

Variational-Adiabatic Quantum Solver for Systems of Linear Equations with Warm Starts

Claudio Sanavio and Sauro Succi

*Italian Institute of Technology, Viale Regina Elena 291, 10161, Rome, Italy**

Fabio Mascherpa, Alessia Marruzzo, and Alfonso Amendola

Eni S.p.A., San Donato Milanese, Italy†

(Dated: June 2, 2025)

We propose a revisited variational quantum solver for linear systems, designed to circumvent the barren plateau phenomenon by combining two key techniques: adiabatic evolution and warm starts. To this end, we define an initial Hamiltonian with a known ground state which is easily implemented on the quantum circuit, and then “adiabatically” evolve the Hamiltonian by tuning a control variable in such a way that the final ground state matches the solution to the given linear system. This evolution is carried out in incremental steps, and the ground state at each step is found by minimizing the energy using the parameter values corresponding to the previous minimum as a warm start to guide the search. As a first test case, the method is applied to several linear systems obtained by discretizing a one-dimensional heat flow equation with different physical assumptions and grid choices. Our method successfully and reliably improves upon the solution to the same problem as obtained by a conventional quantum solver, reaching very close to the global minimum also in the case of very shallow circuit implementations.

I. INTRODUCTION

Linear systems of equations can be used to describe countless physical phenomena, ranging from transport models to linear fluid dynamics or optimization processes in logistics [1]. Linear algebraic systems can arise from the discretization of linear partial differential equations (PDEs) in space and time, and even nonlinear PDEs can be reduced to a linear system by employing suitable linearization techniques, such as Taylor expansion or Carleman linearization as applied to the lattice Boltzmann equation [2–5]. Given the ubiquity of such mathematical problems and the possibly very large complexity of their numerical solution, the relevance of devising efficient methods for this task can hardly be overstated. A straightforward method for solving a linear system of equations involves brute-force inversion, for instance via Gaussian elimination, but this method suffers from numerical instability and a computational complexity of order $\mathcal{O}(N^3)$ in the dimension N of the problem. Faster methods, such as Krylov subspace-based or iterative relaxation methods, can reduce this cost significantly (even down to order $\mathcal{O}(N)$) but introduce some degree of approximation and controlled error [6].

In the last decade, quantum computing for the solution of linear systems has become an emerging field of study, driven by recent technological advances and the promise of a possibly exponential speedup in several different fields, from the simulation of quantum systems to the solution of mathematical models with applications e.g. in engineering, fluid dynamics and geophysics [7–9]. Among these, in 2008 the Harrow-Hassidim-Lloyd

(HHL) algorithm [10] showed how linear systems could be solved by a quantum computer up to an error ϵ with a computational complexity of order $(\log N)/\epsilon$, later improved to $\log(N/\epsilon)$ by Childs et al. [11, 12]. The latter method, further improved with quantum signal processing techniques [13], currently represents the state-of-the-art quantum algorithm for solving a linear system of equations. Unfortunately, while theoretically excellent, this algorithm involves large numbers of logical quantum gates and therefore requires a fault-tolerant quantum computer to give meaningful results. Despite the significant improvements achieved by recent developments in quantum hardware design and operation, current machines do not yet support the levels of error correction needed to suppress all the quantum noise involved in any reasonably sized implementation of the method [14].

For this reason, most research efforts in recent years have shifted to algorithms that could turn out to be useful on so-called Noisy Intermediate-Scale Quantum (NISQ) devices, which comprise hundreds of qubits and noisy quantum gates. Among such methods, hybrid variational algorithms [15] have been attracting particular attention. In a variational quantum linear solver (VQLS) [16], one constructs a quantum circuit whose Hilbert space is large enough to accommodate the original problem (i.e. the number of qubits n needs to be such that $2^n \geq N$) and acts on those qubits with a given operator $U(\theta)$, known as an *ansatz*, typically consisting of one- and two-qubit gates and controlled by a set of parameters θ : one then defines a cost function such that its minimum with respect to θ corresponds to a state $|x(\theta)\rangle$ representing the solution of the linear system.

However, even variational quantum algorithms, although promising due to their logarithmic use of resources and potential applications that venture into the field of quantum machine learning [17, 18], suffer from

* claudio.sanavio@iit.it

† fabio.mascherpa@eni.com

a serious drawback in the form of the so-called Barren Plateaus (BP) [15, 16, 19–21]. In fact, as the number of qubits increases, the gradient of the cost function becomes smaller in magnitude throughout most of the parameter space. Crucially, this decrease turns out to be exponential in the number of qubits, i.e. for any small number $\epsilon > 0$ the probability of finding a set of parameters θ such that $|\nabla_{\theta} f| > \epsilon$ decreases as $P_{\theta}(|\nabla_{\theta} f| \geq \epsilon) \sim \mathcal{O}(2^{-n})$. In such regions, both gradient-based and global optimizers are unable to minimize the cost function efficiently, defeating the purpose of exponentially reducing the problem size in the first place.

The BP problem has been studied extensively, with various approaches and ideas on how to avoid or mitigate it, but it appears to be a built-in feature of the variational quantum algorithm family. In particular, it is worth mentioning one recent study [22], which uncovered a major pitfall of using local cost functions, i.e. cost functions computed from properties of individual qubits or small groups of qubits which do not grow with the system size. Local cost functions were considered appealing for variational methods, because they are better behaved than global ones and may not exhibit BPs [19]. However, in Ref. [22], the authors highlighted a link between the improved scaling properties of variational algorithms employing such cost functions and the reduction in complexity of the variational problem itself if formulated this way, and showed that any problem that can be solved by a variational method with local cost function is also efficiently solvable by a classical algorithm. To put it bluntly, formulating a variational algorithm with a local cost function removes both the BP and any quantum advantage.

In this work, we attempt a different route around this hurdle, building upon two ideas already tested in the context of quantum Hamiltonian simulation. First, following Ref. [23], we redefine the variational problem as a problem family parametrized by a guiding variable, which is used as a control to gradually move from a trivial linear system to the one of interest in a fashion similar to the principle of adiabatic quantum computing. The other key ingredient of the proposed approach is the well-known concept of warm starts in variational methods, described e.g. in Ref. [24]: we use available information on the problem to initialize the search for a minimum in a region with a sufficiently steep descent path towards it—in other words, outside the BP—by changing the parameter landscape in steps small enough to follow the minimum as it shifts from the initial, trivial solution to the one of the problem at hand.

In Section II, we will describe the proposed approach in detail. In Section III, we will demonstrate an example application by solving linear systems derived from a finite-difference formulation of the heat flow equation in one spatial dimension using different physical settings of the problem. Finally, in Section IV, we will discuss the quality of the results and the performance of the algorithm, and present our conclusions and outlook.

II. ADIABATIC VARIATIONAL SOLVER WITH WARM STARTS

Given an invertible linear system

$$A\mathbf{x} = \mathbf{b}, \quad (1)$$

we can employ the standard VQLS [16] to find the solution of the corresponding quantum linear system

$$A|x\rangle \propto |b\rangle, \quad (2)$$

or in inverted form,

$$|x\rangle = \frac{A^{-1}|b\rangle}{\|A^{-1}|b\rangle\|}, \quad (3)$$

by expressing Eq. (1) in the language of quantum states and operators defined on a suitable parametric quantum circuit, and then solving for the parameters. From this point on, we will assume the system Eq. (1) to be $N = 2^n$ -dimensional, padding \mathbf{x} and \mathbf{b} with zeros and A with the identity matrix on any dimensions not present in the original system given. The equivalent quantum system (2) is defined by normalizing the vectors \mathbf{x} and \mathbf{b} to unity, so that they can be interpreted as quantum states $|x\rangle$ and $|b\rangle$, and relaxing the equality since A is not necessarily unitary.

In a variational setting, the state $|x(\theta)\rangle$ is obtained by applying a parametric ansatz $U(\theta)$ to the initial state of the quantum circuit:

$$|x(\theta)\rangle = U(\theta)|0\rangle, \quad (4)$$

and the VQLS identifies the correct values of the parameters to solve Eq. (2) by minimizing the appropriate cost function, according to the variational principle

$$C(\theta^*) = \min_{\theta} C(\theta) \iff |x(\theta^*)\rangle \approx |x\rangle. \quad (5)$$

Note that the approximate relation above is a consequence of the so-called *expressibility* of the ansatz $U(\theta)$ [25]: in general, $|x(\theta^*)\rangle$ will be the closest possible state to the real solution $|x\rangle$ among those accessible by U , which does not necessarily contain $|x\rangle$ itself.

Although elegant, the VQLS as it is offers little practical value, since optimizing a global cost function is generally a hard task due to the cost function landscape taking the form of a BP. Nevertheless, previous works have highlighted both its wide range of potential applications and the quality of the results obtained either for small-scale problems or for more complex ones which allow for local cost functions and efficient classical solutions.

A. Adiabatic variational solver

In order to address the challenges presented by the standard VQLS and to devise a strategy capable of working around them, we considered a variation of the adiabatic quantum solver introduced in Ref. [23], which defines a parametric version of the linear system Eq. (2) and

uses a stochastic evolution simulation to solve it adiabatically. Rather than following that path, we will borrow the idea of an adiabatic parametrization of the problem and integrate it into the VQLS.

First, let A be a non-singular $N \times N$ matrix with $N = 2^n$ and define the parametric linear system

$$A(s) |x(s)\rangle \propto |\bar{b}\rangle, \quad (6)$$

where

$$\begin{aligned} A(s) &:= (1-s)\mathbb{I} + sA, \\ |\bar{b}\rangle &:= |b\rangle \end{aligned} \quad (7)$$

if A is positive, and

$$\begin{aligned} A(s) &:= (1-s)\sigma_z \otimes \mathbb{I} + s\sigma_x \otimes A, \\ |\bar{b}\rangle &:= |+\rangle \otimes |b\rangle \end{aligned} \quad (8)$$

if it is not positive. The latter definition ensures that $A(s)$ is invertible for any s (and hence that the corresponding solution $|x(s)\rangle$ exists), and implies adding an ancilla qubit to perform the algorithm.

The key of the parametric description of the problem is that at $s = 0$ one immediately finds that $|x(0)\rangle = |b\rangle$ or $|x(0)\rangle = |-\rangle \otimes |b\rangle$, depending on the positivity of A , while at $s = 1$ the solution is $|x(1)\rangle = |x\rangle$ or $|x(1)\rangle = |+\rangle \otimes |x\rangle$. Therefore, the parametrization in s allows us to interpolate smoothly between the trivial linear problem with $A(s)$ proportional to the identity and the actual problem defined by Eq. (2).

Furthermore, it can be readily shown that for each s the solution $|x(s)\rangle$ of Eq. (6) is also the ground state of the Hamiltonian

$$H(s) := A^\dagger(s) (\mathbb{I} - \Pi_b) A(s), \quad (9)$$

where $\Pi_b = |\bar{b}\rangle\langle\bar{b}|$ is the projector on the state $|\bar{b}\rangle$, and that this ground state is unique and has eigenvalue 0. Therefore, combining all these elements together, one may treat the problem as an adiabatic search for the ground state of the Hamiltonian $H(s)$ as s gradually increases from 0 to 1 [23].

The problem can then be approached variationally by simply defining a cost function family parametrized by s , which corresponds to the energy expectation value

$$C_s(\theta) := \langle 0 | U^\dagger(\theta) H(s) U(\theta) | 0 \rangle. \quad (10)$$

The solution $|x\rangle$ is obtained by first solving the trivial problem at $s = 0$ and then letting the parameters follow the minimum as s changes. This raises two questions: how do we find the first minimum, and how do we discretize the smooth parametrization into steps small enough to “track” the minimum as it moves through the parameter space.

The first question can be ignored if one assumes an oracle capable of preparing arbitrary quantum states; otherwise, one might suspect that finding the correct parameters to construct the state $|b\rangle$ may be just as hard

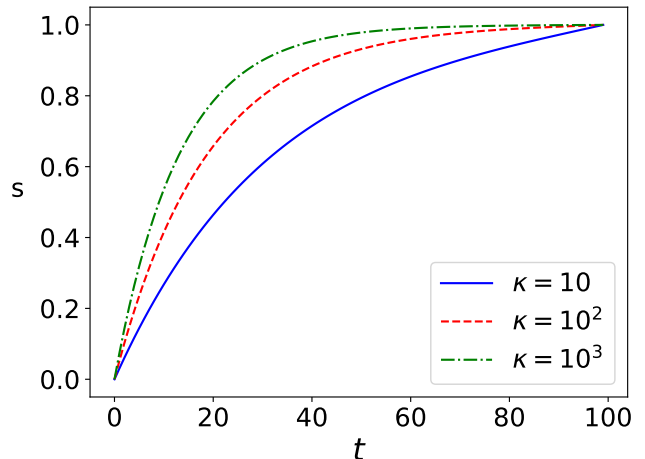


FIG. 1. Increase of s as a function of $T = 100$ values of v for condition numbers $\kappa = 10$ (solid blue line), $\kappa = 10^2$ (dashed red line) and $\kappa = 10^3$ (dot-dashed green line). Note the concentration of $s(v)$ at values closer to 1 for higher κ .

as solving the original problem. Fortunately, this is not the case: it is always possible to apply a unitary transformation S to the linear system Eq. (1) such that \mathbf{b} is mapped onto a vector with a simple expression in terms of the ansatz parameters, such as the first standard basis vector $\mathbf{e}_1 = (1, 0, 0, \dots)^t$:

$$SAS^\dagger S\mathbf{x} = \tilde{A}\tilde{\mathbf{x}} = \mathbf{e}_1, \quad (11)$$

where \tilde{A} and $\tilde{\mathbf{x}}$ are the transformed matrix and solution. One convenient choice for S is known as the Householder matrix [26], which is defined as follows: first subtract \mathbf{e}_1 from the normalized \mathbf{b} to define a vector $\mathbf{v} := \mathbf{b}/\|\mathbf{b}\| - \mathbf{e}_1$, then use this to define a reflection matrix as

$$S_H := \mathbb{I} - 2 \frac{\mathbf{v}\mathbf{v}^\dagger}{\mathbf{v} \cdot \mathbf{v}}. \quad (12)$$

It can be easily checked that S_H is unitary, Hermitian and involutory, i.e. $S = S^\dagger = S^{-1} = S$. From now on, and without loss of generality, we will assume the linear system to be already expressed in this basis, i.e. $\mathbf{b} = \mathbf{e}_1$, or $|b\rangle = |0\rangle$ on the quantum circuit.

The second question is far from trivial, and will be the focus of the rest of this section.

B. Improving efficiency by using warm starts

The question of how to increase s by a controlled amount δs such that the minimum of the new cost function $C_{s+\delta s}(\theta)$ can be found efficiently knowing the previous minimum $C_s(\theta^*)$ is a subtle one. Recent works [24, 27] have proved some rigorous links between a change in the cost function and the distance between the minima before and after the change, in the context of Hamiltonian evolution simulation. In particular,

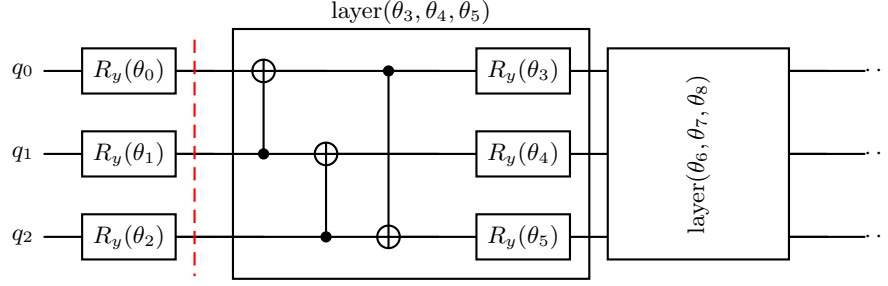


FIG. 2. Structure of the parametric quantum circuit $U(\theta)$ for the case $n = 3$. The circuit starts with a R_y rotation for each qubit, followed by $d = 2$ repetitions of the layer detailed in the first box, which combines more single-qubit rotations and a set of two-qubit CNOT gates arranged in a ring configuration.

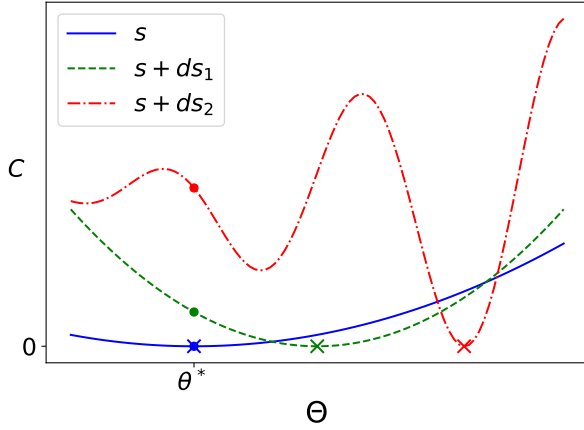


FIG. 3. A slice of the energy landscape along a preferred direction Θ : at a given s , the blue cross locates the minimum of the cost function (solid blue line). If we evolve the system by δs_1 , the new global minimum (green cross) is still in the same valley as the previous one (dashed green line), but if we evolve it by δs_2 we overshoot the convexity region and the new global minimum (red cross) appears in another valley (dot-dashed red line), with no descending path joining it to the previous minimum (red dot).

in Ref. [24] it is shown that the evolution of a quantum state may be simulated variationally with rigorously bounded warm starts, i.e. letting the system evolve in timesteps short enough for the gradient of the cost function (which in that problem is an infidelity measure $\mathcal{I}_{\delta t}(\theta) = 1 - |\langle \psi(\theta) | e^{-iH\delta t} | \psi(\theta^*) \rangle|^2$) to be non-negligible at the previous minimum θ^* , which can be reliably used as the starting point in the new minimization. We try to transfer this procedure to our adiabatic solver, for which no such rigorous bounds were proven, in a more heuristic way.

First, any procedure to determine the step size δs should take into account the fact that, depending on the condition number κ of the matrix A , the rate of change of $H(s)$ and $|x(s)\rangle$ can be expected to increase as s approaches 1. Therefore, splitting the interval between 0 and 1 into a number of identical steps is not an effi-

cient strategy to track the adiabatic transformation of the problem. A more suitable prescription which accounts for κ was introduced in Ref. [23]: let $v \in [v_{\min}, v_{\max}]$, where

$$v_{\min} = \sqrt{\frac{2\kappa^2}{1+\kappa^2}} \log(\kappa\sqrt{1+\kappa^2} - \kappa^2), \quad (13)$$

$$v_{\max} = \sqrt{\frac{2\kappa^2}{1+\kappa^2}} \log(\sqrt{1+\kappa^2} + 1),$$

and define the following dependence of s on v :

$$s(v) := \frac{e^{v\sqrt{\frac{1+\kappa^2}{2\kappa^2}}} + 2\kappa^2 - \kappa^2 e^{-v\sqrt{\frac{1+\kappa^2}{2\kappa^2}}}}{2(1+\kappa^2)}. \quad (14)$$

It can be easily checked that $s(v)$ increases monotonically and that $s(v_{\min}) = 0$ and $s(v_{\max}) = 1$. Applying Eq. (14) to a set of evenly spaced values $v_j = v_{\min} + \frac{j}{T}(v_{\max} - v_{\min})$, with j running from 0 to some maximum integer T , one obtains an adiabatic parametrization that tilts the distribution of s_j towards 1 by an amount depending on the problem at hand. The specific dependence expressed by Eq. (14) is obtained by requiring that $\|\partial_v |x(s(v))\rangle\| \leq 1$ for all v (the details of the derivation may be found in Ref. [23]). In Fig. 1, we show the nonlinear progression of s using $T = 100$ points for different values of κ : note how the points tend to cluster more and more near the upper limit as κ increases.

The other element determining the cost function landscape and its dependence on s is, of course, the structure of the ansatz itself. We chose to adopt the commonly used alternating ansatz introduced in Ref. [16], which consists of a number d of layers each comprising a rotation of each qubit by an independent angle and a loop of two-qubit entangling gates (either CNOT or CZ), as shown in Fig. 2. From now on, for the sake of simplicity, we will assume A to be positive, so the circuit size will always be $n = \log_2 N$; the whole treatment is easily generalized beyond positive matrices by introducing an ancilla qubit and following the definitions given in Eq. (8).

The cost function $C_s(\theta)$ for such a circuit has some useful properties, which will allow us to improve the adiabatic progression defined above and take full advantage

of the warm-start approach. To see how, let us introduce some new notation in order to make the dependence of $C_s(\theta)$ on s explicit: define

$$\begin{aligned}\hat{a} &:= \Pi_b - A\Pi_b - \Pi_b A + A\Pi_b A, \\ \hat{b} &:= -2\Pi_b + A\Pi_b + \Pi_b A, \\ \hat{c} &:= \Pi_b,\end{aligned}\tag{15}$$

so that Eq. (9) may be rewritten as

$$H(s) = s^2 \hat{a} + s \hat{b} + \hat{c}.\tag{16}$$

Then $C_s(\theta)$ can be expanded in powers of s as well, with each coefficient equal to the appropriate expectation value on the quantum circuit state (denoted here by the shorthand $\langle O \rangle_\theta = \langle 0 | U^\dagger(\theta) O U(\theta) | 0 \rangle$ for conciseness):

$$C_s(\theta) = \langle H(s) \rangle_\theta = s^2 \langle \hat{a} \rangle_\theta + s \langle \hat{b} \rangle_\theta + \langle \hat{c} \rangle_\theta.\tag{17}$$

This allows one to calculate the value of the cost function at this particular point in parameter space for any *other* value of s as well: adding an arbitrary increment δs in Eq. (17) and grouping the terms by powers of δs , one immediately finds

$$C_{s+\delta s}(\theta) = \delta s^2 \langle \hat{a} \rangle_\theta + \delta s (2s \langle \hat{a} \rangle_\theta + \langle \hat{b} \rangle_\theta) + C_s(\theta).\tag{18}$$

Note that this expression is exact for any δs .

The dependence of $C_s(\theta)$ on the ansatz parameters, on the other hand, is such that its derivatives can be computed exactly by employing the parameter-shift rule [28]: in particular, the gradient $\nabla[C_s(\theta)]$ and Hessian $\mathcal{H}[C_s(\theta)]$ are given by

$$\begin{aligned}\nabla_i C_s(\theta) &= \frac{1}{2 \sin(\beta)} (C_s(\theta + \beta \mathbf{e}_i) - C_s(\theta - \beta \mathbf{e}_i)), \\ \mathcal{H}_{ij} C_s(\theta) &= \frac{1}{2 \sin(\beta)} (\nabla_j C_s(\theta + \beta \mathbf{e}_i) - \nabla_j C_s(\theta - \beta \mathbf{e}_i)),\end{aligned}\tag{19}$$

for any β , so that two evaluations of the cost function in $\theta \pm \beta \mathbf{e}_i$ give an exact gradient component and four evaluations in $\theta \pm \beta \mathbf{e}_i \pm \beta \mathbf{e}_j$ (or two in $\theta \pm 2\beta \mathbf{e}_i$ plus $C_s(\theta)$ itself for $i = j$) give an exact Hessian matrix element.

Combining this with Eq. (18), we can use $C_s(\theta)$ to construct the Hessian $\mathcal{H}[C_{s+\delta s}(\theta)]$ for any given δs :

$$\begin{aligned}\mathcal{H}[C_{s+\delta s}(\theta)] &= \delta s^2 \mathcal{H}[\langle \hat{a} \rangle_\theta] + \delta s (2s \mathcal{H}[\langle \hat{a} \rangle_\theta] + \mathcal{H}[\langle \hat{b} \rangle_\theta]) + \mathcal{H}[C_s(\theta)],\end{aligned}\tag{20}$$

where each matrix element can be computed by applying parameter shifts as described above.

This may be used as a guide to determine the optimal δs for a warm start. To see how, suppose that we found the global minimum of the cost function for $s = s^*$ at $\theta = \theta^*$, i.e. $C_{s^*}(\theta^*) = 0$. By definition, $\mathcal{H}[C_{s^*}(\theta^*)]$ has only nonnegative eigenvalues at this point. In order to increment s as much as possible without the new cost function $C_{s^*+\delta s}(\theta)$ becoming too flat at $\theta = \theta^*$, one

can use Eq. (20) to find the largest possible δs such that no eigenvalue of $\mathcal{H}[C_{s+\delta s}(\theta^*)]$ becomes negative, i.e. such that θ^* is still within the same trough or valley as the new minimum to be found. Fig. 3 shows a one-dimensional example of how a function may change and its minimum may shift, either remaining within the same convex region as the previous minimum or moving out of reach from it.

Note that this solution of a nonlinear system does not impact the overall complexity of the algorithm, because $\mathcal{H}[C_s(\theta)]$ is an $n_p \times n_p$ symmetric matrix, where $n_p = n(d+1)$ is the number of ansatz parameters and depends on the number of qubits n and the circuit depth (i.e. the number of layers in the ansatz) d . Computing the eigenvalues of $\mathcal{H}[C_{s+\delta s}(\theta)]$ therefore has a cost of order $\mathcal{O}(n_p^3)$. The optimal δs can be computed using iterative Newton-like methods, which determine δs up to some error ϵ in $\mathcal{O}(\log(\log(1/\epsilon)))$ steps: the total complexity of finding δs up to this error is therefore of order $\mathcal{O}(n_p^3 \log(\log(1/\epsilon)))$.

Of course, there is no guarantee that a numerical optimization will actually find a global minimum of the cost function. It could find a local minimum, or the best possible solution compatible with the ansatz, or it could return a saddle point. Therefore, it is necessary to check the Hessian at the current s before moving on to the calculation of the next one through the procedure described above. We combined this strategy with the predetermined sequence of s values given by Eq. (14), using the latter as a fallback option whenever the δs suggested by the Hessian calculation was not suitable for a warm start or the starting point was not a minimum.

More specifically, depending on the result of this convexity analysis, there are four possibilities at each step in the adiabatic variable s .

- $\mathcal{H}[C_{s^*}(\theta^*)]$ not positive: we are not in a minimum. Increment to next s suggested by Eq. (14).
- $\delta s < 0$: all positive δs leave the next minimum within reach. Increment to $s = 1$ directly.
- $0 \leq \delta s \leq \delta s_{\min}$ for a set δs_{\min} : progress via warm starts is impractically slow. Increment by δs_{\min} .
- $\delta s > \delta s_{\min}$: increment by δs .

In our applications, δs_{\min} will be set equal to the increment suggested by Eq. (14), but other choices may also be considered.

In the next section, we will demonstrate this algorithm on a discretized one-dimensional heat flow equation and analyze its performance and results for this particular test case.

III. THE HEAT FLOW EQUATION

To address the problem of solving a linear, one-dimensional differential equation with a variational quan-

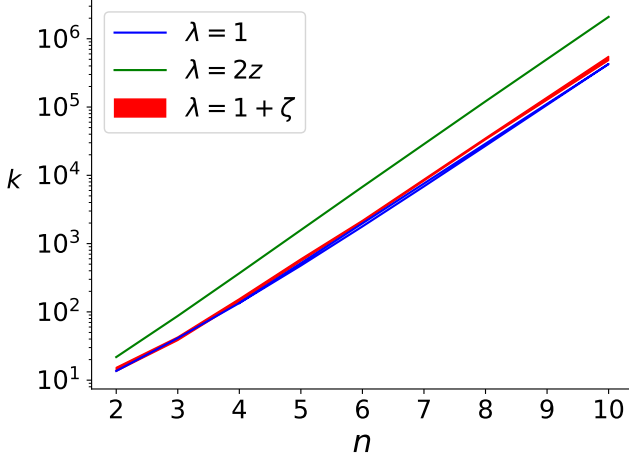


FIG. 4. Condition number κ of the matrix A for the heat flow equation in the discretized form given in Eq. (23), for the cases $\lambda(z) = 1$ (blue line), $\lambda(z) = 1 + \zeta(z)$ with ζ normally distributed with $\sigma = 0.2$ (red line) and $\lambda(z) = 2z$ (green line). κ increases exponentially with the number of qubits for both the analyzed cases.

tum solver, we consider the stationary state of the inhomogeneous non-uniform heat flow equation

$$\frac{\partial}{\partial t} f(z, t) = \frac{\partial}{\partial z} \left(\lambda \frac{\partial f(z, t)}{\partial z} \right) + Q_r(z), \quad (21)$$

where f is the temperature distribution, Q_r is the heat production rate and λ is the thermal conductivity of the medium. Usually, λ is a function of the position z and of the temperature f itself, and thus the general equation is nonlinear. However, in many cases of physical interest, including in applications to geological modeling [9], one may assume a temperature-independent thermal conductivity.

The stationary state is obtained by setting the left-hand side of Eq. (21) to zero and solving

$$\frac{\partial}{\partial z} \left(\lambda(z) \frac{\partial f(z)}{\partial z} \right) = -Q_r(z). \quad (22)$$

In order to solve the problem for a system of length L , we first discretize Eq. (22) into an N -site regular lattice with spacing $\Delta z = L/N$, thus transforming it into a linear problem of the general form Eq. (1). The resulting matrix A , for open boundary conditions, is the $N \times N$ tridiagonal matrix with components

$$\begin{aligned} A_{i,i} &= -\frac{2\lambda_i}{\Delta z^2} \\ A_{i,i-1} &= \frac{1}{\Delta z^2} \left(\lambda_i + \frac{\lambda_{i+1} - \lambda_{i-1}}{4} \right) \\ A_{i,i+1} &= \frac{1}{\Delta z^2} \left(\lambda_i - \frac{\lambda_{i+1} - \lambda_{i-1}}{4} \right) \end{aligned} \quad (23)$$

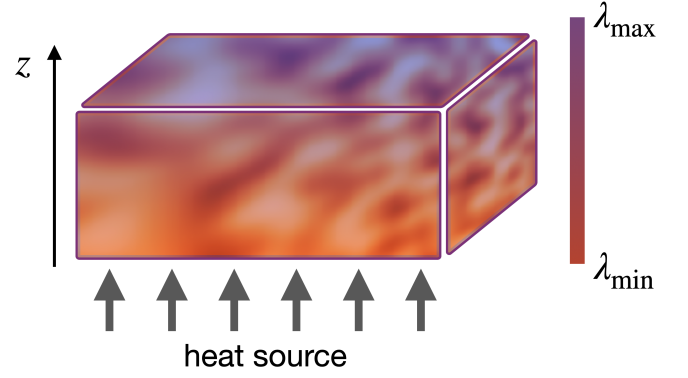


FIG. 5. Representation of a block of some heterogeneous material with conductivity increasing along the z coordinate up to some noise ζ : $\lambda(z) = cz + \zeta$. A heat source located at the bottom of the bulk, where $\lambda(z) = \lambda_{\min}$, is shown and can be represented in Eq. (22) by a pointlike source term $Q_r(z) \propto \delta(z)$. Our goal is to find the stationary state of the heat flow equation for this problem on a finite lattice.

for $i = 1, \dots, N$, $\lambda_i = \lambda(i\Delta z)$, and the discretized source term directly yields the components of the vector \mathbf{b} as $b_i = -Q_r(i\Delta z)$. Note that $N = 2^n$, so each added qubit doubles the spatial resolution of the discretized problem.

We will now proceed to test the adiabatic quantum variational algorithm on several instances of this linear problem, defined by the form of the conductivity function $\lambda(z)$ and of the source term $Q_r(z)$.

For the conductivity, we will first consider a homogeneous material, characterized by $\lambda(z) = \lambda_0$. Next, we will include the presence of impurities in our physical model by adding a noise term, so that $\lambda(z) = \lambda_0 + \zeta$, where ζ is a normally distributed random variable with zero mean and standard deviation σ . Finally, we will consider a different system in which the conductivity increases linearly with the position: $\lambda(z) = \frac{2z}{L}\lambda_0$, with or without added noise. Such a linear increase in conductivity is a common condition in geophysical settings, in which a uniform conductivity variation with depth and pressure is typically observed [29, 30].

As for the heat source, we will compare different cases, all decaying exponentially with respect to z , so that $Q_r(z) = Q_0 e^{-lz/L}$ with $l \geq 0$. In particular, we will let l take on different values and compare the effect of a steeper or slower decay, including the extreme cases $l = 0$, i.e. a constant source term $Q_r(z) = Q_0$, and $l \rightarrow \infty$, i.e. a point source $Q_r(z) \propto \delta(z)$. In practice, we will consider either $\mathbf{b} = Q_0 \mathbf{e}_1$ for the pointlike case, or

$$b_{lj} = Q_0 e^{-jl\Delta z/L} \quad (24)$$

with l chosen between 0, 2 and 5. As discussed at the end of Section II A, in the latter cases we will always perform a change of basis via the Householder transformation Eq. (12) before feeding the problem into our variational algorithm, so that the quantum circuit is always

initialized in the state $|0\rangle = |b\rangle$. A pictorial representation of a bulk of non-homogeneous material, with noisy conductivity along the vertical direction and a pointlike heat source, is shown in Fig. 5.

For the purpose of numerical solution, we choose units such that $L = 1$, $\lambda_0 = 1$ and $Q_0 = 1$. All results presented in the following subsections are dimensionless.

A. Case I: Constant conductivity

In the homogeneous case, we have a constant conductivity $\lambda(z) = \lambda_0 = 1$, so that each $\lambda_i = 1$ in Eq. (23). The lattice spacing is $\Delta z = 2^{-n}$, so that

$$A_{ij} = -2^{1-2n}\delta_{ij} + 2^{-2n}(\delta_{i,j+1} + \delta_{i+1,j}).$$

The eigenvalues of any matrix of this particular form can be computed analytically and are given by $\alpha_k = 4\sin^2\left(\frac{\pi k}{2(N+1)}\right)$, with $k = 1, \dots, N$. This implies that the condition number of A grows quadratically with the number of lattice sites N , or exponentially with the number of qubits n :

$$\kappa = \frac{\alpha_N}{\alpha_1} = \frac{\sin^2\left(\frac{\pi N}{2(N+1)}\right)}{\sin^2\left(\frac{\pi}{2(N+1)}\right)} \sim_\infty N^2 = 2^{2n}.$$

Fig. 4 shows the condition number of the homogeneous case as a function of the number of qubits: the logarithmic scale on the y -axis clearly shows the exponential dependence on n . In the same figure, we show the condition number calculated for a homogeneous system with impurities modeled by adding a noise term with $\sigma = 0.2$ to the constant conductivity and for an inhomogeneous system with $\lambda(z) = 2z$. Again, up to an overall shift, the scaling of the condition number follows the same exponential pattern. This is to be kept in mind while examining the results of the tests described below.

We applied the algorithm to the problem for various configurations of the quantum circuit, differing in number of qubits n and depth of the ansatz d , emulated using the Qiskit library. The minimization was carried out using optimization routines provided by the Scipy package and choosing the limited-memory, bounded Broyden–Fletcher–Goldfarb–Shanno (L-BFGS-B) method to search for the minima. The maximum circuit size tested was $n = 10$, resulting in a system of dimension $N = 2^{10} = 1024$ which can easily be solved numerically by standard linear algebra libraries. We used the numerical solution $|x\rangle$ as a quality check for our results, by measuring the infidelity of the variational solution $|x(1)\rangle$, defined as

$$\mathcal{I} := 1 - |\langle x(1)|x\rangle|^2. \quad (25)$$

The algorithm itself, on the other hand, works by minimizing the energy as expressed by the Hamiltonian $H(s)$,

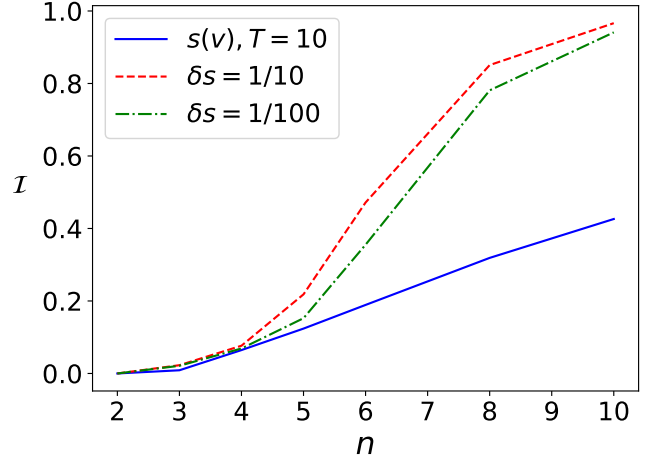


FIG. 6. Infidelity \mathcal{I} of different solutions for $\lambda = 1$, $\mathbf{b} = \mathbf{e}_1$, for all qubit numbers n considered, obtained by changing the adiabatic step δs . Note the very small improvement, at higher n , from a tenfold increase in resolution with a fixed δs (dashed red line and dot-dashed green line) and the substantial impact of adopting a dynamic approach (solid blue line).

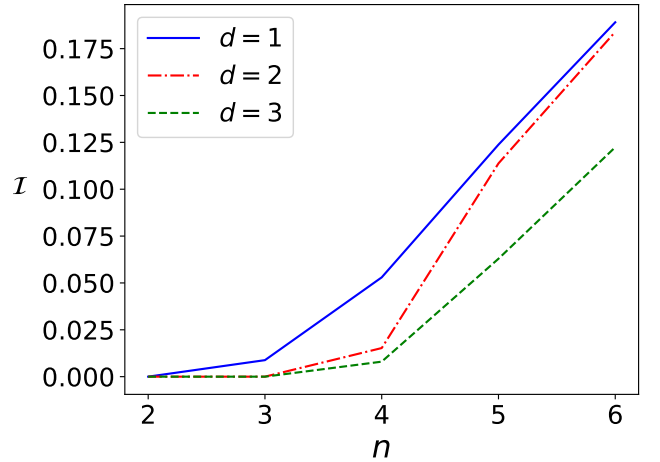


FIG. 7. Infidelity \mathcal{I} of the variational solution for $\lambda = 1$, $\mathbf{b} = \mathbf{e}_1$, computed with $s = s(v)$ for different circuit depths. When the infidelity is around zero ($n = 2$ and 3), the depth is large enough ($d = 1$ and 2 respectively) for the image of the variational circuit to cover the whole Hilbert space and the algorithm manages to reach the exact solution.

which for $s = 1$ amounts to maximizing the accuracy of the solution, defined as

$$\mathcal{A} := \frac{|\langle b|A|x(1)\rangle|^2}{\|A|x(1)\rangle\|^2}. \quad (26)$$

In general, the two figures of merit may exhibit varying degrees of correlation depending on the spectrum of $H(1)$.

Before looking into the actual results of the simulations, let us briefly consider the effect of our parametriza-

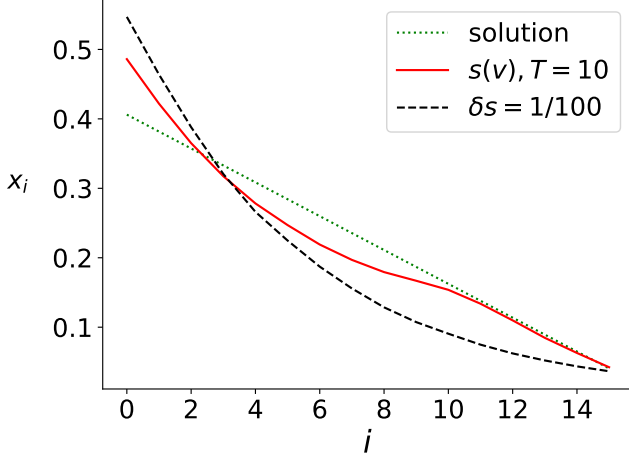


FIG. 8. Components x_i of exact and variational solutions for $\lambda = 1$, $\mathbf{b} = \mathbf{e}_1$, $n = 4$, $d = 2$: constant $\delta s = 1/T$ with and $T = 100$ (dashed black line), $s = s(v)$ with and $T = 10$ (solid red line), and exact solution (dotted green line). The sequence $s(v)$ improves the quality of the solution, coming closer to the exact $|x\rangle$.

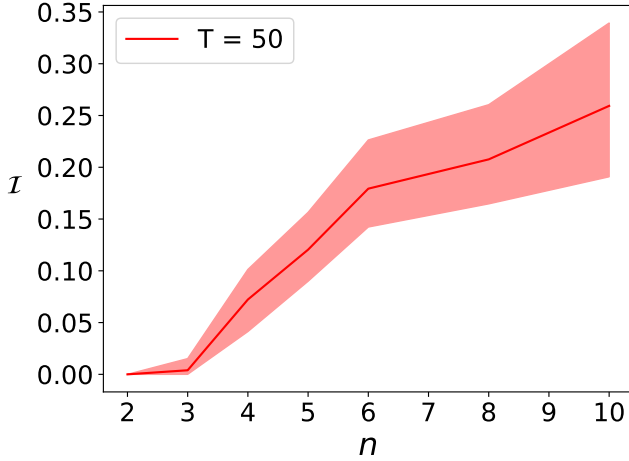


FIG. 9. Mean infidelity \mathcal{I} (solid line) and the range between minimum and maximum infidelity (shaded area) for different numbers of qubits n in the $\lambda = 1 + \zeta$ scenario with $\mathbf{b} = \mathbf{e}_1$, where 10 realizations of the noise ζ with variance $\sigma = 0.2$ were sampled.

tion choice on the quality of the solutions: Fig. 6 shows the infidelity \mathcal{I} plotted against n for the solutions with a point source $\mathbf{b} = \mathbf{e}_1$ obtained using $T = 10$ and 100 evenly spaced adiabatic steps and $T = 10$ dynamic steps as given by Eq. (14). The benefits of a problem-dependent step size can be clearly seen at $n > 4$, where it dramatically improves the quality of the solution, even in this simple example with minimal circuit depth $d = 1$.

Nevertheless, the quality is still unsatisfactory at higher n , even when using the dynamic step. This is explained by the number of variational parameters

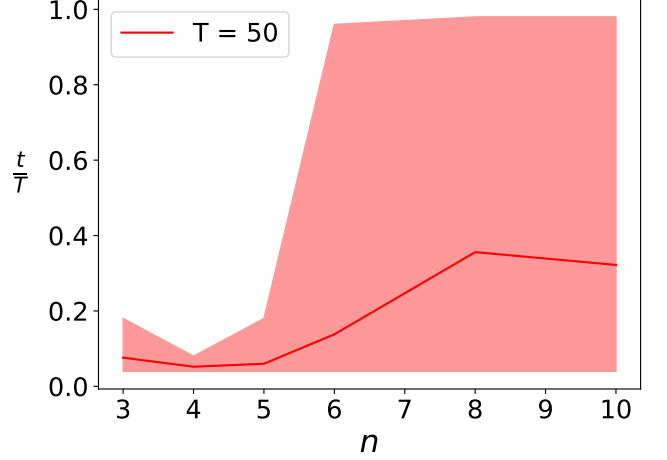


FIG. 10. Mean number of steps t (solid line) and the range between minimum and maximum number of steps (shaded area) for different n in the $\lambda = 1 + \zeta$ scenario with $\mathbf{b} = \mathbf{e}_1$. Although the maximum number of steps occurring in our simulations was $t = T = 50$, on average the algorithm performed a fraction $t/T \approx 0.4$.

$n_p = n(d + 1)$ encoding the solution in the quantum circuit: in general, the state of n qubits with only real gates acting on them is described by $N_p = 2^n - 1$ real numbers, far exceeding the expressibility of the ansatz. Fig. 7 shows how increasing d improves the quality of the solutions found for $n \leq 6$.

The effect of the dynamic adiabatic step is also evident if we look at the solution explicitly: Fig. 8 shows the vector components x_i of $|x\rangle$ and $|x(1)\rangle$ computed by two runs with $T = 100$, one with a fixed $\delta s = 1/T$ and the other using Eq. (14), for $n = 4$ and circuit depth $d = 2$. Again, the dynamic solution is noticeably closer to the exact one, although its overall quality is not excellent. For this result, the infidelity is $\mathcal{I} \approx 0.01$, but the accuracy is quite high ($\mathcal{A} \approx 0.9996$), signaling that the difference between $|x\rangle$ and $|x(1)\rangle$ does not translate to an appreciable difference after applying A . This is an example of how the condition number of A ($\kappa \approx 135$ in this case) may decouple the two quality measures of a solution.

With these general observations in mind, let us now move on to the next physical model and consider a homogeneous medium with impurities. Recall that in this case $\lambda_i = 1 + \zeta_i$, where ζ_i is a normally distributed random variable with standard deviation $\sigma = 0.2$, and the matrix A is computed from λ_i using Eq. (23). We used $T = 50$ in all our calculations, after a number of tests with $T = 10, 100$ showed no significant differences, and performed each run with 10 different seeds in order to average the noise. From now on, the circuit depth will be understood to be $d = 2$ unless otherwise specified.

Fig. 9 shows the effect of the noise on the quality of the solutions found for the point-source case: the results spread fairly evenly around a mean value similar to the

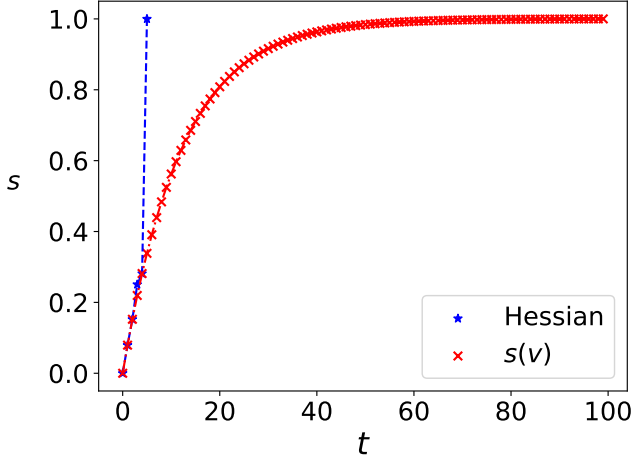


FIG. 11. An example of the advantage brought by the convexity analysis for the case with $\lambda = 1$, $n = 4$ and $T = 100$. The whole sequence $s(v)$ (red crosses) required in order to reach $s = 1$ is shown in red, but a step-by-step analysis of the Hessian allows the system to skip a potentially large fraction of the calculations at a marginal added cost (blue stars).

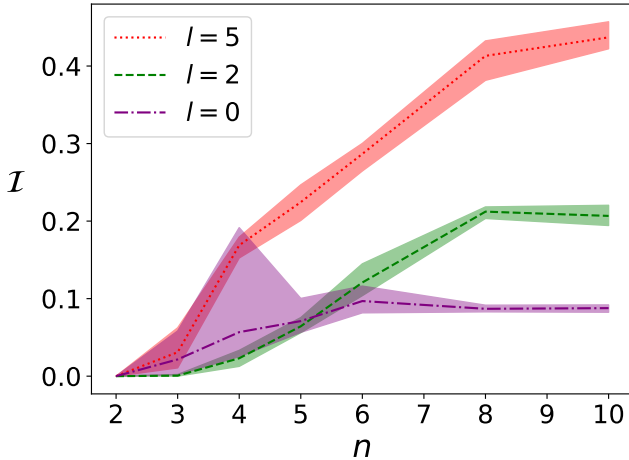


FIG. 12. Mean infidelity \mathcal{I} (solid line) and the range between minimum and maximum infidelity (shaded area) for different numbers of qubits n in the $\lambda = 1 + \zeta$ scenario with exponentially decreasing heat sources, $Q_r(z) = Q_0 e^{-lz/L}$, for $l = 0, 2$ and 5 . The plot is the result of 10 realizations for each n . Note that since we plot the maximum and minimum value reached during our simulations, the shaded area is not symmetric.

result of the noiseless case shown previously. Regarding the adiabatic parametrization, it is interesting to see how the introduction of the Hessian-based evaluation of the optimal δs , as described in Section II B, improves the efficiency of the algorithm. In Fig. 10, we plot the number of effective adiabatic steps t as a fraction of the full $T = 50$ sequence defined in Eq. (14): on average, we observed a reduction in the number of steps down to $t \approx 0.4T$

(even less for larger T , since the maximum δs compatible with the Hessian at any given s is independent of T). Note that the maximum number of steps tends to saturate the required number T as the number of qubit grows: this indicates that the algorithm will often resort to the default step suggested by Eq. (14), either because the point in parameter space is a saddle point and not a minimum (i.e. the Hessian has at least one negative eigenvalue) or because the δs suggested by the Hessian is smaller than the default step, so our implementation ignores it in order to avoid getting stuck in a potentially very long chain of impractically short increments. On the other hand, the minimum number of steps is always very close to zero: this means that in some cases the Hessian analysis yielded a negative δs , so the algorithm could skip all remaining intermediate steps and use the current parameter values as a warm start to solve the problem for $s = 1$ directly. An example of this is shown in Fig. 11.

To gain further insights into the performance of the method on this problem, we now examine the results for the family of exponential source terms \mathbf{b}_l defined by their components $b_{lj} = Q_0 e^{-jl\Delta z/L}$. As previously mentioned, we treat these cases by performing the Householder transformation Eq. (11) on the problem so that the quantum circuit is always initialized at $\theta_i = 0$, i.e. $|b\rangle = |0\rangle$, and all properties of the source term are encoded in the transformation applied to the matrix A .

In Fig. 12, we show the infidelity obtained by solving the cases with $l = 0, 2$ and 5 . Three features stand out and deserve some discussion. First, the quality of the solution consistently deteriorates as n increases, confirming that the circuit depth and expressibility plays the same role as in the case of a point source. Secondly, there seems to be a correlation between better solutions and a smaller spread in infidelity, indicating that if the circuit is capable of finding a good solution its quality is not significantly impacted by the noise in the problem. Finally, we observed an interesting trend correlated to l : lower values of l lead to better solutions, as shown by the decrease in infidelity even at higher qubit numbers. A hint towards a possible explanation for this pattern can be found by analyzing the $l = 0$ case. This particular solution vector exhibits a large overlap with the eigenstate $|a_0\rangle$ of A having the smallest eigenvalue, as illustrated in Fig. 13. Therefore, the solution is less susceptible to the condition number κ of the matrix than another state with a larger overlap with higher eigenstates, since small perturbations of the solution are not amplified into large differences in the source. This allows the optimization procedure to converge quite reliably towards the desired state, thus increasing the likelihood of achieving a high-fidelity result. It also suggests that the method is not necessarily doomed to fail whenever the condition number is very high, as long as the ansatz contains the exact solution (or at least a state very close to it) and the warm-start approach correctly tracks the minimum in the parameter space. The poorer quality of the solutions with $l = 2$ and

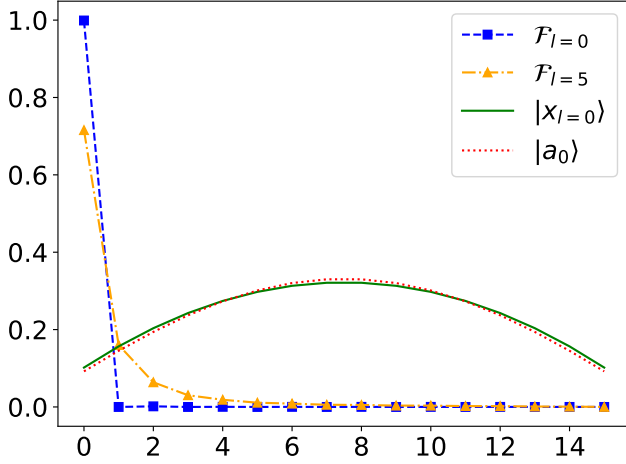


FIG. 13. The overlaps between the solution of the system of equations with $l = 0$ (blue squares) and $l = 5$ (orange triangles). The solid green line is the solution for $l = 0$, while the dashed red line is the eigenvector $|a_0\rangle$ of A with the lowest eigenvalue. The solutions for both $l = 0$ and $l = 5$ have a large overlap with the eigenvector $|a_0\rangle$, thus showing that the effective condition number is lower than the condition of the matrix A .

$l = 5$ did not come from the overlap of these states with $|a_0\rangle$, which is not significantly less than in the $l = 0$ case, but have probably more to do with circuit expressibility.

B. Case II: Linearly increasing conductivity

Let us now turn to the other system of physical interest mentioned at the beginning of the section: a model with increasing conductivity $\lambda(z) = cz$, with $c > 0$ a positive constant. This is a prototypical model used in geophysical science for a homogeneous vertical system, in which conductivity is proportional to pressure, and therefore to soil depth [30].

In this case, keeping open boundary conditions, the matrix A has components

$$A_{ij} = -2^{2-2n}i\delta_{ij} + 2^{-2n}i\left(\frac{3}{2}\delta_{i,j-1} + \frac{1}{2}\delta_{i,j+1}\right). \quad (27)$$

The condition number of A exhibits the same exponential scaling in n as in the constant-conductivity case already shown in Fig. 4. Unless otherwise specified, we will consider the medium to be not fully homogeneous by adding a noise term to the linearly increasing conductivity: $\lambda(z) = cz + \zeta$, with $c = 2$ and ζ normally distributed around 0. This time, we let $\sigma = 0.05$ for the noise, since higher variances often result in the matrix A not being positive and therefore physically descriptive.

Again, we first consider a localized heat source $\mathbf{b} = \mathbf{e}_1$ and will then move on to the exponential family defined in Eq. (24). The infidelity for the point-source scenario,

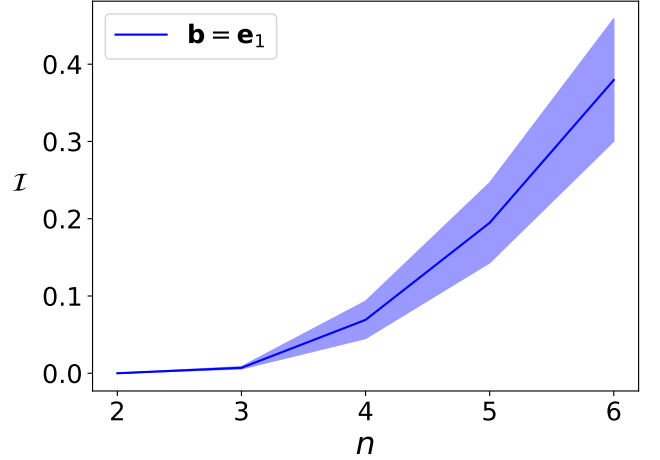


FIG. 14. Mean infidelity \mathcal{I} (continuous line) and the range between minimum and maximum infidelity (shaded area) for different qubit numbers n in the $\lambda(z) = 2z + \zeta$ scenario with a localized heat source $\mathbf{b} = \mathbf{e}_1$. ζ is a normally distributed random variable with mean value 0 and standard deviation $\sigma = 0.05$. The plot is the result of 10 realizations for each n .

plotted in Fig. 14, is generally higher than for the homogeneous case. Since the ground state in this case has a large component along $|a_0\rangle$, as we already saw in the constant-conductivity case, we can blame the larger infidelity on poor expressibility of the quantum circuit, which is evidently unable to provide the correct solution for this instance of the problem.

Switching to the exponential source terms, we find that the infidelity follows the same pattern as in the fully localized case. This happens regardless of the overlap between the solution and the first eigenstates of the matrix A , suggesting that the algorithm is once again struggling either to find its way towards the global minimum or to reach it given the constraints posed by the ansatz.

In order to examine the link between l and the quality of the solutions more thoroughly, we performed a series of runs of our algorithm letting l vary between 0 and 8 in steps of 0.5, removing the noise term from the conductivity and increasing the number of steps to $T = 100$. In Fig. 15, we show the results for $n = 5, 6$ and 8. As before, we consistently found lower infidelities for lower l , although with qualitatively different trends for different qubit numbers. A significant transition occurs near $l = 2$ for $n = 5$ and 6: the quality of the results is extremely poor for lower values of l but abruptly improves above some critical value, possibly due to a different and better local minimum suddenly appearing within reach of the algorithm. Examining the parameter optimization paths for $l = 1.5$ and $l = 2$, we noticed a qualitative difference between the two cases, which is shown in Fig. 16 for a subset of the qubit rotation angles: while for $l = 2$ the adiabatic evolution appears smooth, for $l = 1.5$ the parameters exhibit a piecewise evolution, with several jumps occurring as the algorithm progresses. This can be

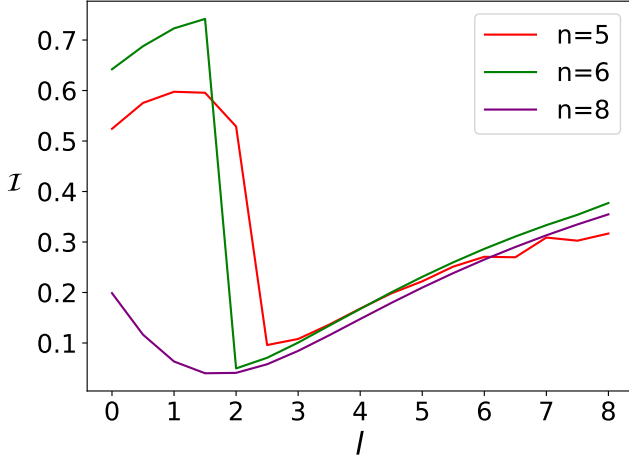


FIG. 15. Infidelity \mathcal{I} as a function of the exponent l of the heat source distribution $Q_r(z) = Q_0 e^{-lz/L}$, for $n = 5, 6$ and 8 and $T = 100$. When n is small (see $n = 5$ the algorithm tends to jump between near local minima and this affects the infidelity of the final solution, but this behavior is reduced as n increases.

interpreted as a sign of new minima appearing outside the path traced by evolving from the initial minimum, and has the important implication that the algorithm occasionally jumped from one minimum to another, crossing the barrier separating them. Aside from the poor quality of the result despite these efforts by the algorithm, it is worth remarking that such crossings between minima are possible in small- n applications such as our tests described here, but become less and less likely as the problem size increases. In fact, these barriers between independent pits and valleys in the cost function landscape can be overcome as long as the parameter space is not too large, but eventually develop into barren plateaus. In this light, it is not surprising that the result for $n = 8$ seen in Fig. 15 is a smooth curve: in this case, the solutions for different source terms \mathbf{b}_l all lie in the same region of the parameter space even for $l \leq 2$ and the corresponding infidelity, accuracy and cost function values neatly fall in line.

IV. DISCUSSION AND CONCLUSION

In this work, we proposed a modified version of the established variational quantum linear solver (VQLS) in an attempt to circumvent the notorious barren plateau (BP) problem, and used it to solve a linear differential equation in order to assess its performance. We focused our attention on the stationary solution of the heat flow equation, and applied our algorithm to several variants of the equation using different choices for the definition of the conductivity and heat source.

Our strategy to avoid the BP problem is twofold: instead of searching the whole parameter space up front,

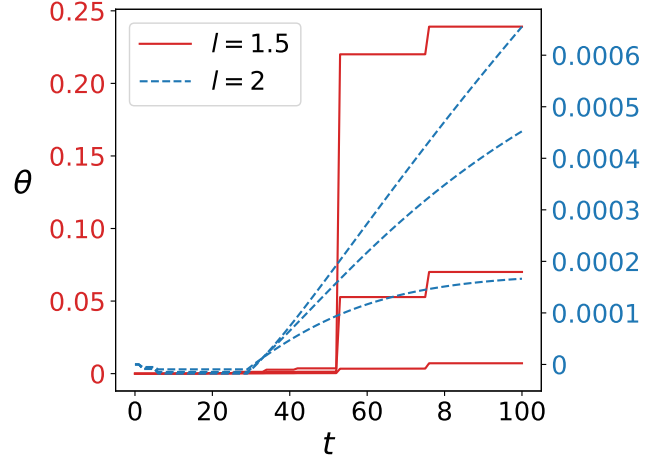


FIG. 16. Optimization paths of three selected parameters θ_i for the exponents $l = 1.5$ (solid red lines) and $l = 2$ (dashed blue lines) of the heat source distribution $Q_r(z) = Q_0 e^{-lz/L}$, for $n = 6$ qubits and up to $T = 100$ adiabatic steps. The angles plotted were chosen for no particular reason and the pattern is similar for all other θ_i . While for $l = 2$ the solution changes smoothly, indicating that the convexity test was successful in generating warm starts throughout the adiabatic evolution, for $l = 1.5$ the algorithm was not always able to find the correct path and occasionally jumped between independent local minima. This effect is reduced for larger n .

we define a parametrization interpolating between a trivial linear system and the system we want to solve, and we follow this interpolation in steps small enough for each solution to serve as a so-called warm start, i.e. a good guess from which to start searching for the next one. The initial convex problem, corresponding to the value $s = 0$ of the adiabatic parameter guiding the transition to the target problem, always has a well known and easily prepared solution (in our implementation, the state $|0\rangle$, corresponding to all ansatz parameters $\theta_i = 0$). Increasing s , the system and the cost function gradually change until at $s = 1$ the cost function is minimized by the state $|x\rangle$ solving the linear system given. The key is to carefully determine the size of each step δs so that the minimum of the cost function at $s + \delta s$ is easy to find starting from the one found at s . This is why starting from an obvious solution is essential in order for the method to work.

Our choice of δs changes dynamically as the algorithm progresses, in order to adapt both to the system to be solved and to the behavior of the cost function at each step. A series of T steps is initially created according to a rule based on the condition number κ of the matrix A , but at each iteration we perform an analysis of the convexity of the cost function and use this to decide whether to move on to the next step in this series or to increase s faster if it is convenient to do so.

We stress that the proposed method consistently performs better than directly optimizing the ansatz param-

eters for the final problem, in terms of quality of the results. This was confirmed directly by running our code with $T = 1$ and finding that both our accuracy and infidelity metrics turned out worse than in the adiabatic case. However, in order to determine whether our method can indeed be considered superior to the conventional VQLS, we also need to analyze its computational complexity, since the standard VQLS involves a single optimization of the parameters, whereas our approach requires optimizing the parameters at each adiabatic step. An advantage typically emerges for systems with a large number of qubits when, by increasing the maximum number of steps polynomially with n (i.e., $T \sim \mathcal{O}(n^\alpha)$ – a problem-dependent requirement), the optimization landscape near the global minimum maintains a convex structure, which ultimately simplifies the optimization procedure required at each step. In this case, the benefits outweigh the cost associated with repeated optimizations.

A. Computational complexity of the algorithm

To assess the overall cost of our algorithm and compare it to that of a conventional VQLS, it is useful to summarize it here and take a look at its parts separately. At each cycle of the adiabatic evolution, we perform the following operations:

- Compute the expectation values of the operators \hat{a}, \hat{b} and \hat{c} defined in Eq. (15) on the quantum circuit, both for the current minimum θ^* and for the shifted parameter values necessary to compute the Hessian $\mathcal{H}[C_s(\theta^*)]$ at the same point. The complexity of this operation is $\mathcal{O}(dnM)$, where M is the maximum number of operators in the Pauli decomposition of \hat{a}, \hat{b} and \hat{c} .
- If the Hessian has no negative eigenvalues, find δs such that the smallest eigenvalue of $\mathcal{H}[C_{s+\delta s}(\theta^*)]$ changes sign, using a classical, Newton-like root-finding method and standard diagonalization routines. The computational cost of this operation is $\mathcal{O}(\text{Poly}(dn) \log(\log(1/\epsilon)))$, where ϵ is the tolerance on δs .
- Increment s by δs and find the minimum of the cost function in the new landscape, using the previous minimum θ^* as a warm start. This part of the total cost depends on the particular optimization routine chosen, but it is generally quite efficient if the previous steps successfully identified a path preserving the convexity of the parameter space.

The cycle is repeated (at most) a set number of times T , which is arbitrary in principle but should be chosen with care, like the circuit depth d or the tolerance ϵ , considering the problem at hand. These hyperparameters concur in determining the performance of the algorithm as well

as its cost and there is no intuitive way to determine their optimal values for any given application up front.

The circuit depth and the form of the ansatz determine the parameter space and hence the set of accessible quantum states, establishing the maximum accuracy achievable by the algorithm for the problem given. The cost function landscape, in combination with the matrix A , in turn entails a maximum step size in order for the convexity to be preserved by the adiabatic step. This must be guessed by setting an appropriate number of steps T : while the optimal T is not obvious, test runs of the algorithm itself can provide some hints. On one extreme, a very low T will cause the cost function to change too much for the previous minimum to be of any use in the search for the new one, resulting in effectively independent searches with no real warm starts. Vice versa, very short steps suggested by setting a high value for T will most likely be overridden by the calculation of the Hessian, leading to a much lower effective cost than T times the cost per step. This means that the warm-start approach is successful in optimizing the adiabatic increments, and also that setting T to values higher than a certain problem-dependent limit will have no impact on the actual cost.

As a side note, one must keep in mind that the Pauli decomposition of an arbitrary n -qubit operator (which is performed only at the very beginning of our algorithm, not at every step) is generally not an efficient operation. This was not discussed here because it is also a problem for conventional VQLS algorithms, but it should be noted that neither method is recommended if this decomposition cannot be carried out using a polynomial number of elementary operators. Alternatively, one might try using classical shadows techniques [31, 32] to approximate the result, but the quality and performance of such methods can be very problem-dependent.

B. Open points and outlook

First of all, let us remark that in this paper we demonstrated our method only via emulator and up to a small number of qubits, $n = 10$, in order to compare its results to classical numerical solutions, which are easily accessible for such problem sizes. Any performance evaluation aimed at certifying some quantum advantage would require at least twice as many qubits and real quantum hardware and was beyond the scope of this work.

We have explored the flexibility of our adiabatic version of the VQLS in several ways, trying to change both the form of our example problem and the hyperparameters of the method, such as the circuit depth d and the number of steps T , in order to give a flavor of what kind of behavior to expect from it. The tests performed are by no means exhaustive, and we plan to keep exploring the potential of our algorithm both practically and theoretically, e.g. by looking for more rigorous criteria to link T to the specific problem given and avoid situations in

which the convexity analysis fails or makes the algorithm too inefficient.

Moreover, real-life implementations necessarily mean working with significantly larger number of parameters if the ansatz is to be sufficiently expressive for a solution to lie within its image. Like the Pauli decomposition problem, this difficulty affects any variational algorithm, but our combination of adiabatic evolution and warm starts can be very valuable in such cases since it prevents the optimization from getting lost in barren plateaus.

It remains to be seen, however, whether this advantage comes at an acceptable cost: it is possible that, depending on the application considered, the adiabatic steps necessary to remain within a convex region in parameter space throughout the evolution are so small that the method does not outperform a standard VQLS with barren plateaus after all. We approached this question only heuristically, but it is certainly worth analyzing with more rigor, for instance by trying to link the behavior of

the Hessian in the case of an adiabatically changing cost function landscape to the size of the parameter space, as done in Ref. [24] for the case of unitary evolution, or to the properties of the linear problem given.

Finally, we would like to remark that other areas of research, e.g. condensed matter physics or quantum chemistry, might provide much scope for testing hybrid quantum computing methods such as this variational algorithm on problems of current relevance and test the limits of such applications, including the potential advantages and limitations of present-day quantum hardware platforms.

ACKNOWLEDGMENTS

We acknowledge financial support from the Italian National Centre for HPC, Big Data and Quantum Computing (CN00000013).

-
- [1] P. Moin, *Fundamentals of Engineering Numerical Analysis*, 2nd ed. (Cambridge University Press, Cambridge, 2010).
 - [2] A. Cali, S. Succi, A. Cancelliere, R. Benzi, and M. Gramignani, Diffusion and hydrodynamic dispersion with the lattice boltzmann method, *Phys. Rev. A* **45**, 5771 (1992).
 - [3] W. Itani and S. Succi, Analysis of Carleman Linearization of Lattice Boltzmann, *Fluids* **7**, 24 (2022), number: 1 Publisher: Multidisciplinary Digital Publishing Institute.
 - [4] C. Sanavio, R. Scatamacchia, C. de Falco, and S. Succi, Three Carleman routes to the quantum simulation of classical fluids, *Physics of Fluids* **36**, 057143 (2024).
 - [5] C. Sanavio, E. Mauri, and S. Succi, Explicit Quantum Circuit for Simulating the Advection-Diffusion-Reaction Dynamics, *IEEE Transactions on Quantum Engineering*, 1 (2025), conference Name: IEEE Transactions on Quantum Engineering.
 - [6] M. H. Gutknecht, A Brief Introduction to Krylov Space Methods for Solving Linear Systems, in *Frontiers of Computational Science*, edited by Y. Kaneda, H. Kawamura, and M. Sasai (Springer Berlin Heidelberg, Berlin, Heidelberg, 2007) pp. 53–62.
 - [7] S. Succi, W. Itani, K. R. Sreenivasan, and R. Steijl, Quantum computing for fluids: Where do we stand?, *Europhysics Letters* 10.1209/0295-5075/acfd7 (2023).
 - [8] C. Sanavio, S. Succi, and R. Steijl, Quantum Computing for Simulation of Fluid Dynamics, in *Quantum Information Science - Recent Advances and Computational Science Applications* (IntechOpen, 2024).
 - [9] A. I. Kauerauf and T. Hantschel, *Fundamentals of Basin and Petroleum Systems Modeling*, 1st ed. (Springer Berlin Heidelberg, 2009).
 - [10] A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum algorithm for linear systems of equations, *Phys. Rev. Lett.* **103**, 150502 (2009).
 - [11] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, Exponential improvement in precision for simulating sparse hamiltonians, in *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '14 (Association for Computing Machinery, New York, NY, USA, 2014) p. 283–292.
 - [12] A. M. Childs, R. Kothari, and R. D. Somma, Quantum algorithm for systems of linear equations with exponentially improved dependence on precision, *SIAM Journal on Computing* **46**, 1920 (2017), number: 6 arXiv:1511.02306 [quant-ph].
 - [13] G. H. Low and I. L. Chuang, Optimal Hamiltonian Simulation by Quantum Signal Processing, *Physical Review Letters* **118**, 010501 (2017).
 - [14] E. Campbell, A series of fast-paced advances in Quantum Error Correction, *Nature Reviews Physics* **6**, 160 (2024), publisher: Nature Publishing Group.
 - [15] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, Variational quantum algorithms, *Nature Reviews Physics* **3**, 625 (2021).
 - [16] C. Bravo-Prieto, R. LaRose, M. Cerezo, Y. Subasi, L. Cincio, and P. J. Coles, Variational Quantum Linear Solver, *Quantum* **7**, 1188 (2023), publisher: Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften.
 - [17] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, *Nature* **549**, 195 (2017).
 - [18] M. Schuld and F. Petruccione, *Machine Learning with Quantum Computers*, Quantum Science and Technology (Springer International Publishing, Cham, 2021).
 - [19] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, Cost function dependent barren plateaus in shallow parametrized quantum circuits, *Nature Communications* **12**, 1791 (2021), number: 1 Publisher: Nature Publishing Group.
 - [20] M. Cerezo, G. Verdon, H.-Y. Huang, L. Cincio, and P. J. Coles, Challenges and opportunities in quantum machine learning, *Nature Computational Science* **2**, 567 (2022).
 - [21] M. Larocca, P. Czarnik, K. Sharma, G. Muraleedharan,

- P. J. Coles, and M. Cerezo, Diagnosing Barren Plateaus with Tools from Quantum Optimal Control, *Quantum* **6**, 824 (2022).
- [22] M. Cerezo, M. Larocca, D. García-Martín, N. L. Diaz, P. Braccia, E. Fontana, M. S. Rudolph, P. Bermejo, A. Ijaz, S. Thanasilp, E. R. Anschuetz, and Z. Holmes, Does provable absence of barren plateaus imply classical simulability? Or, why we need to rethink variational quantum computing (2024), arXiv:2312.09121 [quant-ph].
- [23] Y. Subasi, R. D. Somma, and D. Orsucci, Quantum Algorithms for Systems of Linear Equations Inspired by Adiabatic Quantum Computing, *Physical Review Letters* **122**, 060504 (2019).
- [24] R. Puig, M. Drudis, S. Thanasilp, and Z. Holmes, Variational Quantum Simulation: A Case Study for Understanding Warm Starts, *PRX Quantum* **6**, 010317 (2025), publisher: American Physical Society.
- [25] S. Sim, P. D. Johnson, and A. Aspuru-Guzik, Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms, *Advanced Quantum Technologies* **2**, 1900070 (2019).
- [26] A. S. Householder, Unitary Triangularization of a Non-symmetric Matrix, *J. ACM* **5**, 339 (1958).
- [27] H. Mhiri, R. Puig, S. Lerch, M. S. Rudolph, T. Chotibut, S. Thanasilp, and Z. Holmes, A unifying account of warm start guarantees for patches of quantum landscapes (2025), arXiv:2502.07889 [quant-ph].
- [28] A. Mari, T. R. Bromley, and N. Killoran, Estimating the gradient and higher-order derivatives on quantum hardware, *Phys. Rev. A* **103**, 012405 (2021).
- [29] Y. Ma, J. Wang, F. Hu, E. Yan, Y. Zhang, H. Deng, X. Gao, J. Kang, H. Shi, X. Zhang, J. Zheng, and J. Guo, A case study on thermal conductivity characteristics and prediction of rock and soil mass at a proposed ground source heat pump (GSHP) site, *Scientific Reports* **15**, 8125 (2025), publisher: Nature Publishing Group.
- [30] V. Pasquale, M. Verdoya, and P. Chiozzi, Heat flow and geothermal resources in northern Italy, *Renewable and Sustainable Energy Reviews* **36**, 277 (2014).
- [31] G. Boyd and B. Koczor, Training Variational Quantum Circuits with CoVaR: Covariance Root Finding with Classical Shadows, *Physical Review X* **12**, 041022 (2022).
- [32] H.-Y. Huang, R. Kueng, and J. Preskill, Predicting many properties of a quantum system from very few measurements, *Nature Physics* **16**, 1050 (2020), publisher: Nature Publishing Group.