Solving Linear Systems on Quantum Hardware with Hybrid HHL^{++}

Romina Yalovetzky[1,*], **Pierre Minssen**[1], **Dylan Herman**[1], **and Marco Pistoia**[1]

[1]Global Technology Applied Research, JPMorganChase, New York, NY 10017, USA
[*]romina.yalovetzky@jpmorgan.com

ABSTRACT

The limited capabilities of current quantum hardware significantly constrain the scale of experimental demonstrations of most quantum algorithmic primitives. This makes it challenging to perform benchmarking of the current hardware using useful quantum algorithms, i.e., application-oriented benchmarking. In particular, the Harrow-Hassidim-Lloyd (HHL) algorithm is a critical quantum linear algebra primitive, but the majority of the components of HHL are far out of the reach of noisy intermediate-scale quantum devices, which has led to the proposal of hybrid classical-quantum variants. The goal of this work is to further bridge the gap between proposed near-term friendly implementations of HHL and the kinds of quantum circuits that can be executed on noisy hardware. Our proposal adds to the existing literature of hybrid quantum algorithms for linear algebra that are more compatible with the current scale of quantum devices. Specifically, we propose two modifications to the Hybrid HHL algorithm proposed by Lee *et al.*[1] leading to our algorithm Hybrid HHL^{++}: (1) propose a novel algorithm for determining a scaling factor for the linear system matrix that maximizes the utility of the amount of ancillary qubits allocated to the phase estimation component of HHL, and (2) introduce a heuristic for compressing the HHL circuit. We demonstrate the efficacy of our work by running our modified Hybrid HHL on Quantinuum System Model H-series trapped-ion quantum computers to solve different problem instances of small-scale portfolio optimization problems, leading to the largest experimental demonstrations of HHL for an application to date.

1 Introduction

Recent advances in quantum hardware[2–4] open the path for experimentation with quantum algorithms. This has led to a rise in *application-oriented* benchmarking of various commercial quantum devices[5,6]. Due to the amount of noise present in the Noisy Intermediate-Scale Quantum (NISQ) hardware[7], it might be necessary to introduce modifications to the quantum algorithms being tested to make them amenable to such devices. We are not currently at the scale where asymptotics play a role, and thus we can swap out some of the components of these algorithms that provide the promised asymptotic speedup for ones that are easier to use for testing the current hardware. These optimizations can be removed once the hardware becomes fault-tolerant. The focus of this work will be on introducing such optimizations to the quantum algorithm for linear systems, HHL, and executing the modified algorithm on a trapped-ion quantum device.

The HHL algorithm was introduced by Harrow, Hassidim and Lloyd[8] for solving a quantum version of the linear systems problem and is a key component in various algorithms and applications. In particular, HHL has been proposed as a possible solver for a specific portfolio-management problem[9], known as mean-variance portfolio optimization[10]. However, HHL is known to be cumbersome to deploy[11], especially on NISQ hardware. Recently, hybrid classical-quantum versions of quantum linear systems solvers, such as variational approaches[12,13] and a hybrid version of HHL[1,14] have been developed to enable near-term experimentation with quantum linear algebra techniques.

The Hybrid HHL, originally developed by Lee *et al.*[1], was focused solely on reducing the complexity of the eigenvalue inversion step, at the cost of producing an algorithm that does not have an asymptotic speedup over classical approaches. This is an example of the types of modifications mentioned earlier. This approach avoids, currently, infeasible quantum arithmetic and exponentially deep uniformly controlled rotation gates[15]. However, we believe that further optimizations of the Hybrid HHL are still possible, especially given newly available quantum hardware features. The culmination of the techniques we introduce will be the Hybrid HHL^{++}, which we believe provides a viable near-term application-oriented benchmark, which we show by applying it to small-scale portfolio optimization problems.

To highlight, our work makes the following novel contributions:

1. Introduction of the Hybrid HHL^{++} algorithm, which incorporates two new features into the exisiting Hybrid HHL pipeline:

(a) an efficient and verifiable algorithmic procedure for determining a factor to scale the system matrix by, which allows for resolving the eigenvalues with significantly higher accuracy and making better use of the allocated number of qubits,

(b) and a generic procedure for reducing the complexity of Hybrid HHL circuit, in terms of both the number of qubits and the number of rotations used by the eigenvalue inversion circuit. This protocol makes use of the semiclassical quantum phase estimation[16] and leverages newly available hardware features, such as conditional logic and mid-circuit measurements.

2. To the best of our knowledge, we demonstrate the largest-to-date execution[1,6,14,17] of HHL with a two-qubit gate depth up to 291 on real quantum hardware—the Quantinuum H-series trapped-ion devices—to solve portfolio-optimization problems with S&P 500 assets.

Paper Organization The remainder of this paper is organized as follows. Section 2 reviews both HHL and Hybrid HHL, Section 3 introduces the novel techniques for enhancing Hybrid HHL culminating in the Hybrid HHL^{++} algorithm. Section 4 presents the results of our experiments from simulation and the Quantinuum H-series devices and noisy emulators. Section 5 presents an analysis of future hardware demonstrations by simulating different levels of noise on the quantum hardware and increasing the problem size. Finally, Section 6 summarizes the results of this work, and concludes the article.

2 Review of HHL and Hybrid HHL

The original HHL algorithm was proposed to solve the following quantum analog of a linear system. Given oracle access to the entries of an $2^n 2^n$ Hermitian matrix $\mathbf{A}$ and an n-qubit quantum state $|\mathbf{b}\rangle$ prepare the state,

$$|\mathbf{x}\rangle = \frac{\mathbf{A}^+ |\mathbf{b}\rangle}{\|\mathbf{A}^+ |\mathbf{b}\rangle\|_2}, \tag{1}$$

where $\mathbf{A}^+$ is the Moore-Penrose pseudoinverse of $\mathbf{A}$. The restrictions on $\mathbf{A}$ being square and Hermitian can be relaxed by dilation. The HHL procedure to prepare $|\mathbf{x}\rangle$ is roughly as follows:

- **Step 1:** Perform m-qubit quantum phase estimation (QPE) to the unitary $e^{2\pi i\gamma\mathbf{A}}$ (obtained via a Hamiltonian simulation algorithm) and apply it to the state $|\mathbf{b}\rangle$ to obtain:

$$\sum_\lambda c_\lambda |\psi_\lambda\rangle |2^m\gamma\lambda\rangle, \tag{2}$$

where $|\psi_\lambda\rangle$ is an eigenstate of $\mathbf{A}$ with eigenvalue λ, $c_\lambda = \langle\mathbf{b}|\psi_\lambda\rangle$. For simplicity, we assume that there is a known γ s.t. $\forall\lambda$ s.t. $|c_\lambda| > 0$:

$$2^m\gamma\lambda \in \{-2^{m-1},\ldots,0,\ldots,2^{m-1}-1\} =: \mathscr{B}_m, \tag{3}$$

represented using two's complement (i.e. that phase estimation can be performed exactly).

- **Step 2:** Perform the following unitary (called *eigenvalue inversion*) for $\lambda \neq 0$:

$$|0\rangle|2^m\gamma\lambda\rangle \mapsto \frac{C}{\lambda}|0\rangle|2^m\gamma\lambda\rangle + \sqrt{1-\frac{C^2}{\lambda^2}}|1\rangle|2^m\gamma\lambda\rangle, \tag{4}$$

where C is chosen so that $|C/\lambda| \leq 1$ for all λ.

- **Step 3:** Uncomputing and post-selecting on $|0\rangle$ gives

$$\frac{1}{\mathcal{N}}\sum_\lambda \frac{c_\lambda}{\lambda}|\psi_\lambda\rangle = |\mathbf{x}\rangle. \tag{5}$$

There can be an exponential quantum speedup in terms of the number of queries to the oracles for $\mathbf{A}$ and $|\mathbf{b}\rangle$ when $\mathbf{A}$ is sufficiently sparse and well-conditioned. However, this algorithm is still not feasible on near-term devices. Our work is focused on finding more near-term friendly versions of the original HHL that, at the expense of losing the HHL speedup, can be implemented on current hardware and used as an application-oriented benchmark. The algorithm we will be focused on improving, the Hybrid HHL, is one such approach to making HHL compatible with small quantum devices. In addition, due to

Hamiltonian simulation being generally infeasible on current devices for an arbitrary Hermitian $\mathbf{A}$ and 2^n being small enough that a classical computer can solve the problem, we simply fallback on the following hybrid simulation procedure: compute the unitary matrix $\mathbf{U} = e^{2\pi i\gamma \mathbf{A}}$ on a classical device and use standard quantum circuit packages, e.g., Qiskit[18], to decompose $\mathbf{U}$ into basis gates. Specifically, Qiskit makes use of product formulas to approximate the simulation.

The key step that distinguishes Hybrid HHL, and removes any potential speedup (besides the hybrid Hamiltonian simulation procedure), is that we obtain classical access to the eigenvalues of $\mathbf{A}$ (via either sampling from the QPE output distribution or a classical eigensolver). This allows us to implement the eigenvalue inversion (step 2 in the procedure above) by applying a single multi-controlled rotation for each eigenvalue λ. While utilizing QPE instead of a classical eigensolver to obtain the eigenvalues provides no computational benefit in the current setting, Hybrid HHL and our approach opt to utilize QPE instead so as to simulate the HHL algorithm more closely. The benefits of estimating the eigenvalues a priori in HHL for demonstrations on small hardware are two-fold: (1) it removes the need to use resource-heavy quantum arithmetic and (2) the number of controlled rotations in the eigenvalue inversion circuit only scales with the number of distinct eigenvalues estimated. One alternative to the second point is known as the uniformly controlled rotation gate, which performs a multi-controlled rotation for every basis state that $|2^m\gamma\lambda\rangle$ could be, exponentially scaling with the number of ancillary qubits.

The following is a summary of the two main components of Hybrid HHL:

1. Eigenvalue Estimation

   (a) Perform step 1 of the standard HHL algorithm using the hybrid Hamiltonian simulation procedure described earlier.

   (b) Measure the register containing $|2^m\gamma\lambda\rangle$ to obtain a probability distribution of the eigenvalue estimates such that $|c_\lambda| > 0$ or is above some threshold value.

2. Compute $|\mathbf{x}\rangle$

   (a) Perform step 1 of the standard HHL again on a fresh copy of $|\mathbf{b}\rangle$.

   (b) For each observed λ in step 1, apply the following multi-controlled rotation $\mathsf{R_Y}$:

$$|2^m\gamma\lambda\rangle\langle 2^m\gamma\lambda| \otimes \mathsf{R_Y}(2\sin^{-1}(C/\lambda)) + [\mathbb{I} - |2^m\gamma\lambda\rangle\langle 2^m\gamma\lambda|] \otimes \mathbb{I}. \tag{6}$$

   (c) Uncompute and post select on $|0\rangle$ as in standard HHL.

Lee *et al.* also proposed an additional optimization that allows the number of controlled rotations in the eigenvalue inversion to be reduced further under additional assumptions. We will address this in a later section and present a potential improvement. While the hybrid version of HHL is significantly more near-term compatible than the original version, there are still additional optimizations that can be made. As mentioned in the introduction, our work presents two additional modifications, which we consolidate into the Hybrid HHL^{++} algorithm.

3 Hybrid HHL^{++}

In this section we present our improved Hybrid HHL procedure, which enables larger-scale experimental demonstrations of quantum linear algebra than prior work. As mentioned earlier, Hybrid HHL^{++} differs from the proposal of Lee *et al.* in two ways: the spectral scaling algorithm (Section 3.1) and the heuristic for compressing the HHL circuit (Section 3.2). The following two subsections will describe these modifications in detail. Figure 1 shows the end-to-end flow combining all of the techniques introduced.

3.1 Spectral Scaling Algorithm for Optimizing γ

As hinted at in the presentation of the HHL and Hybrid HHL algorithms in the previous section, the choice of γ, the parameter to scale $\mathbf{A}$ by, is critical to the performance of phase estimation. Given that the eigenvalues of $\mathbf{A}$ could be irrational or rational numbers with incompatible denominators, there may be no such that γ that allows for a perfect phase estimation to m-bits, i.e., an element of $\mathscr{B}_m$. In addition, given that we are working on small quantum devices, it is important to find a γ that makes the best use of the allocated m bits of precision.

The notion of making "best use" of the m bits of precision can actually be made somewhat precise. Specifically, the eigenvalues with the largest magnitude should be associated with the elements with largest magnitude of $\mathscr{B}_m$. If we know that the main precondition for QPE to work $\|\mathbf{A}\|_2 \leq 1$ is satisfied, and blindly apply phase estimation, then the desired relation between the eigenvalues and elements of $\mathscr{B}_m$ just mentioned may not be satisfied. This can lead to both eigenvalues not being

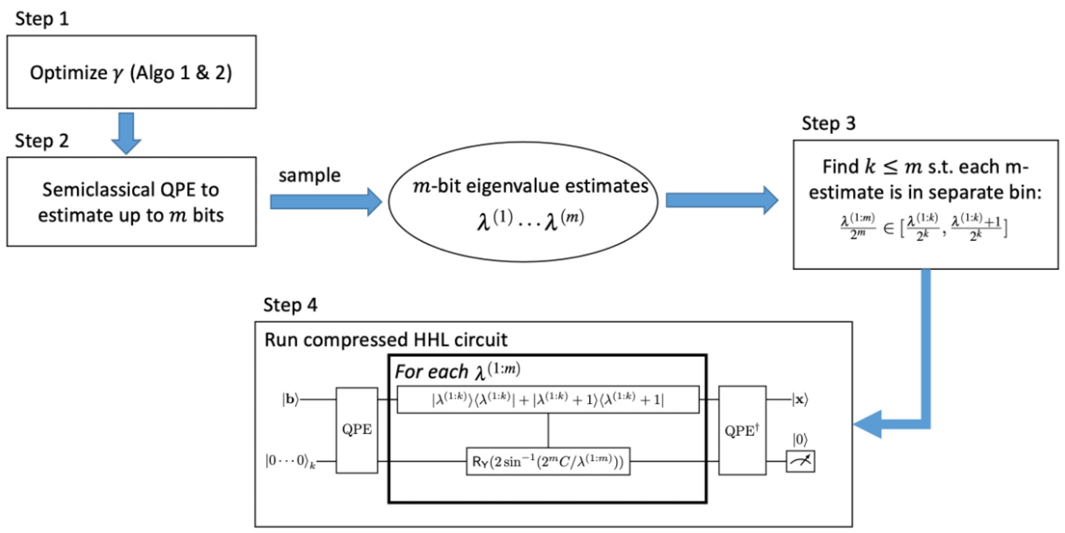

Step 1
Optimize γ (Algo 1 & 2)
Step 2
Semiclassical QPE to estimate up to m bits
sample
m-bit eigenvalue estimates $\lambda^{(1)} \dots \lambda^{(m)}$
Step 3
Find $k \leq m$ s.t. each m-estimate is in separate bin: $\frac{\lambda^{(1:m)}}{2^m} \in [\frac{\lambda^{(1:k)}}{2^k}, \frac{\lambda^{(1:k)}+1}{2^k}]$
Step 4
Run compressed HHL circuit
For each $\lambda^{(1:m)}$
$|\mathbf{b}\rangle$
$|0\cdots0\rangle_k$
QPE
$|\lambda^{(1:k)}\rangle\langle\lambda^{(1:k)}| + |\lambda^{(1:k)}+1\rangle\langle\lambda^{(1:k)}+1|$
$R_Y(2\sin^{-1}(2^m C/\lambda^{(1:m)}))$
$QPE^\dagger$
$|\mathbf{x}\rangle$
$|0\rangle$

Figure 1. End-to-End Flow of Hybrid HHL^{++}: highlights the integration of the two proposed modifications, the spectral scaling algorithm (Section 3.1) in Step 1 and the HHL circuit compression (Section 3.2) techniques in Step 3 and Step 4, into the original Hybrid HHL flow.

able to be properly resolved and a waste allocated resources. Ideally, we would like to find a γ that stretches the spectrum as to take up as much of $\mathscr{B}_m$ as possible. The goal of this section is to present an efficient and simple procedure for finding such a γ, which we call the spectral scaling algorithm. Lastly, we would like to emphasize that γ only needs to be chosen to ensure high-quality estimation of those λ s.t. $|c_\lambda| > 0$. This is implicitly accounted for by sampling from the output distribution of QPE with $\mathbf{U}$ applied to the initial state $|\mathbf{b}\rangle$.

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

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

The spectral scaling procedure is highlighted in Algorithm 1, which shows how to optimize the selection of γ. Given m-bits, the optimal value of γ returned by the algorithm helps to encode the eigenvalues using all available bits in the output distribution of phase estimation. As a result, this makes it easier to distinguish the eigenvalues from each other and estimate the relevant eigenvalues, 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 $\lambda_{\max}$. 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 QPE to compute the m-bit estimates of the eigenvalues of the unitary $\mathbf{U}$ using gamma computed during the previous iteration. Specifically, Algorithm 1 post-processes the output distribution to get a new m-bit estimation of $2^m\gamma\lambda_{\max}$ in order to update γ for the next iteration.

The number of iterations to find the optimal γ using Algorithm 1 is in $\Theta\left({}^1\!/_m \log_2\left({}^\alpha\!/_{\lambda_{\max}}\right)\right)$. 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 ${}^{(2^m-1)}\!/_x$. In practice, we also want to underestimate γ in order to prevent amplitudes of basis states near 2^m 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 $m \geq \log_2({}^{\lambda_{\max}}\!/_{\lambda_{\min}}) = \log_2(\kappa)$. In practice, we first run Algorithm 1 with an initial precision of m bits. Then, if $\lambda_{\min}$, estimated via 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 $\mathbf{A}$ is assumed invertible and consequently, no eigenvalue can be 0. Thus, we increase m just enough to guarantee that the estimation of $\lambda_{\min}$ is different from 0, at which point $m \geq \log_2(\kappa)$.

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 $\|\mathbf{A}\|_F := \sqrt{\mathrm{Tr}(\mathbf{A}^\dagger\mathbf{A})}$. Indeed, since $\mathbf{A}$ is Hermitian, $||\mathbf{A}||_F = \sqrt{\sum_{i=0}^{N-1}\lambda_i^2}$, and so the Frobenius norm of A is a valid overapproximation of $\lambda_{\max}$. However, to make the procedure general, we incorporate a subroutine the finding the overestimate. For this reason, we propose Algorithm 2, which, using one execution of QPE, tests the validity of any initial guess of α. 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.

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

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

```
// Initialize scaling parameter
$\Gamma := {}^1\!/_{2^{m+1}\alpha}$
$p :=$ $m$-bit output distribution of QPE using unitary $e^{2\pi i\Gamma\mathbf{A}}$ 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 $\lambda_{\max}$
```

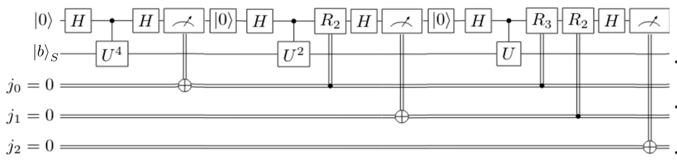
The idea of Algorithm 2 is the following. If α is a valid guess, there will be no overflow in the output distribution of QPE applied to $e^{2\pi i\gamma\mathbf{A}}$ with $\gamma = {}^1\!/_\alpha$. Then, performing $m+1$ right bit shifts should reduce all n-bit eigenvalue estimates to 0. To test this,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

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^m by 2^{m-1} in the update of γ. Similarly, in Algorithm 2, replacing ${}^1\!/_{2^{m+1}}$ by ${}^1\!/_{2^m}$ in the definition of Γ suffices to support negative eigenvalues. Accounting for negative eigenvalues is crucial, since $\mathbf{A}$, in Equation (14), may be indefinite.

3.2 Heuristic for Compressing HHL circuit

In the preliminary eigenvalue estimation step of Hybrid HHL, one could utilize the aforementioned approach of Lee *et al.* , where the textbook implementation of phase estimation is used. However, QPE is still difficult to implement on NISQ hardware[19] because the number of ancillary qubits grows with the desired bit precision. Additionally, an m-qubit quantum Fourier transform (QFT) involves at least one controlled rotation between any two qubits, and $\binom{m}{2}$ in total. Therefore, a quantum device with all-to-all connectivity is preferred to limit the use of SWAP gates.

A phase estimation variant[20,21] has been identified that better lends itself to NISQ hardware and utilizes the semiclassical inverse QFT[16]. We will refer to this variant as semiclassical QPE. This is a nonunitary version of the inverse QFT that estimates each bit of the eigenvalue sequentially. A diagram of this variant is displayed in Figure 2a, which shows how to efficiently estimate the eigenvalues of the unitary operator $\mathbf{U}$ to three bits of precision. This variant leverages techniques for dynamic quantum circuits made available on recent hardware: mid-circuit measurements, ground-state resets, conditional logic, and


(a) Semiclassical QPE circuit

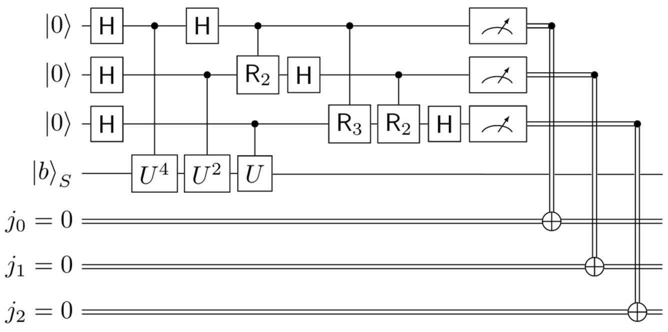

(b) Standard QPE circuit

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

Algorithm 3: Semiclassical Phase Estimation[20, 21]

```
Input: a bit precision $m$, unitary operator $U$, and input quantum state $|b\rangle$
Step 1: Initialize one only ancilla qubit at $|0\rangle$, a register $S$ to the $|b\rangle$ state, and $m$ classical registers $\{j_0, j_1, \ldots, j_{m-1}\}$ to
  store the results measured in the ancilla qubit.
Step 1: Apply Hadamard H to the ancillary qubit
Step 2: Apply the unitary $\mathbf{U}^{2^{m-1}}$ on the $S$ register conditioned on the ancilla qubit.
Step 3: Apply Hadamard H to the ancillary qubit.
Step 4: Measure the ancilla qubit and store the results on the classical register $j_0$.
for $i = 2 \cdots m$ do
    Step 5.1: Apply Hadamard H to the ancillary qubit
    Step 5.2: Apply the unitary $\mathbf{U}^{2^{m-i}}$ on the $S$ register conditioned on the ancilla qubit.
    for $k = i \cdots 2$ do
        Step 6.1: Apply the phase gate $R_k$ on the ancilla qubits conditioned on the classical register $j_{i-k}$
    end
    Step 7: Apply Hadamard H to the ancillary qubit
    Step 8: Measure the ancilla qubit and store the results on the classical register $j_{m-i}$
end
```

qubit reuse. This semiclassical variant of QPE has been previously demonstrated on superconducting quantum hardware[22]. However, to the best of our knowledge, our work is the first one benchmarking its performance on trapped-ion devices, and the first to incorporate it into a quantum algorithm executed end-to-end on commercially available hardware for solving a practical problem.

The semiclassical QPE is mathematically equivalent to performing the original inverse QFT and measuring the eigenvalue register. It is also similar to iterative QPE (iQPE)[23] 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 $\mathbf{U}$ in order to estimate its corresponding eigenvalue. Conversely, semiclassical QPE can estimate eigenvalues without prior information of the eigenvectors. The description of how to implement the semiclassical QPE is shown below in Algorithm 3.

The semiclassical QPE is used to perform the eigenvalue estimation in Hybrid HHL^{++}, and it has two properties that make it more suitable for near-term devices than the standard version of phase estimation:

1. the procedure requires only one ancillary qubit for an arbitrary bit precision,

2. and it replaces two-qubit gates with one-qubit gates controlled by classical registers.

These differences can be observed by comparing Figures 2a and 2b.

Our work proposes to modify the eigenvalue estimation step (step 1) of Hybrid HHL to use the semiclassical QPE. However, in step 2, we still utilize the textbook implementation of QPE prior to the eigenvalue inversion step. The benefits of this will be highlighted shortly, where we discuss our heuristic for reducing the depth of the eigenvalue inversion circuit. In the *Results* section, we will report data from extensive benchmarking of the semiclassical QPE on a trapped-ion device.

Note that the semiclassical phase estimation cannot be used for the eigenvalue inversion since it repeatedly collapses the quantum state of the ancillary qubit through mid-circuit measurements. Therefore, this eigenvalue estimation procedure cannot be incorporated as part of a deeper circuit that relies on the quantum states encoded in this ancillary register. Thus, we specifically utilize the semiclassical procedure for solely estimating the rotation angles, which can be done to arbitrary precision with a single ancillary qubit.

Recall that the eigenvalue inversion for Hybrid HHL consists of applying the following controlled rotation for each estimated eigenvalue λ:

$$|2^m\gamma\lambda\rangle\langle 2^m\gamma\lambda| \otimes \mathsf{R_Y}(2\sin^{-1}(C/\lambda)) + [\mathbb{I} - |2^m\gamma\lambda\rangle\langle 2^m\gamma\lambda|] \otimes \mathbb{I}. \tag{7}$$

However, as observed by Lee *et al.* , the purpose of QPE in the eigenvalue estimation and inversion steps is quite different. Specifically, in step (1) of Hybrid HHL the goal is to estimate the rotation angles $2\sin^{-1}(C/\lambda)$. This is where the most precision is required. In step (2), the goal is to just be able to just find a bijection between distinct eigenvalue estimates in step (1) and the values the control register of the rotation can take. More specifically, we just want

$$|f(\lambda)\rangle\langle f(\lambda)| \otimes \mathsf{R_Y}(2\sin^{-1}(C/\lambda)) + [\mathbb{I} - |f(\lambda)\rangle\langle f(\lambda)|] \otimes \mathbb{I}, \tag{8}$$

where f is bijection from the set of eigenvalues with $|c_\lambda| > 0$ and k-bit integers, *where k may be less than m*. This means in some cases, we can actually run a QPE that uses fewer qubits in step (2) of Hybrid HHL than step (1). The approach of Ref.[1] utilized a concept called eigenmeans. This technique can result in $k < m$ when the bit representations of $2^m\gamma\lambda$ agree on the first k bits but all differ on the last $m-k$ bits. Unfortunately, this appears to be a strong assumption and the authors also showed it working for a single example. In addition, an important point to emphasize is that the eigenmean approach does not allow for reducing the number of ancillary qubits used by the phase estimation circuit. The use of eigenmeans only allows for reducing the number of controlled rotations and potentially the qubits they control on, not the size of the QPE circuit.

In this section, we present an alternative to the eigenmeans approach that is more generic and allows for reducing the complexity of the overall HHL circuit, i.e., an eigenvalue inversion circuit sandwiched between two standard QPEs (see Figure 1). The first step of our procedure is to utilize the spectral scaling algorithm of Section 3.1 (Algorithm 1 and 2) to determine a scaling factor and a number of qubits m that is sufficient to resolve the relevant eigenvalues of $\mathbf{A}$. The observed eigenvalues will be used to compute the rotation angles for Equation (8). As mentioned earlier, the number of ancillary qubits used by the semiclassical QPE does not increase with the desired precision. Thus, we will make the following simplifying assumption that there exists a $\gamma \in \mathbb{R}$ and integer m such that for all eigenvalues λ of $\mathbf{A}$: $2^m\gamma\lambda \in \mathscr{B}_m$. We will denotes the m-bits of $2^m\gamma\lambda$ by $\lambda^{(1)}\cdots\lambda^{(m)} = \lambda^{(1:m)}$. We will also assume that $2^m\gamma\lambda \neq 2^{m-1}-1$.

Suppose that instead we use an k-bit QPE for $k < m$, which performs the following unitary[24]:

$$\sum_\lambda c_\lambda |\psi_\lambda\rangle |\mathbf{0}_m\rangle \mapsto \sum_\lambda c_\lambda |\psi_\lambda\rangle \otimes \sum_{\ell=0}^{2^m-1} 2^{-k} \frac{1-e^{2\pi i 2^k(2^{-m}\lambda^{(k+1:m)}-2^{-k}\ell)}}{1-e^{2\pi i(2^{-m}\lambda^{(k+1:m)}-2^{-k}\ell)}} |\lambda^{(1:k)}+\ell \mod 2^k\rangle, \tag{9}$$

where by assumption, $\frac{\lambda^{(1:m)}}{2^m} \in [\frac{\lambda^{(1:k)}}{2^k}, \frac{\lambda^{(1:k)}+1}{2^k}]$. Specifically, we want the smallest k such that only one $\lambda^{(1:m)}$ lies in each of these intervals. This allows us to perfectly distinguish the eigenvalues of $\mathbf{A}$. It is well known[25] that the probability mass that phase estimation assigns to this region is,

$$2^{-k}\frac{1-e^{2\pi i 2^k(2^{-m}\lambda^{(k+1:m)})}}{1-e^{2\pi i(2^{-m}\lambda^{(k+1:m)})}} + 2^{-k}\frac{1-e^{2\pi i 2^k(2^{-m}\lambda^{(k+1:m)}-2^{-k})}}{1-e^{2\pi i(2^{-m}\lambda^{(k+1:m)}-2^{-k})}} \tag{10}$$

$$\geq \frac{8}{\pi^2} \approx 0.81. \tag{11}$$

Thus if we were perform the following controlled rotation:

$$(|\lambda^{(1:k)}\rangle\langle\lambda^{(1:k)}| + |\lambda^{(1:k)}+1\rangle\langle\lambda^{(1:k)}+1|) \otimes \mathsf{R_Y}(2\sin^{-1}(2^m C/\lambda^{(1:m)})) + |\perp\rangle\langle\perp| \otimes \mathbb{I}, \tag{12}$$

and then invert QPE, we would obtain a state with at most a trace distance of 0.19 from the state obtained by using an m-qubit (exact) QPE. Note that $|\perp\rangle\langle\perp|$ corresponds to the projector onto the complement of $|\lambda^{(1:k)}\rangle\langle\lambda^{(1:k)}| + |\lambda^{(1:k)}+1\rangle\langle\lambda^{(1:k)}+1|$. In addition, the trace distance can be reduced to ε by increasing the QPE precision by $\lceil \log(2+\frac{2}{\varepsilon}) \rceil$ additional qubits[24].

If there are two $\lambda^{(1:m)}$ that fall into adjacent intervals, then we can either increase k to cause them to be further separated, or choose the angle of rotation controlled on their common endpoint to be an average of the reciprocals. In addition, it may be that we only need to control on $\lambda^{(1:k)}$, depending on where in the interval $\lambda^{(1:m)}$ falls. However, such a case could be an indication that we can further reduce the precision k.

One may notice that this procedure appears to work in a regime that is opposite to where the eigenmeans procedure introduced by Lee *et al.* worked. Specifically, the above procedure saves when the higher-order bits of the eigenvalues differ, whereas, the eigenmeans procedure saves when the lower-order bits differ. Thus if $\exists\, t \leq m$ s.t. $\lambda^{(1:t)}$ agree for all λ, i.e., the case of fixed eigenmeans coined by Ref.[1], then instead of only controlling on at most 2^{m-t} values from an m-qubit register, it actually suffices to perform QPE on $2^t\gamma\mathbf{A}$ with only $m-t$ qubits. The periodicity of the complex exponential with cause phase estimation to ignore the most significant bits. This reconciles the difference between our approach and the eigenmeans, and shows that our proposal works generically.

We summarize the overall procedure below.

1. Using semiclassical QPE, optimize the scaling factor γ using the algorithms from the previous subsection. Determine a number of bits m that is sufficient to resolve eigenvalues of $\gamma\mathbf{A}$ to high precision. These are the m-bit estimations $\lambda^{(1:m)}$. We can also estimate the probability of the estimates and only accept ones above a certain threshold (for exact estimation this corresponds to thresholding on $|c_\lambda|$). This is done in Step 1 and 2 from Figure 1.

2. Reduce the precision of QPE starting from m to obtain the smallest $k \leq m$ s.t. $\frac{\lambda^{(1:m)}}{2^m} \in [\frac{\lambda^{(1:k)}}{2^k}, \frac{\lambda^{(1:k)}+1}{2^k}]$ is satisfied (i.e., no two m-bit estimates lie between two k-bit estimates). Potentially a different γ can be chosen as well. If all $\lambda^{(1:m)}$ agree on the first t-bits, multiply the matrix by 2^t. If the intervals for two $\lambda^{(1:m)}$ share an endpoint, then we can assign the rotation angle to be the average of the reciprocals of the $\lambda^{(1:m)}$. This is done in Step 3 from Figure 1.

3. Construct the HHL circuit by performing a k-bit QPE. Then for each $\lambda^{(1:m)}$ apply the rotation:

$$(|\lambda^{(1:k)}\rangle\langle\lambda^{(1:k)}| + |\lambda^{(1:k)}+1\rangle\langle\lambda^{(1:k)}+1|) \otimes \mathsf{R_Y}(2\sin^{-1}(2^m C/\lambda^{(1:m)})) + |\perp\rangle\langle\perp| \otimes \mathbb{I}, \tag{13}$$

then invert the k-bit QPE. Depending on where the eigenvalue falls in the interval, controlling on $\lambda^{(1:k)}$ only may suffice. As mentioned earlier, this is an indication that we can potentially reduce the precision to cause $\lambda^{(1:m)}$ to fall between to k-bit estimates. This is done in Step 4 from Figure 1.

The benefit of this procedure is that when running HHL, both the size of the standard QPE and the eigenvalue inversion circuit can be significantly reduced. This saves on both depth and ancillary qubits. The use of the semiclassical QPE allows for high-precision estimates with only one ancillary qubit. Thus, the whole procedure can be substantially more near-term friendly. However, we emphasize that this approach is more of a heuristic, yet, we believe it is well motivated by the discussion in this subsection. The main message is that once we have classical access to the m-bit eigenvalue estimates, the only goal of the phase estimation in the HHL circuit is to produce a mapping that separates distinct eigenvalues.

4 Results

In this section, we detail the outcomes from the experiments with Hybrid HHL^{++} on the Quantinuum H-series devices. We also include results from a noisy emulator of the machine provided by Quantinuum and noiseless simulation. These demonstrations are at a larger scale than prior hybridizations of HHL and showcase both the benefits of our techniques and the advancement of quantum hardware. We believe that this implies that Hybrid HHL^{++} is a good application-oriented benchmark for small hardware.

Our application of choice is portfolio optimization formulated as a linear system, hence the emphasis of an application-oriented benchmark. This is based on the formulation of Rebentrost and Lloyd, which we detail below.

4.1 Portfolio Optimization as a Linear System

HHL has been proposed as a possible solver for a specific portfolio-management problem[9], 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* $\Sigma \in \mathbb{R}^{\tilde{N}\tilde{N}}$, the *expected returns* $\mathbf{r} \in \mathbb{R}^{\tilde{N}}$, and the *prices* $\mathbf{p} \in \mathbb{R}^{\tilde{N}}$ of the assets. Its objective is to minimize the *risk*, represented by the quadratic form $\mathbf{w}^\mathsf{T}\Sigma\mathbf{w}$, subject to the desired *expected total return* $\mu \in \mathbb{R}$ and *budget* $\xi \in \mathbb{R}$. The solution $\mathbf{w} \in \mathbb{R}^{\tilde{N}}$ is the *allocation vector* that weighs each asset in the portfolio. 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.

This problem can be stated as a convex quadratic program:

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

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

$$\begin{bmatrix} 0 & 0 & \vec{r}^{\mathsf{T}} \\ 0 & 0 & \mathbf{p}^{\mathsf{T}} \\ \mathbf{r} & \mathbf{p} & \Sigma \end{bmatrix} \begin{bmatrix} \eta \\ \theta \\ \mathbf{w} \end{bmatrix} = \begin{bmatrix} \mu \\ \xi \\ \mathbf{0} \end{bmatrix} \tag{14}$$

where $\eta,\theta\in\mathbb{R}$ are the Lagrange multipliers. We will denote this linear system by $A\mathbf{x}=\vec{b}$, with $A\in\mathbb{R}^{NN}$ and $\mathbf{x},\mathbf{b}\in\mathbb{R}^{N}$, 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 $|\mathbf{x}\rangle=|\eta,\theta,\mathbf{w}\rangle$ allows us to recover $|\mathbf{w}\rangle$.

Following this approach, Rebentrost and Lloyd[9] 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[26], 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.

4.2 Experiments on a Trapped-ion Device

In order to run the circuits on Quantinuum hardware, we transpiled and optimized the circuits from IBM's Qiskit[18] to the H-series devices' native gates using Quantinuum's pytket package[27]. As the error rates of two-qubit gates are an order of magnitude larger than those of one-qubit gates[28], and the numbers of both gate types are similar in the circuits used, we will only present the number of ZZPhase gates, the H-series native two-qubit gate, for the circuits. The ZZPhase is equivalent to $\mathrm{R}_{ZZ}(\theta)$, and, up to one-qubit gates, it is realized via the Mølmer-Sørensen interaction[28]. The error is related to the magnitude of the angle of rotation θ.

The data for the linear system matrix **A** comes from S&P 500 stock data. Specifically, we utilized 2 assets, leading to a 44 matrix. The matrix is constructed using the procedure for formulating constrained portfolio optimization as a linear system that was highlighted in the previous subsection. We created six different problem instances and show results using these instances in noiseless simulation using the Qiskit QASM simulator and on the Quantinuum trapped-ion hardware and noisy emulators. We took advantage of the emulator by executing the circuits with different numbers of shots to identify the minimum such that there are no significant changes in the results. Note that typically trapped-ion devices require significantly more time than superconducting devices to run perform a single shot. For reference, for the Quantinuum H-series devices, each shot of a QAOA with $p=2$ and 32 qubits takes 1 second[29]. Thus, we also need to make a compromise on the number of shots to ensure that the experiments can be executed within a reasonable time frame.

The experiments are split into three categories resulting in different subsections. First, we show how the spectral scaling algorithm, introduced in Section 3.1, works to determine the optimal γ named γ_{op}. Then, with this optimal value of the scaling parameter, we perform a comparative analysis between the semiclassical and textbook phase estimation procedures on hardware in Section 4.2.2. Lastly, we utilize the controlled-SWAP test on hardware to check the quality of the portfolio proposed by the algorithm in Section 4.2.3.

4.2.1 Spectral Scaling Algorithm Benchmarks

In this section we present results obtained on hardware that show how Algorithm 1 and Algorithm 2 optimize the scaling parameter γ, allowing for effectively estimating the relevant eigenvalues in the output distribution of quantum phase estimation. These proposed novel algorithms allow for reducing the resources needed to increase the bit precision to accurately identify the relevant eigenvalues. The optimized scaling parameter γ_{op} makes the best use of the bit precision available. In this work, we leverage the semiclassical QPE, which, as explained before, lends itself to near-term hardware. What is more, the estimation of the eigenvalues with lower bit precision, thanks to using an optimal scaling parameter, also allows to reduce the number of qubits in the HHL circuit.

To show how the optimization of γ works, we select one of the problem instances corresponding to the optimization of a portfolio of two assets (44 system) and we executed the semiclassical QPE one of the Quantinuum machines. 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 semiclassical QPE with $\Gamma = \gamma \cdot 2^{-4} = 50 \cdot 2^{-4}$. The output probability distribution is shown in Figure 3 (a), together with the distribution obtained in noiseless simulation ("qasm_simulator") and the theoretical values of the eigenvalues, calculated analytically ("theoretical value"). We observe that the outputs from both the quantum device and noiseless simulation are concentrated around zero, thus $\gamma = 50$ is a valid guess. An example for an invalid input would be $\gamma = 3200$. The output distribution of semiclassical QPE with $\Gamma = \gamma \cdot 2^{-4} = 200$ is plotted on Figure 3 (b). As this distribution is not concentrated around zero, $\gamma = 1600$ is not a valid guess.

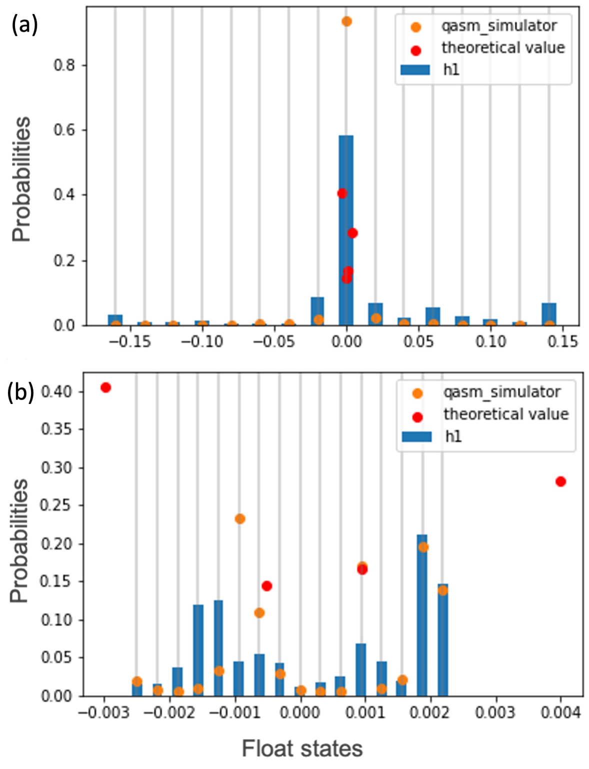


Figure 3. Probability distributions over the four-bit eigenvalue estimates from the semiclassical QPE executions using $e^{i\mathbf{A}2\pi\Gamma}$ for $\Gamma = 50 \cdot 2^{-4}$ (a) and $\Gamma = 200$ (b). The blue bars represent the experimental results on the trapped-ion device with 2000 shots. The theoretical (classically calculated) eigenvalues are represented with the red dots and the results from the noiseless simulator, Qiskit QASM simulator, are represented with the orange dots. As shown on (b), as the theoretical eigenvalues exceed the values we can encode, the distribution we observed shows that values overflowed.

Now that we confirmed $\gamma = 50$ is a valid guess, we can execute Algorithm 1. We ran semiclassical QPE for estimating the eigenvalues of $\mathbf{A}$ with this value of γ, and we used $|\mathbf{b}\rangle$ as the initial state. The output probability distribution obtained both on hardware and in noiseless simulation are displayed in Figure 4 (a), together with the theoretical values. 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 spectral scaling parameter 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 4 (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 scaling factr optimization procedure. For clarification, the distribution of classically calculated, theoretical, values in the plots are distributed according to $\{|c_{\lambda_i}|^2\}_{i=0}^{N-1}$, using the notation introduced in Section 2.

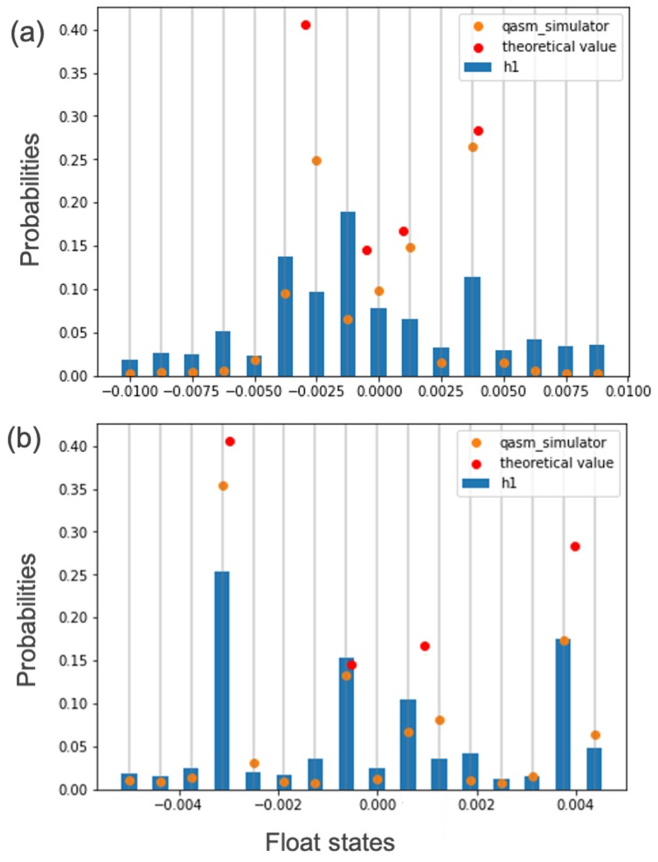


Figure 4. Probability distributions over the four-bit eigenvalue estimates from the semiclassical QPE run using $e^{iA2\pi\gamma}$ for $\gamma = 50$ (a) and $\gamma = 100$.The blue bars represent the experimental results on the trapped-ion device - 2000 shots (a), 1000 shots (b). The theoretical (classically calculated) eigenvalues are represented with the red dots and the results from noiseless simulation with the Qiskit QASM simulator are represented with the orange dots.

4.2.2 Semiclassical and Standard Quantum Phase Estimation Benchmarks

The aim of this subsection is to present the results from benchmarking the semiclassical quantum phase estimation to estimate the eigenvalues on hardware. First, we compare its performance to estimate the eigenvalues up to different bit precisions. For the same problem instance we found the optimal γ_{op} by analyzing results on hardware in previous subsection, we executed the semiclassical QPE to the unitary with this optimal γ_{op} on the hardware to estimate the eigenvalues of $\mathbf{A}$ to different precisions: three, four and five. We compare the number of gates and qubits required for both the standard QPE and the semiclassical implementations in Table 1. We can see that this variant of QPE employs fewer qubits and gates. As the precision grows, the number of two-qubit gates increases for both implementations. However, the number of two-qubit gates saved using this variant of 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 this implementation does not change. This contrasts with the linear growth in the standard QPE.

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

Table 1. Comparison of the number of two-qubit ZZPhase gates and qubits in both QPE implementations for estimating eigenvalues to different precisions.

In order to quantify the performance of both implementations, we compared the empirical distributions of measurement results from the circuit execution on quantum hardware to the distribution obtained from noiseless simulation. One way to compare two probability mass functions, p and q, is to use the fidelity metric[24]:

$$F(p,q) = (\sum_i \sqrt{p_i q_i})^2, \tag{15}$$

where $F(p,q) \in [0,1]$. This is a measure of the closeness of the measured quantum state, in the computational basis, on the quantum hardware to the quantum state measure on the noisy simulator. The higher this number, the better, as it means that the quantum state prepared on quantum hardware is closer to the one obtained from noiseless simulation. We approximate the variability we would expect if we were to repeatedly sample from the two unknown distributions and calculate the fidelity of the sample each time. For this, we use the bootstrap method that repeatedly resamples values from the original sample with replacement and calculates the fidelity, which is the statistic, of each resample. By doing this, we obtain the "bootstrap distribution" of the statistic. We report the median and the standard deviation of these distributions obtained with $9,999$ resamples.

We compare the achieved fidelity in both the standard and semiclassical phase estimation for three different bit precisions, in Table 2. 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 semiclassical 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±1.2	90.4±1.7	42.6±2.4
Semiclassical QPE	98.1±1.4	95.0±1.6	43.2±1.0

Table 2. Fidelity expressed in % between the probability distributions from the QPE experiments ran on quantum hardware and noiseless simulation, with 2000 shots each. We report the median and the standard deviation of the bootstrap distributions. We utilized 2000 shots, since we observed on the noisy emulator that a higher number does not change the fidelity significantly.

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 1, conditional logic, mid-circuit measurement, and qubit reset and reuse, result in the semiclassical QPE circuit being shallower than the standard implementation. As a consequence, the achieved fidelity with semiclassical QPE is, in median, 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.

Another benchmark we perform is the following. Given a fixed bit precision of four bits, we repeat the procedure for the remaining five problem instances. For this, we followed the steps in Algorithm 1 and 2, and we performed the semiclassical QPE on the Quantinuum's emulator.. With this, the optimized scaling parameter γ_{op} was obtained for each problem instance. We used the obtained value of γ_{op} to perform the semiclassical QPE applied to the unitary $e^{2\pi i \gamma_{op} \mathbf{A}}$ on hardware. The eigenvalues were estimated up to four bits. We quantify the performance of this variant of QPE by calculating the fidelity between its output distribution obtained both on the emulator and on hardware with the one obtained with noiseless simulation. The total number of qubits of the circuits is three and the median and standard deviation of two-qubit ZZPhase depth across the circuits is 63.00 ± 4.60.

In Figure 5 we compare the fidelity between the output distribution from both hardware and the emulator with the distribution obtained in noiseless simulation, respectively. We report the results for five different problem instances and display the median and the standard deviation of the bootstrap distributions corresponding to the fidelity. We first observe that the obtained fidelity on the quantum device is very high. This high fidelity allows us to utilize the output distribution obtained with this variant of quantum phase estimation to accurately estimate the eigenvalues. Additionally, we observe that the results on hardware are on average at least as good as results from its emulator, showing that the emulator gives a good lower bound of what can be expected on hardware.

4.2.3 Validating Portfolio Quality on Hardware

Now that we have benchmarked all the components required to perform Hybrid HHL^{++}, in this section, we check the quality of the solution produced by the algorithm for the constrained portfolio optimization problem with two assets. We followed the steps presented for Hybrid HHL^{++} to find the optimal scaling parameter γ_{op} for each problem instance, and we estimated the set of eigenvalues up to four bits with the semiclassical QPE executed on the emulator. Then, the eigenvalue inversion component was constructed using the heuristic method introduced in Section 3.2. We found the proper k for each problem instance such that $k \leq 4$ and each estimate is in separate bin in the output distribution of the semiclassical QPE utilizing the optimal scaling parameter. Note that k is the number of ancillary qubits utilized and the angles of the rotations are calculated with the higher resolution m-bit estimates. Note that the k depends on each problem instance as well as the size of the set, which

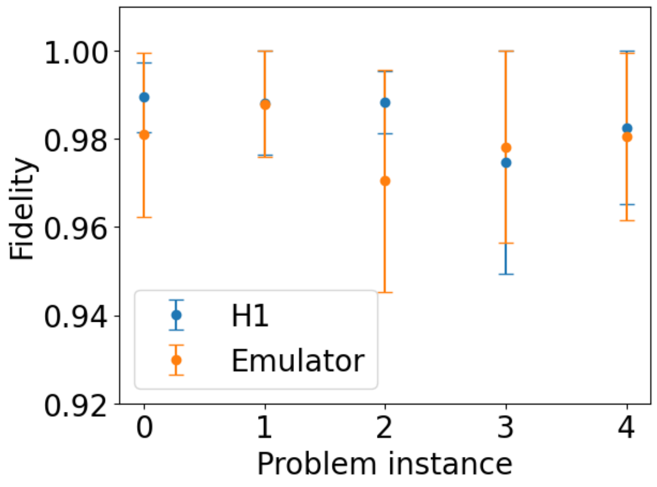


Figure 5. Fidelity between the output distribution of the semiclassical QPE executed on both the hardware and its emulator with 400 shots and the output distribution obtained in noiseless simulation, respectively, as a function of the portfolio-optimization problem instances. The semiclassical QPE is performed to the unitary corresponding to the optimal scaling parameter obtained with the proposed algorithms using the output of the circuits on the emulator. We utilized this number of shots as it has been seen on the noisy emulator of the hardware that for a higher number, there are no significant changes. It is displayed the median and standard deviation, as error bars, of the bootstrap distributions.

also depends on the classical post-processing made over the probability distribution. The compression heuristic helps when the number of qubits, k, used by the standard QPE in the HHL circuit is smaller than the bit precision, m, used by the semiclassical QPE to estimate the eigenvalues. In particular for the problems considered, the set size varied between four and six.

A way of quantifying how good the HHL algorithm is at solving the portfolio optimization problem is by comparing the estimation of the best portfolio to the classical calculated solution is by using a controlled-SWAP test[26] 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 HHL and the classical solution loaded onto a quantum state[30].

We added a new qubit to the HHL circuit called the *swap ancillary qubit*, and we loaded the normalized classically calculated solution to the linear system, i.e., $\mathbf{x}_c$, onto this qubit. The circuits also have one rotation ancillary qubit. We ran HHL and the controlled-SWAP test between the HHL output state $|\mathbf{x}\rangle$ and the quantum state encoding the classical solution $|\mathbf{x}_c\rangle$ on quantum hardware for the five problem instances of portfolio optimization considered.

We measured the rotation and swap ancillary qubits. With the results from the measurements of these two qubits using 1000 shots, we calculated the inner product between the quantum states as:

$$\sqrt{2\frac{P(|10\rangle)}{P(|10\rangle)+P(|11\rangle)}-1}, \tag{16}$$

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. A uniform distribution would result in a inner product equal to 0.

We compare the inner product obtained on the Quantinuum H1 hardware to the results obtained on its emulator with approximated noise and in noiseless simulation. Figure 6 shows the inner products corresponding to five portfolio-optimization problems considered corresponding to two assets. We first observe that the inner products obtained on hardware are very high, indicating that the solution obtained (the optimal portfolio) is very close to the analytical solution. We see that the results obtained on hardware either overlap with the results obtained in noiseless simulation or they are worse, as it occurs with one problem instance. Note that the error corresponds to the sampling error ε due to the $\mathcal{O}(1/\varepsilon^2)$ shots. The overlapping or the closeness of these values show how good the reduced HHL circuits perform on hardware. Additionally, we note that the results on hardware are on average at least as good as results from its emulator, as expected.

We summarize the results obtained across the instances in Table 3, where we compare the median and the standard deviation of the inner products obtained across the problem instances, together with the circuits' characteristics. We note that the results obtained on real hardware are, in median, very close to one and although it is worse than the ones in noiseless simulator, this difference is only of 4.17% in median.

These high inner product values obtained on hardware show how the proposed Hybrid HHL^{++} allows for solving on real quantum hardware, with high accuracy, linear system problems corresponding to portfolio optimization with two assets. Our novel contributions to improve upon the Hybrid HHL algorithm has reduced the complexity of HHL circuit. By doing this, we have managed to execute the algorithm on hardware and solve, with high accuracy, a practically-relevant problem.

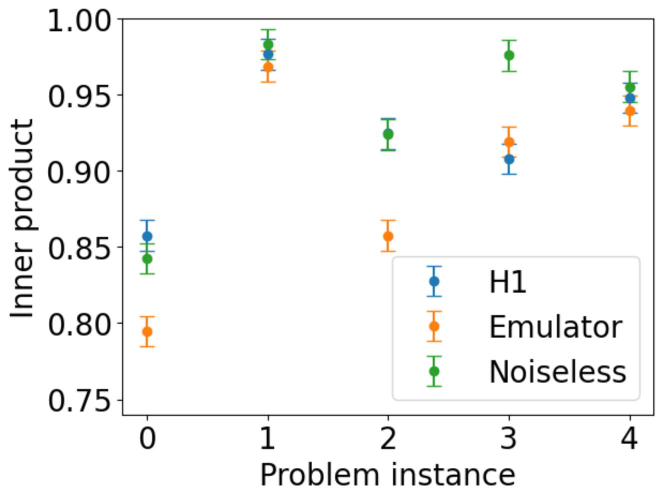


Figure 6. Inner product between the output quantum state of the HHL circuit and the analytical solution loaded as a quantum state for each of the portfolio-optimization problem instances with two assets. This metric quantifies the quality of the solution obtained. It is displayed the inner product corresponding to the execution of the circuits on hardware, its emulator with approximated noise and in noiseless simulation with 1000 shots. We utilized this number of shots as it has been seen on the noisy emulator of the hardware that for a higher number, there are no significant changes. The error bars correspond to the sampling error ε corresponding to $\mathcal{O}(1/\varepsilon^2)$ shots.

		Inner Product		
Qubits	ZZPhase Depth	H1	Emulator	Noiseless
10.00±0.00	262.00 ± 19.36	0.92± 0.04	0.92± 0.06	0.96± 0.05

Table 3. Summary of the results of the execution of the reduced HHL circuits followed by the controlled-SWAP to obtain the inner product between the output state and the analytical solution loaded as a quantum state for five portfolio-optimization problem instances. It is displayed the number of qubits, the ZZPhase depth and the inner product obtained from the execution of the circuits on the trapped-ion device, its emulator with approximated noise and noiseless simulation. It is reported the median and the standard deviation across the circuits corresponding to the five problem instances considered.

5 Scaling up for Future Hardware Demonstrations

In principle, the proposed Hybrid HHL^{++} can be applied to any QLSP, including portfolio-optimization problems, for any given size. However, the noise present in the quantum near-term hardware may be a significant limitation on how big the linear system can be. Note that some conditions of the linear system can also prevent from obtaining speedups with HHL, for example the condition number. Refer to[11]. We quantify how the noise in the device impacts the output of the semiclassical QPE and therefore the estimation of the eigenvalues to control on and the quality of the solution obtained with HHL, respectively. For this, we considered four portfolios with six assets, being the linear system 88. The characteristics of the circuits for the semiclassical QPE and HHL can been seen in Table 4. In terms of the noise, we focus on the two-quit fault probability (also known as “p2”) as the source of noise to study because the faulty two-qubit gates are currently the dominant source of error[28]. This is the probability of a fault occurring during a two-qubit gate. It corresponds to the asymmetric depolarizing probability of the device’s fully entangling two-qubit gate.

We utilize the emulator and modify the value corresponding to this fault probability. Note that the default value is 1.3810^{-3}, see Ref.[31]. We present results with both increased and decreased levels of noise For each value of the fault probability, we executed the semiclassical QPE circuits with 400 shots on the emulator tuned with that value, and we calculated the fidelity with the output of the noiseless simulation. The results for the four problem instances considered can be seen in Figure 7 (a) where we plot the fidelity as a function of the value of the two-qubit fault probability used to model the noise in the emulator. As expected, we observe that as the fault probability is reduced, the fidelity increases for the four problem instances considered. We note that this increase is not continuous but there are some slight ups and downs. However, the trend is clear. Additionally we observe that the difference in fidelity among the problem instances is mainly due to the eigenvalue distribution of each problem.

We did the same experiment but for the HHL circuit controlling on the bitstrings of the eigenvalues calculated analytically up to four bits and with the angles calculated using these values. We executed these HHL circuits on the emulator with 2000 shots using the different values of two-qubit fault probability, and we computed the inner product between the output quantum state of the HHL circuit and the analytical solution loaded as a quantum state. In practice one would do the steps in the proposed Hybrid HHL^{++} algorithm and actually utilize the output distribution from the semiclassical QPE executed on hardware to get the estimates. However, as discussed before, for this problem size (88), we have seen the the fidelity obtained on hardware is not good enough to allow us to use this output distribution. For this reason, for these experiments we used the real eigenvalues to estimate the four-bit bitstrings and to calculate the angles to rotate. We found the optimal scaling parameter following the proposed algorithm by performing the semiclassical QPE in noiseless simulator. In Fig.7 (b) we plot the inner product as a function of the level of noise for the problem instances considered. This plot gives us a sense of how the noise impacts the quality of the solution obtained and it allows us to get an idea of what future experiments on hardware would look like. We observe that as we reduce the two-qubit fault probability, the inner product improves for all the instances and for low levels of noise, it actually obtains very high values, demonstrating that the solution obtained with HHL is very close to the analytical solution. Note that when utilizing Hybrid HHL^{++} we will not estimate the eigenvalues perfectly as we did for these experiments, and so we expect to see a decrease in the inner product. However, we believe that the rate at which the quality of the HHL solution improves as we decrease the noise will still hold and this analysis allows us to gain an understanding of how future demonstrations on hardware for larger problem sizes would look like.

	Qubits	ZZPhase Depth
Semiclassical QPE	4	356
HHL	12	964

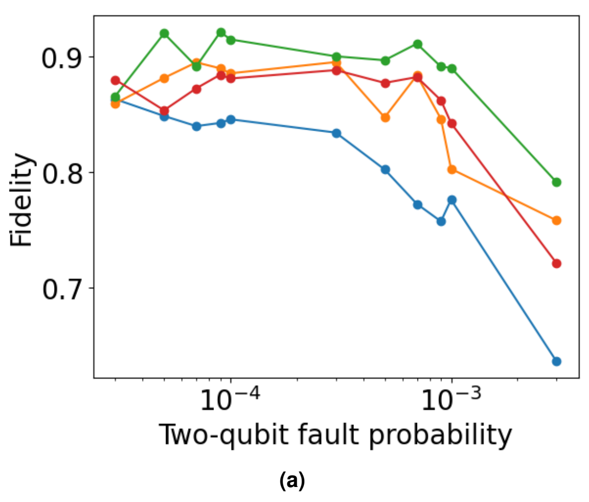
Table 4. Circuit characteristics of the semiclassical QPE and HHL corresponding to portfolio-optimization instances with six assets (88 systems). It is reported both the number of qubits and the ZZPhase depth of the circuits. Note that the values are the same for all the problem instances because the rotations are controlled on the set of the distinct analytical eigenvalues whose size is the same for all the instances.

6 Discussion

In this work, we leverage newly hardware features with the development of algorithmic tools to make the HHL algorithm more feasible on NISQ hardware. This results in a new application-oriented benchmark for small quantum hardware on quantum linear algebra tasks. This work represents a milestone towards the evaluation of quantum algorithms to tackle relevant use cases in quantum hardware available today. We improved on the Hybrid HHL introduced by Lee *et al.*in two ways.

We developed an algorithm that optimizes the scaling parameter γ for the Hamiltonian simulation required by phase estimation. We showed that the scaling of $\mathbf{A}$ by the γ, obtained with the algorithm, enabled resolving the relevant eigenvalues in the output distribution of phase estimation more accurately. Moreover, we introduce a heuristic method that reduces the complexity of the HHL circuit.

By levering newly-available hardware features we implemented a variant of the quantum phase estimation, named semiclassical QPE. This version of QPE is particularly suitable for NISQ devices since it addresses the scarcity of qubits by using just one ancilla for an arbitrary bit precision. It also tackles the fact that faulty two-qubit gates are currently the dominant source of error by replacing two-qubit gates with one-qubit gates controlled by classical bits. We have calculated with experimental results obtained on the Quantinuum H1 machine the fidelity between experimental and noiselessly simulated measurement distributions. First, for one problem instance we compared the results obtained with different number of bit precision Then, we experimentally showed that the semiclassical variant achieves high fidelity for four-bit precision, over five different problem instances. This is achieved while reducing the number of qubits and the two-qubit gate count required. The main aim of this work is to allow for hardware demonstrations of quantum algorithms for tackling applications on near-term hardware. We do not claim any speedup with our method, it is a step forward towards evaluations on real hardware.



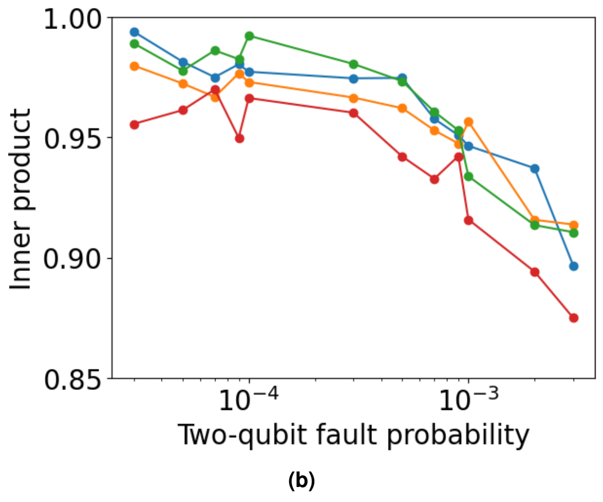


Figure 7. Benchmark metrics as a function of the value of the two-qubit fault probability (“p2”) utilized to simulate the noise in the emulator. Each color plots a different portfolio-optimization problem instance consisting of six assets (88 linear systems of equations). The metric in (a) is the fidelity between the output distribution obtained from executing the semiclassical QPE circuit on emulator with the given noise and the distribution obtained in noiseless simulation. The metric in (b) is the inner product between the output quantum state of the HHL circuit executed on the emulator with the given level of noise and the analytical solution loaded as a quantum state.

Acknowledgments

We thank Tony Uttley, Brian Neyenhuis and the rest of the Quantinuum Quantum Solutions team for assisting us on the execution of the experiments on the Quantinuum H-series trapped-ion devices. 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

Author contributions statement

R.Y. devised the algorithm idea. P.M. developed the algorithmic tools. P.M., R.Y., D.H. implemented the quantum algorithms and performed the experiments. P.M., R.Y., D.H. analyzed the data. M.P. supervised the project. All authors contributed to the technical discussions, evaluation of the results, and the writing of the manuscript.

Additional Information

The authors declare no competing interests.

Data Availability Statement

The data will be shared upon reasonable request. Please contact the corresponding author, Romina Yalovetzky.

References

1. Lee, Y., Joo, J. & Lee, S. Hybrid quantum linear equation algorithm and its experimental test on ibm quantum experience. *Sci. reports* **9**, 1–12 (Mar. 2019). Doi: 10.1038/s41598-019-41324-9.
2. Arute, F. *et al.* Quantum supremacy using a programmable superconducting processor. *Nature* **574**, 505–510, DOI: 10.1038/s41586-019-1666-5 (2019).
3. Wu, Y. *et al.* Strong quantum computational advantage using a superconducting quantum processor. *Phys. Rev. Lett.* **127**, DOI: 10.1103/physrevlett.127.180501 (2021).
4. Madsen, L. S. *et al.* Quantum computational advantage with a programmable photonic processor. *Nature* **606**, 75–81, DOI: 10.1038/s41586-022-04725-x (2022).
5. Lubinski, T. *et al.* Application-oriented performance benchmarks for quantum computing. *arXiv* (2023). 2110.03137.
6. Lubinski, T. *et al.* Quantum algorithm exploration using application-oriented performance benchmarks. *arXiv* (2024). 2402.08985.
7. Preskill, J. Quantum computing in the nisq era and beyond. *Quantum* **2**, 79 (Aug. 2018). Doi: 10.22331/q-2018-08-06-79.
8. Harrow, A. W., Hassidim, A. & Lloyd, S. Quantum algorithm for linear systems of equations. *Phys. review letters* **103**, 150502 (Oct. 2009). Doi: 10.1103/PhysRevLett.103.150502.
9. Rebentrost, P. & Lloyd, S. Quantum computational finance: quantum algorithm for portfolio optimization. *arXiv* (2018). arXiv:1811.03975.
10. Markowitz, H. Portfolio selection. *The J. Finance* **7**, 77–91 (Mar. 1952). Doi: 10.1111/j.1540-6261.1952.tb01525.x.
11. Aaronson, S. Read the fine print. *Nat. Phys.* **11**, 291–293 (Apr. 2015). Doi: 10.1038/nphys3272.
12. Bravo-Prieto, C. *et al.* Variational quantum linear solver. *arXiv* (2019). arXiv:1909.05820.
13. Huang, H.-Y., Bharti, K. & Rebentrost, P. Near-term quantum algorithms for linear systems of equations. *arXiv* (2019). arXiv:1909.07344.
14. Zhang, M., Dong, L., Zeng, Y. & Cao, N. Improved circuit implementation of the hhl algorithm and its simulations on qiskit. *Sci. Reports* **12**, 13287 (2022).
15. Möttönen, M., Vartiainen, J. J., Bergholm, V. & Salomaa, M. M. Transformation of quantum states using uniformly controlled rotations. *Quantum Inf. Comput.* **5**, 467–473 (Sep. 2019). Doi: 10.26421/QIC5.6-5.
16. Griffiths, R. B. & Niu, C.-S. Semiclassical fourier transform for quantum computation. *Phys. Rev. Lett.* **76**, 3228–3231 (Apr 1996). Doi: 10.1103/PhysRevLett.76.3228.
17. Zhang, X., Yang, Z. & Zhang, X. Simplified experimental scheme of quantum algorithm for solving linear equations with single photons. *Opt. Express* **27**, 3369–3378 (2019).
18. Aleksandrowicz, G. *et al.* Qiskit: An open-source framework for quantum computing. *zenodo* (2019). Doi: 10.5281/zenodo.2562111.
19. Mohammadbagherpoor, H., Oh, Y.-H., Singh, A., Yu, X. & Rindos, A. J. Experimental challenges of implementing quantum phase estimation algorithms on ibm quantum computer. *arXiv* (2019). arXiv:1903.07605.
20. Beauregard, S. Circuit for shor's algorithm using 2n+3 qubits. *Quantum Inf. Comput.* **3**, 175–185 (Mar. 2003). Doi: 10.26421/QIC3.2-8.
21. Mosca, M. & Ekert, A. The hidden subgroup problem and eigenvalue estimation on a quantum computer. In *NASA International Conference on Quantum Computing and Quantum Communications*, 174–188 (Springer, Feb. 1998). Doi: 10.1007/3-540-49208-9_15.
22. Córcoles, A. *et al.* Exploiting dynamic quantum circuits in a quantum algorithm with superconducting qubits. *Phys. Rev. Lett.* **127**, DOI: 10.1103/physrevlett.127.100501 (2021).
23. Dobšíček, M., Johansson, G., Shumeiko, V. & Wendin, G. Arbitrary accuracy iterative quantum phase estimation algorithm using a single ancillary qubit: A two-qubit benchmark. *Phys. Rev. A* **76**, 030306 (Sep. 2007). Doi: 10.1103/PhysRevA.76.030306.
24. Nielsen, M. A. & Chuang, I. L. *Quantum Computation and Quantum Information* (Cambridge University Press, 2000).
25. Brassard, G., Høyer, P., Mosca, M. & Tapp, A. Quantum amplitude amplification and estimation, DOI: 10.1090/conm/305/05215 (2002).

26. Beausoleil, R. G. and Munro, W. J. and Spiller, T. P. and van Dam, W. K. *Tests of quantum information (US Patent 7,559,101 B2)* (2008).

27. Sivarajah, S. *et al.* t$|$ket$\rangle$: a retargetable compiler for nisq devices. *Quantum Sci. Technol.* **6**, 014003 (Nov. 2020). Doi: 10.1088/2058-9565/ab8e92.

28. Pino, J. *et al.* Demonstration of the trapped-ion quantum ccd computer architecture. *Nature* **592**, 209–213 (Apr. 2021). Doi: 10.1038/s41586-021-03318-4.

29. Moses, S. *et al.* A race track trapped-ion quantum processor. *arXiv preprint arXiv:2305.03828* (2023).

30. Buhrman, H., Cleve, R., Watrous, J. & De Wolf, R. Quantum fingerprinting. *Phys. Rev. Lett.* **87**, 167902 (Sep. 2001). Doi: 10.1103/PhysRevLett.87.167902.

31. Quantinuum H1 emulator product data sheet [retrieved 03/02/2024]. URL: https://www.quantinuum.com/hardware/h1/quantumemulation.