Refer to Section 3.1 for the experimental set-up and to Section 3.2 and Section 3.4 for the explanation on how the scaling parameter of this portfolio-optimization problem was optimized and then used for estimating the eigenvalues.

To quantify the performance of NISQ-HHL, we contrasted the estimation of the best portfolio to the classical calculated solution by executing a controlled-SWAP test [20] between the quantum allocation state produced by NISQ-HHL and the classical solution loaded onto a quantum state. With it, we computed the magnitude of the inner product between the quantum states. Refer to Section 3.5 for more details about this test and its execution.

We compared the performance and circuit characteristics of NISQ-HHL with an HHL implementation that uses the uniformly controlled rotation gate. To do so, we executed both circuits on the Quantinuum System Model H1-1 and in the Qiskit statevector simulator. As we can see in Table 2, the NISQ-HHL eigenvalue inversion circuit is significantly shallower. Its circuit depth is 13.04% shallower than the uniformly controlled rotation. When executed on simulator, NISQ-HHL increased the magnitude of the inner product by 40.67% 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. The inner products calculated with experimental results are significantly smaller than the ones calculated using the simulator. This stems from the fact that the circuits executed are very deep for NISQ devices. Nevertheless, NISQ-HHL performed better than the HHL implementation with the uniformly controlled rotation. This is because our novel approach employs more precision in bits for the rotation angles, and the shallower circuit used is less prone to hardware noise.

NISQ-HHL can be easily applied to a portfolio of larger size. Particularly, we considered two portfolio-optimization problems consisting of 6 and 14 S&P 500 assets respectively. Even though NISQ-HHL consists of fewer controlled rotations, the number of qubits is still significant: 14 for 6 assets and 16 for 14 assets. The circuits are still very deep too. These characteristics prevent us from running these experiments on real hardware. However, we can still calculate the inner products on a simulator. In Table 3, we compare the numbers of controlled rotations, the circuit depths and the inner products. The inner products calculated with NISQ-HHL are significantly high (very close to 1) for both sets of assets. Moreover, the number of rotations, and as a result the end-to-end circuit depths, are an order of magnitude less with NISQ-HHL compared to the uniformly controlled rotation circuit. The inner products obtained for the 14-asset problem are slightly higher than for the 6-asset 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. In some cases, this processing may estimate the elements of Λ_b more accurately than in others, depending on the problem. A future research project could be improving the post-processing technique to better identify the elements of Λ_b.

These results support the evidence that, on a sufficiently powerful near-term quantum computer containing enough qubits and capable of handling the required circuit depths, NISQ-HHL

	HHL	NISQ-HHL	Relative Change	HHL	NISQ-HHL	Relative Change
Assets	6	6	-	14	14	-
Rotations	64	4	-93.75%	64	5	-92.19%
Circuit Depth	12911	1877	-85.46%	11786	6514	-44.73%
Inner Product	0.92	0.86	-6.5%	0.95	0.98	3.16%

Table 3: Comparison of NISQ-HHL and HHL for the problems with 6 and 14 assets. HHL uses the uniformly controlled rotation gate. Both algorithms are augmented with controlled-SWAP test circuits. The comparison includes the inner products calculated using measurements collected from the Qiskit statevector simulator. The "Relative Change" column shows the relative change of the value of that particular row obtained with NISQ-HHL in comparison to the one obtained with HHL.

will produce accurate results when optimizing portfolios with many assets. This statement holds under the assumption that $\vec{b}$ can be efficiently loaded onto a quantum state, and A is sparse and well-conditioned. We studied how the condition number of A (κ) 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 Fig. 4. 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 [21, 22] will be required to harness the theoretically proven speedup of NISQ-HHL when increasing the number of assets. 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.

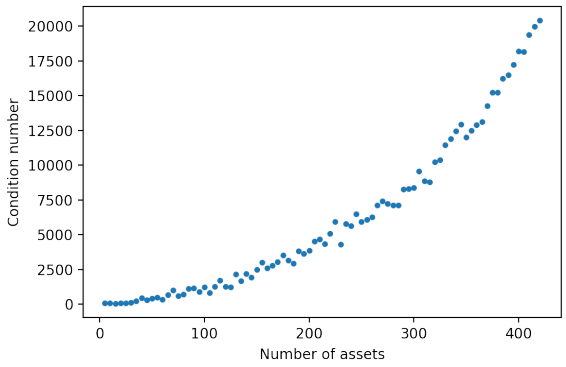

Figure 4: Condition number of the matrix A as a function of the number of the S&P 500 assets considered. A is built with the assets' historical data from 2018 to August 2021.

3 Methods

3.1 Experimental Set-up

For all the experiments conducted on quantum hardware, we considered a portfolio-optimization problem with two S&P 500 assets, formulated as a QLSP $A|x\rangle = |b\rangle$, for demonstrating the validity of NISQ-HHL. After incorporating the two constraints, the matrix A of the linear system is of size 4×4. The experimental results presented were obtained on the trapped-ion Quantinuum System Model H1-1 given its support for mid-circuit measurements, qubit resets and reuses, and QCL [23]. In order to do so, we transpiled and optimized the circuits from Qiskit to H1-1's native gates using Quantinuum's pytket package [24].

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 [25]. In order to perform experiments with NISQ-HHL on quan-

tum hardware, 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 the Qiskit transpiler [26], which decomposes it into basis gates. As the error rates of two-qubit gates are an order of magnitude larger than those of one-qubit gates [23], and the numbers of both gate types are similar in the circuits used, we only present the H1-1 two-qubit gate, ZZMax, counts for the circuits. The ZZMax is equivalent to $\mathsf{R_{ZZ}}(\pi/2)$, and, up to one-qubit gates, it is realized via the Mølmer-Sørensen interaction [23].

Note that, apart from the eigenvalue estimation component, where the standard QPE is replaced by the QCL-QPE, 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.

3.2 Benchmark of Standard QPE and QCL-QPE

In order to quantify the performance of the standard QPE and the QCL-QPE, we compared the empirical distribution of measurement results from the circuit execution on the Quantinuum System Model H1-1 to the distribution obtained from the Qiskit QASM simulator. We compared the probability mass functions, p and q, using the fidelity metric [17]: $F(p,q) = (\sum_i \sqrt{p_i q_i})^2, F(p,q) \in [0,1]$. We expressed it in percentage between the probability distributions from the QPE experiments ran on the Quantinuum System Model H1-1 and in the Qiskit QASM simulation, with 2000 shots each.

3.3 Optimization of the Scaling Parameter for Portfolio Optimization

In order to find the optimal γ to scale the matrix A of the portfolio-optimization problem with two S&P 500 assets, we started with an initial guess of 0.02 for $\lambda_{\max}$, corresponding to $\gamma = 50$. We verified with Algorithm 2 that $\alpha = 0.02$ was indeed an overestimation of $\lambda_{\max}$, 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 Fig. 5a, 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 Fig. 5b. As this distribution is not concentrated around zero, $\gamma = 3200$ is not a valid guess.

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 Fig. 3b. 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. In Fig. 3c, it can be seen that the x-axis range of the distribution using the optimal γ is smaller than that of distributions using a lower γ (Fig. 3a and b). As the number of bins is fixed by the four-bit precision, the bin intervals are smaller and there is a better alignment between the classically calculated reference values and the eigenvalues computed using NISQ-HHL both on real hardware and a simulator. For clarification, the classically calculated values in the plots correspond to $\{|\beta_i|^2\}_{i=0}^{N-1}$.

3.4 Eigenvalue Estimation for Portfolio Optimization with NISQ-HHL

After having optimized the scaling parameter γ, we executed QCL-QPE applied to the propagator $U := e^{iA2\pi\gamma}$ of A with the optimal value of γ. We ran 1000 shots and we built the probability distribution shown in Fig. 6. We did a classical post-processing over the distribution to select the estimates of the elements in Λ_b.

In the general case, NISQ-HHL uses the following technique to select the states that best represent the relevant eigenvalues (i.e., Λ_b). Given the definition of Λ_b, we are looking for each eigenvalue, λ_i, that satisfies $|\beta_i/\lambda_i| > \epsilon$, for a config-

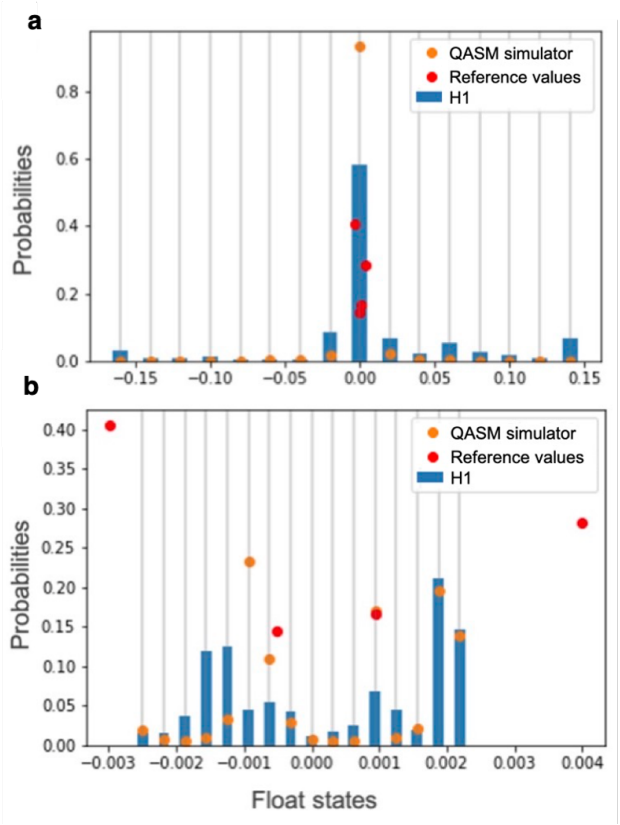

Figure 5: Probability distributions over the four-bit eigenvalue estimates from the QCL-QPE executions using $e^{iA2\pi\Gamma}$ for $\Gamma = 50 \cdot 2^{-4}$ (**a**) and $\Gamma = 200$ (**b**). The blue bars represent the experimental results on the H1-1 machine with 2000 shots. The classically calculated reference values are represented with the red dots and the results from the Qiskit QASM simulator with the orange dots. On (**b**) as the classically calculated eigenvalues exceed the values we can encode, the distribution shows that values overflowed.

urable threshold ϵ. 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 Fig. 6. 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.

As the eigenvalues might have infinitely-long binary representations and they are represented with a finite number of bits, we expect to a see, when estimating a single eigenvalue, a distribution of potential estimates that is not a single point mass. However, in this case, approximately 81% of the probability mass is concentrated around the two closest estimates to an eigenvalue [27], where the actual eigenvalue lies between. Our goal is to identify the two closest estimates to each eigenvalue in the QCL-QPE output distribution returned when given a superposition of eigenvectors as input. Then based on a probability threshold, along with threshold ϵ on $|\beta_i/\lambda_i|$, decide whether to perform controlled rotations using both of the estimates, one of the estimates, or none. In order to see this behaviour, we also analyzed the distribution obtained by executing the circuit in the Qiskit QASM simulator. We can see that the probabilities on simulator of the states indicated with the red arrows are significantly higher than the noise-threshold for the simulation results. This indicates that we should consider these states as estimates of the elements in Λ_b. By doing this, we built a set of six estimates, indicated with both the black and the red arrows, to condition the rotations on in the eigenvalue inversion component. We compare these estimates with the classically calculated eigenvalues, the red dots, in Fig. 6. We see that each of the two largest eigenvalues, indicated with the red dots and the green arrows, are in the middle of two of the estimates in the set we have just built indicated with a red and a black arrow respectively. This shows that we have correctly identified the probability splitting between the closest estimates to the eigenvalues. In addition, this shows that the set of estimates we have built by doing this classical post-processing, are a good approximation of the relevant eigenvalues.

3.5 Benchmark of NISQ-HHL for Portfolio Optimization

A way of comparing the NISQ-HHL estimation of the best portfolio to the classical calculated solution is by using a controlled-SWAP test [20] between the quantum states that represent them. 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. We loaded the normalized classically calculated solution to the linear system, i.e., $\vec{x}_c$, onto a set of ancillary qubits. Then we ran NISQ-HHL

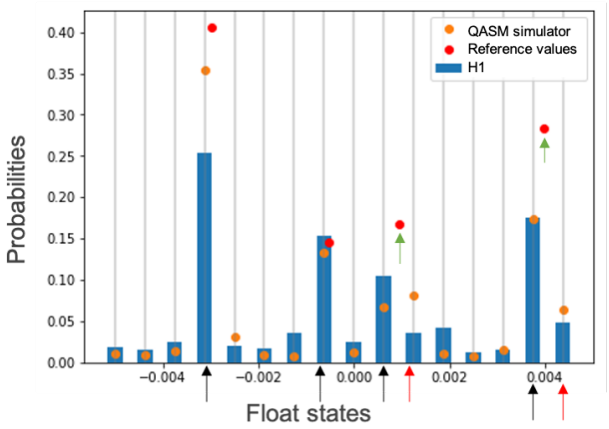

Figure 6: Probability distribution over the four-bit eigenvalue estimates from the QCL-QPE run using $e^{iA2\pi\gamma}$ with $\gamma = 100$. The blue bars represent the experimental results on the H1-1 machine with 1000 shots. The classically calculated reference values are represented with the red dots and the results from the Qiskit QASM simulator with the orange dots.

and the controlled-SWAP test between the NISQ-HHL output state $|x\rangle$ and the quantum state encoding the classical solution $|x\rangle_c$. A controlled-SWAP test requires an extra ancilla (called the swap ancillary qubit), Hadamard gates and controlled SWAP gates. We did this for the NISQ-HHL circuit conditioning the rotations on the relevant estimates of the eigenvalues and for the HHL circuit implemented with the uniformly controlled rotation gate.

We executed the circuits in the Qiskit statevector simulator and on the Quantinuum H1-1 hardware. In all the cases, the circuits consisted of two ancillary qubits that encode $|x\rangle_c$, three control ancillary qubits, one rotation ancillary qubit and one swap ancillary qubit. With the results from the measurements of the two last mentioned ancillary 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.

4 Conclusions

In this work, we introduced NISQ-HHL, a novel extension of the HHL algorithm that incorporates an eigenvalue inversion component suitable for execution on NISQ devices. We executed it on the trapped-ion Quantinuum System Model H1-1 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. QCL-QPE reduces the number of ancillas to just one for an arbitrary bit precision. It reduces the number of qubits dramatically and it replaces two-qubit gates with one-qubit gates controlled by classical bits. These features are crucial for its execution on near-term hardware.

We experimentally showed the benefits of QCL-QPE over the standard implementation on NISQ devices as it reduces the number of qubits and circuit depth. 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. We showed that this scaling allowed us to effectively estimate the relevant eigenvalues in the QCL-QPE output. These estimates 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 Quantinuum H1-1 system. Moreover, we calculated the inner product between the quantum portfolio state and the classically calculated solution loaded as a quantum state. We obtained a higher inner product using NISQ-HHL than the uniformly controlled rotations method when executing on hardware and in the Qiskit statevector simulator. Finally, we showed that NISQ-HHL can scale to larger portfolios than those supported on current hardware.

Acknowledgments

We thank Tony Uttley, Brian Neyenhuis and the rest of the Quantinuum team for assisting us on the execution of the experiments on the Quantinuum System Model H1-1. We also thank Michele Mosca and his team at University of Waterloo for their insights, and Aram Harrow from Massachusetts Institute of Tech-

nology for his precious feedback.

Disclaimer

References

[1] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. "Quantum Algorithm for Linear Systems of Equations". In: *Physical Review Letters, 103(15):150502* (2009). DOI: 10.1103/PhysRevLett.103.150502.

[2] Patrick Rebentrost and Seth Lloyd. "Quantum computational finance: quantum algorithm for portfolio optimization". In: *arXiv preprint arXiv:1802.08227* (2018).

[3] Scott Aaronson. "Read the fine print". In: *Nature Physics, 11(4):291–293* (2015). DOI: 10.1038/nphys3272.

[4] John Preskill. "Quantum Computing in the NISQ era and beyond". In: *Quantum, 2:79* (2018). DOI: 10.22331/q-2018-08-06-79.

[5] Lidia Ruiz-Perez and Juan Carlos Garcia-Escartin. "Quantum arithmetic with the quantum Fourier transform". In: *Quantum Information Processing, 16(6):152* (2017). DOI: 10.1007/s11128-017-1603-1.

[6] Mikko Mottonen et al. "Transformation of Quantum States Using Uniformly Controlled Rotations". In: *Quantum Info. Comput., 5(6):467–473* (2005). DOI: 10.26421/QIC5.6-5.

[7] Yonghae Lee, Jaewoo Joo, and Soojoon Lee. "Hybrid quantum linear equation algorithm and its experimental test on ibm quantum experience". In: *Scientific reports, 9(1):1–12* (2019). DOI: 10.1038/s41598-019-41324-9.

[8] Jacob Biamonte et al. "Quantum machine learning". In: *Nature* 549.7671 (2017), pp. 195–202. ISSN: 1476-4687. DOI: 10.1038/nature23474.

[9] Marco Pistoia et al. "Quantum Machine Learning for Finance". In: *IEEE/ACM International Conference On Computer Aided Design (ICCAD)* (2021). ICCAD Special Session Paper. DOI: 10.1109/ICCAD51958.2021.9643469.

[10] Thomas Häner, Martin Roetteler, and Krysta M Svore. "Optimizing quantum circuits for arithmetic". In: *arXiv preprint arXiv:1909.05820* (2018).

[11] Mikko Möttönen et al. "Quantum circuits for general multiqubit gates". In: *Physical Review Letters, 93(13):130502* (2004). DOI: 10.1103/PhysRevLett.93.130502.

[12] Hamed Mohammadbagherpoor et al. "Experimental challenges of implementing quantum phase estimation algorithms on ibm quantum computer". In: *arXiv preprint arXiv::1903.07605* (2019).

[13] Robert B Griffiths and Chi-Sheng Niu. "Semiclassical Fourier transform for quantum computation". In: *Physical Review Letters* 76.17 (1996), p. 3228. DOI: 10.1103/PhysRevLett.76.3228.

[14] Stephane Beauregard. "Circuit for Shor's algorithm using 2n+3 qubits". In: *Quantum Info. Comput., 3(2)* (2003). DOI: 10.26421/QIC3.2-8.

[15] Michele Mosca and Artur Ekert. "The hidden subgroup problem and eigenvalue estimation on a quantum computer". In: *In NASA International Conference on Quantum Computing and Quantum Communications, pages 174–188. Springer* (1998). DOI: 10.1007/3-540-49208-9_15.

[16] Miroslav Dobšíček et al. “Arbitrary accuracy iterative quantum phase estimation algorithm using a single ancillary qubit: A two-qubit benchmark”. In: *Physical Review A* 76.3 (2007), p. 030306. DOI: 10.1103/PhysRevA.76.030306.

[17] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010. DOI: 10.1017/CBO9780511976667.

[18] Iordanis Kerenidis and Anupam Prakash. “Quantum Gradient Descent for Linear Systems and Least Squares”. In: *Physical Review A* 101.2 (2020). ISSN: 2469-9934. DOI: 10.1103/physreva.101.022316.

[19] Danial Dervovic et al. “Quantum linear systems algorithms: a primer”. In: *arXiv preprint arXiv:1802.08227* (2018).

[20] Harry Buhrman et al. “Quantum Fingerprinting”. In: *Physical Review Letters* 87.16 (2001). ISSN: 1079-7114. DOI: 10.1103/physrevlett.87.167902.

[21] B. D. Clader, B. C. Jacobs, and C. R. Sprouse. “Preconditioned Quantum Linear System Algorithm”. In: *Physical Review Letters, 110(25):250504* (2013). DOI: 10.1103/PhysRevLett.110.250504.

[22] Changpeng Shao and Hua Xiang. “Quantum circulant preconditioner for a linear system of equations”. In: *Physical Review A, 98(6):062321* (2018). DOI: 10.1103/PhysRevA.98.062321.

[23] Juan M Pino et al. “Demonstration of the trapped-ion quantum CCD computer architecture”. In: *Nature* 592.7853 (2021), pp. 209–213. DOI: 10.1038/s41586-021-03318-4.

[24] Seyon Sivarajah et al. “t|ket⟩: a retargetable compiler for NISQ devices”. In: *Quantum Science and Technology, 6(1):014003* (2020). DOI: 10.1088/2058-9565/ab8e92.

[25] András 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). DOI: 10.1145/3313276.3316366.

[26] Gadi Aleksandrowicz et al. “Qiskit: An open-source framework for quantum computing”. In: *Zenodo* (2019). DOI: 10.5281/zenodo.2562111.

[27] Gilles Brassard et al. “Quantum amplitude amplification and estimation”. In: *Quantum Computation and Information* (2002), pp. 53–74. ISSN: 0271-4132. DOI: 10.1090/conm/305/05215.