

# Notes on Quantum Computing for Thermal Science

Pietro Asinari<sup>a,b</sup>, Nada Alghamdi<sup>a</sup>, Paolo De Angelis<sup>a</sup>, Giulio Barletta<sup>a</sup>, Giovanni Trezza<sup>a,c</sup>,  
Marina Provenzano<sup>a</sup>, Matteo Maria Piredda<sup>a</sup>, Matteo Fasano<sup>a</sup>, Eliodoro Chiavazzo<sup>a,b</sup>

<sup>a</sup>*Dipartimento Energia, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129, Torino (TO), Italy*

<sup>b</sup>*Istituto Nazionale di Ricerca Metrologica, Strada delle Cacce 91, 10135, Torino (TO), Italy*

<sup>c</sup>*Université Grenoble Alpes, 1130 Rue de la Piscine, St Martin D'Heres, 38402, France*

---

## Abstract

This document explores the potential of quantum computing for solving linear systems of interest in engineering. In particular, we focus on heat conduction as a paradigmatic example in thermal science. Conceived as a living document, it will be continuously updated with experimental findings and insights for the research community in Thermal Science. By experiments, we refer both to the search for the most effective algorithms and to the performance of real quantum hardware. Those fields are currently evolving rapidly, driving a technological race to define the best architectures. The development of novel algorithms for engineering problems aims at harnessing the unique strengths of quantum computing. Expectations are high, as users seek concrete evidence of quantum supremacy – a true game changer for engineering applications. Among all heat transfer mechanisms (conduction, convection, radiation), we start with conduction as a paradigmatic test case in the field being characterized by a rich mathematical foundation for our investigations.

*Keywords:* quantum computing, thermal science, heat conduction, variational quantum algorithm, quantum circuit ansatz, real data loading / encoding

---



---

\*Corresponding author

Email address: [pietro.asinari@polito.it](mailto:pietro.asinari@polito.it) (Pietro Asinari)

## Contents (version v01.18)

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Philosophical remark . . . . .	4
<b>2</b>	<b>Heat conduction equation</b>	<b>5</b>
2.1	Discrete Fourier Transform (FT) . . . . .	6
<b>3</b>	<b>Variational Quantum Eigensolver (VQE)</b>	<b>9</b>
3.1	Quantum data structure . . . . .	10
3.2	Normalization . . . . .	17
3.3	Observable . . . . .	18
3.4	Quantum circuit ansatz . . . . .	19
3.5	Optimization . . . . .	20
3.6	De-normalization . . . . .	20
3.7	Practical details of implementation . . . . .	21
3.7.1	Decomposition in Pauli matrices . . . . .	21
3.7.2	Efficient circuit ansatz . . . . .	23
3.7.3	Minimization of the loss function . . . . .	25
3.8	Simulated results . . . . .	25
3.8.1	Simulated results by Qiskit . . . . .	25
3.8.2	Simulated results by Qrisp . . . . .	25
3.9	VQE based on diagonalizing the measurement . . . . .	26
3.9.1	Quantum Fourier Transform (QFT) . . . . .	28
3.9.2	Hadamard test approach . . . . .	29
<b>4</b>	<b>Conclusions</b>	<b>35</b>
<b>Appendix A</b>	<b>Two-athlete strategy</b>	<b>36</b>
<b>Appendix B</b>	<b>Prisoner's dilemma</b>	<b>38</b>
<b>Appendix C</b>	<b>Hilbert space versus Bloch sphere</b>	<b>39</b>
<b>Appendix D</b>	<b>Real data loading/encoding</b>	<b>41</b>
<b>Appendix E</b>	<b>How discrete FT works</b>	<b>44</b>
<b>Appendix F</b>	<b>Example codes for VQE</b>	<b>48</b>

## 1. Introduction

Quantum computing could transform fields like computational science and engineering with possibly strong impact on material science, renewable energy and even finance by revolutionizing data processing. The ambitious goal is quantum advantage possibly outperforming classical computers in specific tasks. Here, in order to overcome this challenge, we focus on solving the heat conduction equation numerically as a paradigmatic application of quantum computing within the heat transfer and thermal science community. In most of the engineering applications, the computational domain - where heat conduction occurs - is discretized by a spatial mesh with  $N$  nodes. State-of-the-art CFD simulations can utilize up to 780 billion mesh cells on advanced supercomputers. For example, a study documented the use of a grid with 780 billion cells ( $N \sim 7.8 \times 10^{11}$ ) on Tianhe-2, leveraging over 1.376 million heterogeneous cores [1]. This is where – potentially – a quantum advantage could become interesting.

To better understand the potential quantum advantage, let us first recall how a classical computer processes real numbers. A common approach is to represent them in scientific notation as  $\mu \times 10^\nu$ , where  $1 \leq \mu < 10$  is the significand (or mantissa) and  $\nu$  is the exponent. The precision by which a classical computer can store a real number depends on the number of bits available for encoding the mantissa. To illustrate this, consider a simplified scenario where the computer has only three bits to store the mantissa. With  $N = 2^3 = 8$  possible binary configurations, the mantissa must be approximated to the closest available value within the range  $\mu \in [1, 10[$ . A reasonable discretization scheme is  $\mu \in \{\mu_0, \mu_1, \dots, \mu_{N-1}\}$ , where the generic significand can be expressed as  $\mu_b = 1 + (10 - 1)b/2^3$  and  $b$  is an integer from 0 to  $2^3 - 1$  included. The difference between two consecutive significands, namely the precision, is  $\Delta\mu = \mu_{b+1} - \mu_b = (10 - 1)/2^3 = 1.125$ . Naturally, real-world classical computers operate with far greater precision. A widely used standard for representing real numbers is the IEEE 754 double-precision floating-point format, which allocates 64 bits per number. Within this format, 53 bits are dedicated to encoding the mantissa, leading to a much finer discretization  $(10 - 1)/2^{53} \approx 1 \times 10^{-15}$ . This implies that, in a classical system, 53 bits are used solely to encode the significand of a single real number. More generally, given  $n_{\text{classic}}$  available classical bits, the number of real numbers that can be stored (with double-precision) is given by

$$N_{\text{classic}} = \lfloor n_{\text{classic}}/64 \rfloor, \quad (1)$$

where  $\lfloor \cdot \rfloor$  denotes the floor function, which returns the largest integer less than or equal to its argument. This constraint on classical storage is a key limitation that quantum computing aims to overcome.

The key distinction lies in the following fundamental property: a quantum system with  $n$  qubits has

$$N = 2^n \quad (\gg n \text{ typically}) \quad (2)$$

computational basis states, similar to a classical system. However – and this is the crucial difference – a quantum system can exist in a superposition of all these basis states, with each state weighted by a complex probability amplitude. These amplitudes, which may be

continuous real (or complex) numbers, determine the probability of measuring each basis state upon observation. As a result, a quantum computer can encode and manipulate a number of real values that is at least proportional to the number of computational basis states. This exponential scaling in the number of quantum states provides a potential advantage over classical systems. However, the actual precision of stored information is constrained by interactions with the environment, which cause decoherence, i.e. a process that disrupts quantum superpositions and limits computational performance. Moreover, there is a limit in the measurement resolution due to the actual hardware. To fix ideas, let us consider the following example: In November 2024, IBM released the IBM Heron R2 processor with  $n = 156$  qubits, which could – in principle – accommodate  $N = 2^{156} \sim 10^{46}$  field values on a mesh. On the other hand, a classical computer with the same number of classical bits  $n_{\text{classic}} = 156$  can store at most  $N_{\text{classic}} = 2$  real field values in double-precision. Hence it is clear that quantum computing may pave the way to significantly larger meshes than those currently used.

However, there is a problem. Currently, we are in the noisy intermediate-scale quantum (NISQ) era, with processors of up to 1,000 non-fault-tolerant qubits. Overcoming noise and decoherence remains a significant challenge, making it crucial to align quantum hardware advancements with specific application needs. In reality, the current NISQ computers are still rarely advantageous over classical computers for most of applications, which therefore must be investigated individually. Let us focus here on solving the heat conduction equation.

### 1.1. Philosophical remark

Before proceeding further, it is worth first discussing how an ideally reversible quantum computer can model an irreversible phenomenon. To clarify this point, let us consider an ideal quantum computer as a specific example of a generic quantum system. Among all possible quantum systems, a particularly illustrative one is the Schrödinger equation for a free particle in one dimension:  $i \hbar \partial_t \Psi = \hat{p}/(2m) \Psi = -\hbar^2/(2m) \partial_z^2 \Psi$  or equivalently  $\partial_t \Psi = i \hbar/(2m) \partial_z^2 \Psi$ , where  $\Psi$  represents the wavefunction of the quantum system. This equation is classified as dispersive, meaning it supports wave-like solutions with frequency-dependent phase velocities. Its purely imaginary time evolution results in phase oscillations without any decay. In other words, the Schrödinger equation is not dissipative – it describes reversible, wave-like behavior rather than an irreversible process. By contrast, the heat conduction equation, which we aim to model here, namely  $\partial_t T = D \partial_z^2 T$ , is inherently dissipative.

In an ideal quantum computer, where there is no interaction with the environment, state evolution is unitary and therefore reversible. The only source of irreversibility in such a system is measurement.

The key idea, therefore, is to construct a reversible quantum evolution that, upon measurement, collapses onto the target dissipative dynamics. The measurement in quantum mechanics consists of three ingredients: (i) the state, (ii) the observable and (iii) its expectation value. The state of a system is represented by its wavefunction  $\Psi$ , which contains all

the information about the system. The observable refers to any physical quantity that can be measured, such as position, momentum, or energy, and it is represented by an operator  $\hat{O}$ . The expectation value of an observable is the average result one would obtain from many measurements of that observable, and it is calculated by taking the inner product  $\langle \Psi | \hat{O} | \Psi \rangle$ , where  $\hat{O}$  is the operator corresponding to the observable. When an observable is measured, the system's wavefunction collapses to one of the eigenstates of the corresponding operator, and the measurement result will be one of the associated eigenvalues. The expectation value is the weighted average of these eigenvalues, with the weights being the probabilities of the system being found in each eigenstate. Hence the measurement process in quantum mechanics is often used as a way to model irreversible phenomena.

## 2. Heat conduction equation

Let us consider the one-dimensional heat conduction equation, as a paradigmatic application to the heat transfer community, namely

$$\frac{\partial T}{\partial t} = D \frac{\partial^2 T}{\partial z^2}, \quad (3)$$

with the function  $T = T(z, t)$  being the local temperature, and the positive coefficient  $D$  the thermal diffusivity of the medium. Let us consider a constant diffusivity, a given initial profile  $T(z, 0)$  and the periodic spatial boundary condition. This problem can be solved analytically using the Fourier transform and it is usually trivial for most of the current classical numerical techniques.

Let us solve the previous equation by the classical finite-difference (FD) method, which consists in solving differential equations by approximating derivatives with finite differences. Both the spatial domain and time domain are discretized by a regular mesh: the unknown function  $T$  is evaluated at the generic  $l$ -th mesh node and at the  $\tau$ -th time step, namely  $T_l^\tau = T(z_l, t_\tau)$  where  $z_l = l \Delta z$  with  $0 \leq l \leq (N - 1)$  and  $t_\tau = \tau \Delta t$  with  $0 \leq \tau \leq N_t$  ( $\tau = 0$  identifies the given initial profile). The quantities  $\Delta z$  and  $\Delta t$  are the spatial and temporal partitions of the grid, while  $N$  is the number of space mesh nodes and  $N_t$  is the number of time steps. Let us use a fully implicit FD scheme that yields the stability of the solution for arbitrary diffusivity of the equation and the grid size:

$$\frac{T_l^{\tau+1} - T_l^\tau}{\Delta t} = D \frac{T_{l-1}^{\tau+1} - 2 T_l^{\tau+1} + T_{l+1}^{\tau+1}}{\Delta z^2}. \quad (4)$$

The previous formula can be reformulated as

$$-r T_{l-1}^{\tau+1} + (1 + 2r) T_l^{\tau+1} - r T_{l+1}^{\tau+1} = T_l^\tau, \quad (5)$$

where  $r = D \Delta t / \Delta z^2$  is the (dimensionless) numerical Fourier number. Let us define a new

operator  $\hat{C}$  as

$$\hat{C} = \begin{bmatrix} (1+2r) & -r & 0 & 0 & \dots & -r \\ -r & (1+2r) & -r & 0 & \dots & 0 \\ 0 & -r & (1+2r) & -r & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & -r & (1+2r) & -r & 0 \\ 0 & \dots & 0 & -r & (1+2r) & -r \\ -r & \dots & 0 & 0 & -r & (1+2r) \end{bmatrix}, \quad (6)$$

which can be used to formulate a linear system of equations which is consistent with Eq. (4). In particular, using the operator  $\hat{C}$  defined by Eq. (6), Eq. (4) becomes in compact form

$$\hat{C} \vec{T}^+ = \vec{T}, \quad (7)$$

where  $\vec{T}^+$  stands for  $\vec{T}^{\tau+1} = (T_0^{\tau+1}, T_1^{\tau+1}, T_2^{\tau+1}, \dots, T_{N-1}^{\tau+1})^T$  and  $\vec{T}$  stands for  $\vec{T}^\tau = (T_0^\tau, T_1^\tau, T_2^\tau, \dots, T_{N-1}^\tau)^T$ . Clearly the inverse of operator  $\hat{C}$  can be used as a time-progress operator for the temperature profile subject to heat conduction, namely

$$\vec{T}^+ = \vec{T}(t + \Delta t) = \hat{C}^{-1} \vec{T}, \quad (8)$$

which can be also generalized by the following formula

$$\vec{T}(t + \tau \Delta t) = (\hat{C}^{-1})^\tau \vec{T}. \quad (9)$$

In the following section, for the sake of simplicity and without loss of generality, we will focus on Eq. (8) only.

### 2.1. Discrete Fourier Transform (FT)

The one-dimensional heat conduction problem defined by Eq. (7) can be solved numerically by means of the direct method given by Eq. (8), which requires to invert the operator  $\hat{C}$  of Eq. (6). There are also other alternative methods for special cases, which involve some transformations. When discretizing the heat equation using finite differences, the resulting matrix  $\hat{C}$  is diagonalizable by the Fourier transform only if the problem involves a periodic domain, leading to a circulant matrix structure, as in the present one-dimensional case. In this case, the discrete Fourier Transform (FT) efficiently diagonalizes  $\hat{C}$ , which seems quite natural also in the context of quantum computing [2]. For non-periodic boundary conditions, such as Dirichlet or Neumann,  $\hat{C}$  becomes a standard tridiagonal (non-circulant) matrix; here, the discrete Sine Transform (ST) and discrete Cosine Transform (CT) serve as the appropriate diagonalizing tools, matching the boundary constraints (Dirichlet for ST, Neumann for CT). In more complex cases – such as variable coefficients, irregular domains, or non-uniform grids – no standard transform diagonalizes  $\hat{C}$ , and numerical methods like eigen-decomposition or iterative solvers are typically used instead.

In the present one-dimensional example, because the mesh is regular and the domain is periodic, the discrete FT efficiently diagonalizes  $\hat{C}$ . In the usual mathematical notation,

the discrete FT takes as input the column vector  $\vec{T}$ , which can be defined by a proper orthonormal basis  $\vec{e}_0, \vec{e}_1, \vec{e}_2, \dots, \vec{e}_{N-1}$ , namely

$$\vec{T} = \sum_{l=0}^{N-1} T_l \vec{e}_l, \quad (10)$$

where  $T_l$  is the nodal value for the  $l$ -th mesh node and  $\|\vec{e}_l\| = 1$ . The discrete FT outputs the transformed data, a column vector of complex numbers  $\vec{\tilde{T}}$  defined in the same orthonormal basis, namely

$$\vec{\tilde{T}} = \sum_{k=0}^{N-1} \tilde{T}_k \vec{e}_k. \quad (11)$$

Please note that using the subscript  $k$  instead of  $l$  in the previous expression is unessential because the nodes are the same: it is just a matter of convention for making more evident the meaning of this sum. Each component of the transformed data is defined as<sup>1</sup>

$$\tilde{T}_k = \frac{1}{\sqrt{N}} \sum_{l=0}^{N-1} T_l \omega_N^{kl}, \quad (12)$$

where

$$\omega_N = e^{i2\pi/N}, \quad (13)$$

and the parameter  $i$  is the usual imaginary unit ( $i = \sqrt{-1}$ ). Please note that the factor  $1/\sqrt{N}$  in front of Eq. (12) is chosen to realize a unitary transformation by construction, which allows to implement this transformation by a unitary quantum circuit [2]. Moreover, the positive sign of the argument of the exponent of Eq. (13) is quite common in the quantum community and it implies an anti-clockwise rotation in the complex plane (Argand plane). See Appendix E for details about the physical meaning of the discrete FT. It is also useful to compute the wavenumber spectrum by the transformed field  $\vec{\tilde{T}}$ , which describes how the variance of the temperature field is distributed over different harmonic components. In case of a classical field, the wavenumber spectrum  $\vec{p}^c$  is defined as

$$\vec{p}^c = \frac{1}{N} \vec{\tilde{T}} \odot \vec{\tilde{T}}^*, \quad (14)$$

where  $\odot$  represents the Hadamard (element-wise) product and the superscript  $*$  means the complex conjugate. Another useful concept is the inverse transform, which is given by:

$$T_l = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \tilde{T}_k \omega_N^{-kl}. \quad (15)$$

---

<sup>1</sup>In some numerical routines, e.g. the “*scipy.fft*” function of the SciPy platform [3], the standard Fourier transform is defined with regards to  $e^{-i2\pi/N}$  and without the prefactor  $1/\sqrt{N}$ . It is possible to convert the results based on the standard form to those in the present document, by (i) multiplying them by  $1/\sqrt{N}$  and (ii) taking the complex conjugate.

The previous definition can be interpreted as a decomposition of the original field in Fourier modes, i.e. rotations in the complex plane with wavenumber  $k$  from 0 to  $N - 1$ .

At this point, it is possible to introduce a matrix notation, which is more convenient for solving the linear system of equations for heat conduction. Let us introduce the FT operator  $\hat{U}_{FT}$ , where the generic component at the  $k$ -th row and at the  $l$ -th column is given by  $1/\sqrt{N} \omega_N^{kl}$ , namely

$$\hat{U}_{FT} = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & \omega_N & \omega_N^2 & \omega_N^3 & \dots & \omega_N^{N-1} \\ 1 & \omega_N^2 & \omega_N^4 & \omega_N^6 & \dots & \omega_N^{2(N-1)} \\ 1 & \omega_N^3 & \omega_N^6 & \omega_N^9 & \dots & \omega_N^{3(N-1)} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & \omega_N^{N-1} & \omega_N^{2(N-1)} & \omega_N^{3(N-1)} & \dots & \omega_N^{(N-1)(N-1)} \end{bmatrix}. \quad (16)$$

In this way, the vector  $\vec{T}$  given by Eq. (11) and Fourier coefficients given by Eq. (12) can be expressed as

$$\vec{T} = \hat{U}_{FT} \vec{T}. \quad (17)$$

It is worth to highlight that the adopted definition of the matrix  $\hat{U}_{FT}$  makes it a unitary transformation, i.e.  $\hat{U}_{FT}^\dagger \hat{U}_{FT} = \hat{U}_{FT} \hat{U}_{FT}^\dagger = I$ , where  $\hat{U}_{FT}^\dagger$  denotes the conjugate transpose of  $\hat{U}_{FT}$ , (namely Hermitian transpose). The latter transpose is relatively simple to be computed and it allows one to express the initial temperature profile of the heat conduction problem as

$$\vec{T} = \hat{U}_{FT}^\dagger \vec{T}, \quad (18)$$

Introducing the previous definition into Eq. (7) yields

$$\hat{D} \vec{T}^+ = \vec{T}, \quad (19)$$

where  $\hat{D} = \hat{U}_{FT} \hat{C} \hat{U}_{FT}^\dagger$  is a diagonal operator and the elements on the diagonal are the eigenvalues of the operator  $\hat{C}$ . In order to find these eigenvalues, let us apply the definition given by Eq. (12) to the finite-difference formula given by Eq. (5) and let us use the same nomenclature adopted in Eq. (7), namely

$$-r \mathcal{F}(T_{l-1}^+) + (1 + 2r) \tilde{T}_k^+ - r \mathcal{F}(T_{l+1}^+) = \tilde{T}_k, \quad (20)$$

where  $\mathcal{F}(\cdot)$  means the linear transform defined by Eq. (12), i.e.

$$\mathcal{F}(T_{l-1}^+) = \mathcal{F}_k^W = \frac{1}{\sqrt{N}} \sum_{l=0}^{N-1} T_{(l-1) \bmod N} \omega_N^{kl}, \quad (21)$$

$$\mathcal{F}(T_{l+1}^+) = \mathcal{F}_k^E = \frac{1}{\sqrt{N}} \sum_{l=0}^{N-1} T_{(l+1) \bmod N} \omega_N^{kl}, \quad (22)$$



where  $\text{mod}$  is the modulo operation, which returns the remainder of a division. It is important to highlight that this  $\text{mod}$  operation is essential because the adopted labeling based on  $l$  goes only from 0 to  $N - 1$ . Simplifying  $\mathcal{F}_k^W$  yields

$$\mathcal{F}_k^W = \frac{1}{\sqrt{N}} \sum_{l'=0}^{N-1} T_{l'} \omega_N^{kl'+k} = \omega_N^k \tilde{T}_k, \quad (23)$$

where we set  $l' = (l - 1) \text{ mod } N$  which implies  $l = (l' + 1) \text{ mod } N$ . Proceeding similarly for  $\mathcal{F}_k^E$  and substituting these results in Eq. (20) yields

$$[1 + 2r - r\omega_N^k - r\omega_N^{-k}] \tilde{T}_k^+ = \tilde{T}_k. \quad (24)$$

Recalling the Euler's formula yields

$$\left[1 + 2r - 2r \cos\left(\frac{2\pi k}{N}\right)\right] \tilde{T}_k^+ = \tilde{T}_k. \quad (25)$$

and consequently

$$\left[1 + 4r \sin^2\left(\frac{\pi k}{N}\right)\right] \tilde{T}_k^+ = \tilde{T}_k. \quad (26)$$

Comparing Eq. (26) with Eq. (19), it is clear that  $\hat{D}$  is a diagonal matrix with the diagonal elements equal to

$$\hat{D}_{kk} = 1 + 4r \sin^2\left(\frac{\pi k}{N}\right). \quad (27)$$

It is easy to compute the inverse matrix by replacing the main diagonal elements of the matrix  $\hat{D}$  with their reciprocals, namely  $\hat{R} = \hat{D}^{-1}$ . The latter can be used to express the solution of the heat conduction problem as

$$\vec{T}^+ = \hat{U}_{FT}^\dagger \hat{R} \hat{U}_{FT} \vec{T}, \quad (28)$$

which is an alternative route to Eq. (8).

### 3. Variational Quantum Eigensolver (VQE)

Quantum computing is intimately connected with quantum information and deeply rooted in quantum physics. The rapid rate of progress in this field and its cross-disciplinary nature have made it difficult for newcomers to obtain a broad overview of the most important techniques and results [2]. There is still a lot of work to do in hardware and software development before demonstrating any quantum computing supremacy.

Here we focus on solving a linear system of equations as a paradigmatic task for many engineering problems. There are many computational strategies for solving a linear system of equations by quantum computing, which are still an active field of research. Let us start with the variational quantum eigensolver (VQE). VQE is a hybrid algorithm that uses both classical operations and quantum operations to find the ground state (i.e. the

stationary state of lowest energy) of a quantum system, which is designed to produce relevant information for the original problem of interest. In our case, the first step is to design a quantum system which allows one to derive the solution of the linear system of equations of interest, i.e. Eq. (7). Variational quantum algorithms are promising candidates for observing quantum computation utility on noisy near-term devices. For this reason, VQE is implemented in most of the open-source software development kit, e.g. Qiskit (Quantum Information Software Kit) by IBM [4].

### 3.1. Quantum data structure

First of all, before familiarizing with the data structure of a quantum computer, let us recall some basics of binary coding, which is useful for both classical and quantum computing systems. To represent a number in binary, every digit has to be either 0 or 1 (as opposed to decimal numbers where a digit can take any value from 0 to 9). A decimal integer, e.g. the already-mentioned index  $l$  which identifies the mesh node, can be decomposed in terms of a binary number, which is represented by an equivalent bit string, namely

$$l = \sum_{b=0}^{n-1} \beta_b 2^{n-1-b}. \quad (29)$$

where  $\beta_b \in \{0, 1\}$  is the value of the  $b$ -th computational unit in a binary computational system with  $n$  (qu)bits. For example, the decimal number 6, sometimes indicated as  $6_{10}$  in order to emphasize the basis number equal to 10, corresponds to the binary number 110, or even better  $6_{10} = 110_2$ . This means that each decimal integer  $d$  can be converted into an equivalent sequence of bits  $\beta_b$  in the binary numeral system, i.e.  $\beta_0\beta_1\dots\beta_{n-1}$ . Note that the digital bits  $\beta_b$  are listed according to the big-endian convention [2]. In the following, we will use equivalently the decimal integer  $l$  or the corresponding bit string  $\beta_0\beta_1\dots\beta_{n-1}$ .

Let us move now to the data structure of a quantum computer. The building block of a quantum computer is a qubit. A qubit is a two-level quantum-mechanical system, with many states which are linear combinations (often called superpositions) of the fundamental basis states, corresponding to the states 0 and 1 for a classical bit. Let us indicate the quantum states by the Dirac notation  $|\cdot\rangle$ , which stands for the standard notation for normalized vectors in quantum mechanics. Consequently the computational basis states of each qubit, or simply the computational basis, are  $|0\rangle$  and  $|1\rangle$ . The state vector for the generic  $q$ -th qubit in a system can be expressed as

$$|\psi_q\rangle = \delta_q^{[0]} |0\rangle + \delta_q^{[1]} |1\rangle, \quad (30)$$

where  $\delta_q^{[0]} \in \mathbb{C}$  and  $\delta_q^{[1]} \in \mathbb{C}$  are complex numbers that represent the weight of  $|0\rangle$  and  $|1\rangle$  states of the superposition, and are called complex probability amplitudes. In principle, these two complex numbers may suggest that there are four degrees of freedom in each qubit, but there are also two physical constraints to be considered. First of all, if these complex numbers are presented in polar form, it is possible to realize that their global phases can be disregarded, because only the relative phase matters with regards to the expectation value

of any observable [2]. Secondly, the corresponding probabilities are normalized such that  $(\delta_q^{[0]})^2 + (\delta_q^{[1]})^2 = 1$  [2]. Taking into account these two constraints, the state of each qubit can be described by two angles  $\varphi_q$  and  $\zeta_q$ , and the state vector can be expressed as:

$$|\psi_q\rangle = \cos(\varphi_q/2) |0\rangle + e^{i\zeta_q} \sin(\varphi_q/2) |1\rangle, \quad (31)$$

which can be visualized by means of the so-called Bloch sphere (see Fig. 1 and more details in Appendix C) [2]. Comparing Eq. (30) and Eq. (31) yields

$$\delta_q^{[0]} = \cos(\varphi_q/2) \quad \text{and} \quad \delta_q^{[1]} = e^{i\zeta_q} \sin(\varphi_q/2) = e^{i\zeta_q} \cos(\varphi_q/2 - \pi/2). \quad (32)$$

Therefore, if one considers  $n$  qubits separately, i.e. isolated from each other, they could be used to store  $N_{\text{sep}} = 2n$  real values, which are not many (currently  $N_{\text{sep}} \ll N$ ). In many applications, assuming real probability amplitudes, i.e.  $\zeta_q = 0$ , yields to a further contraction of the representable numbers, namely  $N_{\text{sep}}^{\text{real}} = n$ .

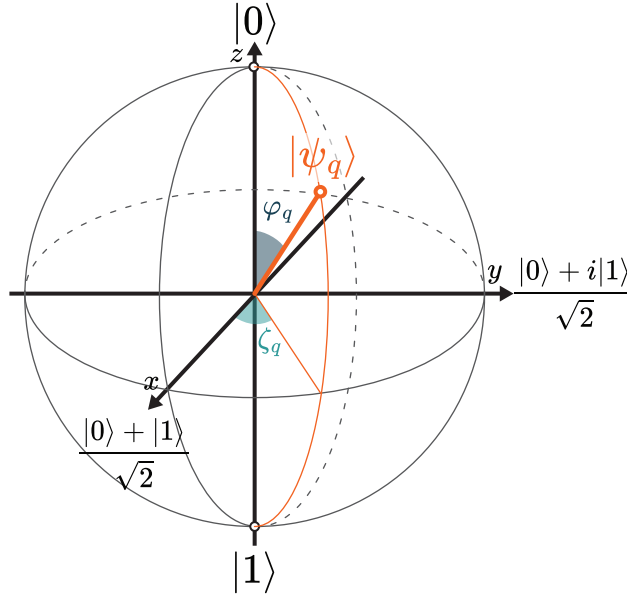


Figure 1: Bloch sphere representation of the qubit state  $|\psi_q\rangle$ . More details about the construction of the Bloch sphere representation of a single qubit are reported in Appendix C.

A quantum computer involves more than one qubit and therefore we need to familiarize also with the multi-qubit representation. For the sake of simplicity, let us consider first the very special case where  $n$  qubits are isolated from each other, i.e. the quantum computer is in a separable quantum state. The fundamental tool for combining formally multiple qubits isolated from each other into a large single state vector is the tensor product, indicated by the  $\otimes$  symbol. As an example, let us consider a composite system made of two separable qubits  $|\psi_0\rangle$  and  $|\psi_1\rangle$  (hence without entanglement). The state vector of the composite system is

expressed by  $|\psi_{\text{sep}}\rangle$  and can be computed as

$$\begin{aligned}
|\psi_{\text{sep}}\rangle &= |\psi_0\rangle \otimes |\psi_1\rangle = \left( \delta_0^{[0]} |0\rangle + \delta_0^{[1]} |1\rangle \right) \otimes \left( \delta_1^{[0]} |0\rangle + \delta_1^{[1]} |1\rangle \right) = \\
&= \delta_0^{[0]} \delta_1^{[0]} |0\rangle |0\rangle + \delta_0^{[0]} \delta_1^{[1]} |0\rangle |1\rangle + \delta_0^{[1]} \delta_1^{[0]} |1\rangle |0\rangle + \delta_0^{[1]} \delta_1^{[1]} |1\rangle |1\rangle = \\
&= \delta_0^{[0]} \delta_1^{[0]} |00\rangle + \delta_0^{[0]} \delta_1^{[1]} |01\rangle + \delta_0^{[1]} \delta_1^{[0]} |10\rangle + \delta_0^{[1]} \delta_1^{[1]} |11\rangle.
\end{aligned} \tag{33}$$

In the previous derivation, the abbreviated notation for tensor product has been adopted, namely  $|\beta_0\rangle \otimes |\beta_1\rangle = |\beta_0\rangle |\beta_1\rangle = |\beta_0\beta_1\rangle$  (it is possible to assume that  $|\beta_0\rangle |\beta_1\rangle = |\beta_0\beta_1\rangle$  is always valid because the computational basis is always separable [2]), where  $\beta_q \in \{0, 1\}$  and  $|\beta\rangle$  is one of the computational basis states of the  $q$ -th qubit. Let us link Eq. (33) with the computational mesh. Let us convert the already-mentioned index  $l$ , which identifies a mesh node, to a binary number represented by a string  $\beta_0\beta_1$  such that

$$l = \beta_0 2^1 + \beta_1 2^0. \tag{34}$$

Let  $j$  be this string, namely  $j = \beta_0\beta_1$ , which is useful to identify both the mesh nodes by  $j_2 = l_{10}$  but also the corresponding computational basis states  $|j\rangle$ , namely

$$|j\rangle = |\beta_0\beta_1\rangle, \tag{35}$$

where again it is worth noting that the big-endian convention is adopted [2]. The corresponding complex probability amplitude of the computational basis state  $|j\rangle$ , in case of a composite system made of two (separable) qubits, can be expressed as

$$\alpha_j^{\text{sep}} = \delta_0^{|\beta_0\rangle} \delta_1^{|\beta_1\rangle}, \tag{36}$$

where  $\alpha_j^{\text{sep}} \in \mathbb{C}$  in general. Therefore, the state vector of this composite separable system can be expressed as

$$|\psi_{\text{sep}}\rangle = |\psi_0\rangle |\psi_1\rangle = \sum_{j=00}^{11} \alpha_j^{\text{sep}} |j\rangle = \alpha_{00}^{\text{sep}} |00\rangle + \alpha_{01}^{\text{sep}} |01\rangle + \alpha_{10}^{\text{sep}} |10\rangle + \alpha_{11}^{\text{sep}} |11\rangle, \tag{37}$$

where the same abbreviated notation of the tensor product has been used also for the state vector of the composite system. A column vector representation is also useful for understanding how the tensor product works, namely

$$\begin{pmatrix} \delta_0^{[0]} \\ \delta_0^{[1]} \end{pmatrix} \otimes \begin{pmatrix} \delta_1^{[0]} \\ \delta_1^{[1]} \end{pmatrix} = \begin{pmatrix} \delta_0^{[0]} \delta_1^{[0]} \\ \delta_0^{[0]} \delta_1^{[1]} \\ \delta_0^{[1]} \delta_1^{[0]} \\ \delta_0^{[1]} \delta_1^{[1]} \end{pmatrix} = \begin{pmatrix} \alpha_{00}^{\text{sep}} \\ \alpha_{01}^{\text{sep}} \\ \alpha_{10}^{\text{sep}} \\ \alpha_{11}^{\text{sep}} \end{pmatrix}. \tag{38}$$

The tensor product between two vectors produces a larger vector, and it should not be confused with the dyadic product in fluid-dynamics, which would lead to a second order tensor

instead<sup>2</sup>. It is worth noting that the number of elements in the set  $\{\alpha_{00}^{\text{sep}}, \alpha_{01}^{\text{sep}}, \alpha_{10}^{\text{sep}}, \alpha_{11}^{\text{sep}}\}$  grows exponentially with  $n$ , but all these terms depend on amplitudes  $\delta_0^{[0]}, \delta_0^{[1]}, \delta_1^{[0]}$  and  $\delta_1^{[1]}$ : there are  $2n$  such terms, meaning their number grows linearly with  $n$ . Therefore, in separable states, the representable numbers grow only linearly with  $n$ .

The tensor product can be used to represent only separable states, i.e. states without entanglement. *Entanglement* is a fundamental property of quantum mechanics where the quantum state of a system composed of multiple subsystems cannot be described as a simple product of the states of its individual parts [2]. On the contrary, the system exists in a superposition of correlated states, such that the measurement of one subsystem instantaneously affects the state of the other, no matter how far apart they are [2]. An intuitive example of entanglement for non-experts is reported in Appendix A. In order to understand that entanglement allows one to represent (many more) new correlated states, which are unreachable by Eq. (38), let us consider the following example based on only two qubits. Let us limit ourselves to only real probability amplitudes, i.e.  $\zeta_q = 0$  for  $q \in \{0, 1\}$ . A well-known entangled state in the computational basis is the Bell state (see also the Appendix B for an intuitive example), which is a maximally entangled state:

$$\begin{pmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{pmatrix} = \begin{pmatrix} 1/\sqrt{2} \\ 0 \\ 0 \\ 1/\sqrt{2} \end{pmatrix}. \quad (39)$$

This state cannot be factorized into a product of two individual qubit states, i.e., recalling that we assumed  $\zeta_0 = \zeta_1 = 0$  for simplicity, there are no  $\varphi_0$  and  $\varphi_1$  in Eq. (32) which can be combined in Eq. (38) to represent the previous entangled (correlated) state. This is clear because  $\alpha_{00}$  and  $\alpha_{11}$  require non-zero  $\delta_0^{[\beta_0]}$  and  $\delta_1^{[\beta_1]}$ , but this is contradictory with the conditions derived by  $\alpha_{01}$  and  $\alpha_{10}$ . The Bell state can be included instead, by generalizing Eq. (37) for the unknown vector state

$$|\psi\rangle = \sum_{j=00}^{11} \alpha_j |j\rangle = \alpha_{00} |00\rangle + \alpha_{01} |01\rangle + \alpha_{10} |10\rangle + \alpha_{11} |11\rangle, \quad (40)$$

where now the generic  $\alpha_j \in \{\alpha_{00}, \alpha_{01}, \alpha_{10}, \alpha_{11}\}$  can be any complex number, only fulfilling the normalization condition  $\sum_j \alpha_j^2 = 1$ . The previous generalization is useful to realize that the number of elements in the set  $\{\alpha_{00}, \alpha_{01}, \alpha_{10}, \alpha_{11}\}$  grows exponentially with  $n$ , and these terms are now independent from each other: Therefore, in generic states, the representable numbers grow exponentially with  $n$ . This exponential increase of the number of computational basis states can be understood by introducing some correlation among qubits due to entanglement.

---

<sup>2</sup>The tensor product  $\otimes$  of two vectors  $|\psi_A\rangle$  and  $|\psi_B\rangle$  creates a new vector  $|\psi_A\rangle \otimes |\psi_B\rangle$  in a space with dimension  $d_A \times d_B$ , which is larger than the individual spaces having dimensions  $d_A$  and  $d_B$ , respectively. On the other hand, the dyadic product (sometimes indicated by the same symbol in fluid-dynamics), produces a matrix (operator), not a vector, and it is indicated here by  $|\psi_A\rangle \langle \psi_B|$ .

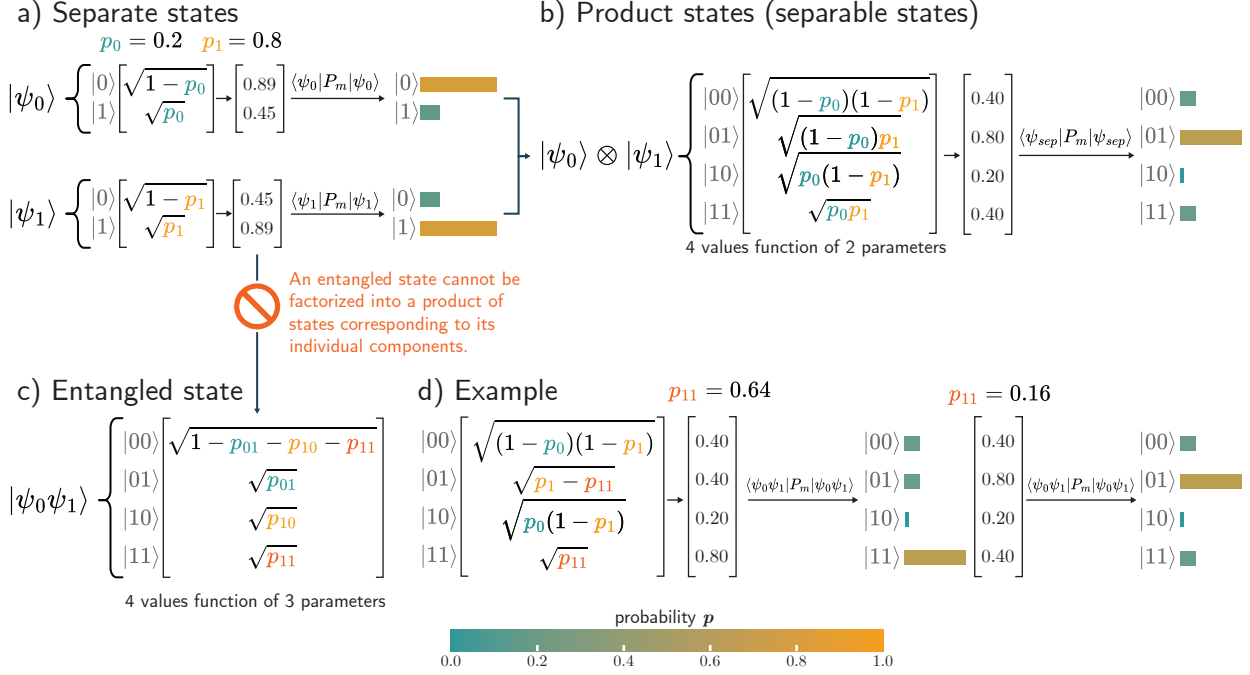


Figure 2: Schematic of a composite system of two qubits. (a) Separate states, (b) tensor product of the two individual states, (c) entangled state, and (d) an example where two marginal probabilities are fixed ( $p_{00}$ ,  $p_{10}$ ) and consistent with the previous case, while varying  $p_{11}$ . The measurement probabilities for each basis state are indicated using colored bars and computed using the projector  $P_m = |m\rangle\langle m|$ , where  $|m\rangle \in \{|0\rangle, |1\rangle\}$  for single-qubit states or  $|m\rangle \in \{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$  for the two-qubit case.

The fundamental difference between a separable two-qubit system (i.e. a system obtained by the tensor product of two individual states) and an entangled two-qubit system is depicted in Fig. 2. This example clearly shows that entanglement enables the exploration of a much broader space of states compared to what is achievable using only separable states. Hence, the entanglement is the key to exploit fully the tremendous potential of all computational basis states. It is the entanglement or – with other words – the presence of correlated states, which makes the microscopic scenario discussed here substantially different from other microscopic theories, e.g. the kinetic theory of gases. In the latter theory, the assumption of molecular chaos (also known as the *Stosszahlansatz*), is a key assumption in the derivation of the Boltzmann equation for dilute gases. The assumption of molecular chaos states that before a collision occurs, the velocities of two colliding particles are uncorrelated.

After becoming more familiar with the quantum nomenclature, let us come back to a more realistic quantum computer. A quantum computer with  $n$  qubit has typically a much larger computational basis than the previous example. In the following, the argument above will be generalized to any number  $n$  of qubits. An  $n$  qubit system has  $N = 2^n$  computational basis states, analogously to a classical system, but, unlike classical bits that exist in one state at a time, qubits can also exist in superpositions of all these states, according to some probabilities, which can also be correlated and hence can be freely explored, thanks to the entanglement. Therefore, a quantum computer can store (and process) an amount of real

numbers which is at least equal to the number of computational basis states (see next). In principle, the total number of real degrees of freedom in a general  $n$ -qubit quantum state is  $2^{n+1} - 2$ , accounting for normalization and the fact that the absolute phase of the state has no physical significance. If we impose the condition that the amplitudes are real, the normalization constraint still applies, but the global phase is restricted to  $\pm 1$ , which does not introduce a continuous degree of freedom. As a result, the number of available degrees of freedom reduces to  $2^n - 1$ . In the following, we will sometimes simplify this expression by stating that a quantum computer can store (and process) at least  $2^n$  real numbers, leveraging the imaginary parts of certain complex amplitudes to compensate for the missing degree of freedom. This proves that, in practice, we have significant control over operating with  $2^n$  real numbers by choosing appropriate gate sets or adding auxiliary qubits. Moreover, if we purposely restrict our gate set to only real-valued operations, the resulting quantum state will remain in the space of real amplitudes.

Again, for the sake of simplicity, let us first consider the discrete system state vector representing  $n$  separable qubits (without entanglement), namely

$$|\psi_{\text{sep}}\rangle = |\psi_0\rangle \otimes |\psi_1\rangle \otimes \dots \otimes |\psi_{n-1}\rangle = |\psi_0\rangle |\psi_1\rangle \dots |\psi_{n-1}\rangle, \quad (41)$$

where the last expression is an abbreviated notation for the tensor product among individual qubits (with big-endian convention [2]). More precisely, the system quantum state involves associating a complex coefficient  $\alpha_j^{\text{sep}} \in \mathbb{C}$  (called an amplitude) with each computational basis state  $|j\rangle$ . The amplitude for separable states is actually a composite amplitude which comes from multiplying the qubit individual amplitudes  $\delta_q^{|\beta_q\rangle}$  with each other, namely

$$\alpha_j^{\text{sep}} = \delta_0^{|\beta_0\rangle} \delta_1^{|\beta_1\rangle} \dots \delta_{n-1}^{|\beta_{n-1}\rangle} = \prod_{q=0}^{n-1} \delta_q^{|\beta_q\rangle}, \quad (42)$$

which generalizes Eq. (36). Using the relations given by Eqs. (32) yields

$$\alpha_j^{\text{sep}} = \prod_{q=0}^{n-1} \delta_q^{|\beta_q\rangle} = \prod_{q=0}^{n-1} e^{i\zeta_q \beta_q} \cos[\varphi_q/2 - \beta_q(\pi/2)], \quad (43)$$

where we took advantage of the property  $\cos(\varphi_q/2 - \pi/2) = \sin(\varphi_q/2)$  and  $\beta_q \in \{0, 1\}$  as usual, depending on the considered  $q$ -th qubit. Again, let us convert the already-mentioned index  $l$ , which identifies a mesh node, to a binary number represented by a string  $j = \beta_0\beta_1 \dots \beta_{n-1}$  by the following formula:

$$l = \sum_{q=0}^{n-1} \beta_q 2^{n-1-q}, \quad (44)$$

which ensures that  $j_2 = l_{10}$  in accordance with the big-endian convention [2]. The computational basis state corresponding to  $j$  is indicated by  $|j\rangle$ , which generalizes the binary representation given by Eq. (35), defined as

$$|j\rangle = |\beta_0\beta_1 \dots \beta_{n-1}\rangle, \quad (45)$$

where  $|j\rangle \in \{|0\rangle, |1\rangle\}^{\otimes n}$ , which means that  $|j\rangle$  can be  $|00\dots 0\rangle, |00\dots 1\rangle, \dots, |11\dots 1\rangle$  (the computational basis is always separable). Clearly there are  $N = 2^n$  elements in the set  $\{|0\rangle, |1\rangle\}^{\otimes n}$ . As done in the previous example, it is possible to exploit the full capability of the previous set by including also correlated states by entanglement, which are many more (by far!), in the formula for the state vector of the system in order to ensure a general validity now, namely

$$|\psi\rangle = |\psi_0 \psi_1 \dots \psi_{n-1}\rangle = \sum_{|j\rangle \in \{|0\rangle, |1\rangle\}^{\otimes n}} \alpha_j |j\rangle, \quad (46)$$

which generalizes Eq. (40). Please note that, in general,  $|\psi_0 \psi_1 \dots \psi_{n-1}\rangle \neq |\psi_0\rangle |\psi_1\rangle \dots |\psi_{n-1}\rangle$ , namely the state vector of the composite system is usually not separable. The probability  $p_j$  of finding the system state in the computational basis state  $|j\rangle$  is given by  $p_j = \alpha_j^2$ , where  $\sum_j p_j = \sum_j \alpha_j^2 = 1$ . In other words, one can say that the discrete vector quantum state  $|\psi\rangle$  is normalized, namely  $\langle\psi|\psi\rangle = 1$  where  $\langle\cdot|\cdot\rangle$  denotes the inner product.

For the sake of simplicity, let us suppose to construct a quantum circuit such that the output state is characterized by real amplitudes  $\alpha_j = a_j \in \mathbb{R}$ . In the output of such quantum circuit, the non-trivial (real) amplitudes  $a_j$  are  $2^n$  real numbers (where typically  $2^n \gg n$ ), which can be used to store a huge number of relevant information for the problem of interest. On the other hand, in a classical computer, real numbers are typically stored as floating-point approximations rather than exact values (e.g. according to the IEEE 754 floating-point standard). In particular,  $n$  classical bits can store a fixed-precision binary representation of a real number, which is equivalent to store one coded state among all available computational states (which are  $2^n$ ). If one wants to express the same concept by using a classical probability distribution, it would be like the classical probability distribution is equal to one only for the coded state and zero otherwise (Dirac delta distribution). Hence the difference between a quantum computer and a classical computer is that we can load  $2^n$  real numbers in the corresponding amplitudes of a quantum circuit thanks to superposition (non-trivial probability distribution of states), while we can code only one discrete state at the time in a classical computer among all possible states (Dirac delta distribution of states). The (potentially) tremendous advantage is clear ( $2^n \gg \lfloor n_{\text{classic}}/64 \rfloor$ ) and it can be represented in a non-rigorous way by the following expression (which is meaningful at the current stage of development of the quantum technology)

$$N_{\text{sep}} \ll N = 2^n. \quad (47)$$

Next, we need to understand how a quantum circuit can be used to perform the desired calculations. There are five main steps:

1. Identification of the quantum state (Normalization);
2. Design of the quantum system (Observable);
3. Selection of the quantum parametrized trial solution (Quantum ansatz);
4. Minimization of the loss function (Optimization);
5. Extraction of useful results (De-normalization).



### 3.2. Normalization

The first step is to identify the quantum state where to store the relevant information by normalizing Eq. (7). Because a discrete quantum state  $|\psi\rangle$  is normalized, namely  $\langle\psi|\psi\rangle = 1$ , let us divide Eq. (7) by a factor such that it becomes possible to identify a quantum state which depends on the temperature profile. In particular, let us choose this factor as follows:

$$\frac{1}{(\vec{T}^+ \cdot \vec{T}^+)(\vec{T} \cdot \vec{T})} \hat{C} \vec{T}^+ = \frac{1}{(\vec{T}^+ \cdot \vec{T}^+)(\vec{T} \cdot \vec{T})} \vec{T}, \quad (48)$$

where  $\vec{T} \cdot \vec{T}$  denotes the inner product (i.e.,  $\vec{T}^\dagger \cdot \vec{T}$ , and since  $\vec{T}$  is real-valued,  $\vec{T}^\dagger$  reduces to the transpose of  $\vec{T}$ ); similarly for  $\vec{T}^+ \cdot \vec{T}^+$ . Let us define the quantum state  $|b\rangle$  for mapping the initial temperature profile, namely

$$|b\rangle = \frac{1}{\sqrt{\vec{T} \cdot \vec{T}}} \vec{T}, \quad (49)$$

where, by construction,  $\langle b|b\rangle = 1$ . Before proceeding, let us be sure to appreciate the true meaning of the previous relation. Essentially it implies that each node  $z_l$  of the computational mesh is associated with a quantum computational basis state  $|j\rangle$  (where  $|j\rangle \in \{0, 1\}^{\otimes n}$  involves the binary representation of integer  $l$ ). Similarly let us proceed with the quantum state  $|x\rangle$  for mapping the updated temperature profile at the new time step, namely

$$|x\rangle = \frac{1}{\sqrt{\vec{T}^+ \cdot \vec{T}^+}} \vec{T}^+, \quad (50)$$

where the same normalization holds. Introducing the definitions given by Eq. (49) and Eq. (50) into Eq. (48) yields

$$\sqrt{\frac{\vec{T}^+ \cdot \vec{T}^+}{\vec{T} \cdot \vec{T}}} \hat{C} |x\rangle = |b\rangle. \quad (51)$$

It is also possible to define a normalization factor  $f$  given by

$$f = \sqrt{\frac{\vec{T}^+ \cdot \vec{T}^+}{\vec{T} \cdot \vec{T}}}, \quad (52)$$

and to derive the linear system of target equations as

$$\hat{A} |x\rangle = |b\rangle, \quad (53)$$

where  $\hat{A} = f \hat{C}$ . The problem is that the normalization factor is not known at the beginning of the numerical procedure because it depends on the solution  $\vec{T}^+$ , which derives from solving the linear system of equations. This means that  $\hat{A}$  can be used to discuss the theoretical setup, but the practical numerical procedure must involve  $\hat{C}$ , because the latter depends only on the adopted FD formula.

### 3.3. Observable

The second step is to derive a quantum system, which provides information relevant to solve the target problem. Let us follow the strategy proposed in Ref. [5] to construct a Hamiltonian, which admits the quantum state  $|x\rangle$  as the ground state. Although heat conduction is intrinsically a dissipative process and thus not Hamiltonian in the strict sense, we introduce a ‘Hamiltonian’ here to refer to the matrix arising from the finite-difference discretization, which structurally resembles a Hamiltonian operator. This formalism facilitates the following analysis (see section 1.1 for more details). Applying the methodology described in Ref. [5] to Eq. (53) yields the following Hamiltonian

$$\hat{H}' = \hat{A}^\dagger (I - |b\rangle \langle b|) \hat{A}, \quad (54)$$

where  $|\cdot\rangle \langle \cdot|$  denotes the outer product and  $\hat{A}^\dagger$  is the conjugate transpose of  $\hat{A}$  in general, while, in this case,  $\hat{A}^\dagger = \hat{A}^T$ , because  $\hat{A}$  is real. In this case,  $\hat{A}$  is also symmetric and therefore  $\hat{A}^T = \hat{A}$ . As pointed out before, let us formulate the quantum algorithm in terms of the practical operator  $\hat{C}$  which does not involve the normalization factor. Since  $\hat{A}^T = \hat{A} = f \hat{C}$ , the previous Hamiltonian can be computed as  $\hat{H}' = f^2 \hat{H}$  where

$$\hat{H} \equiv \hat{O} = \hat{C}^T (I - |b\rangle \langle b|) \hat{C}. \quad (55)$$

In the previous formula, we implicitly remind that the Hamiltonian is just a special case of quantum observable (where energy is the actual observed quantity): therefore, it makes sense to use instead the symbol  $\hat{O}$  from now on for making the procedure as universal as possible. Let us decompose this operator as  $\hat{O} = \hat{C}^T \hat{M} \hat{C}$ , where  $\hat{M} = I - |b\rangle \langle b|$  is a projector operator. It is possible to prove that  $\hat{M}$  is a projector operator because  $\hat{M}^2 = I - 2|b\rangle \langle b| + |b\rangle \langle b| = \hat{M}$ . It projects onto the orthogonal complement of  $|b\rangle$ . That is, it removes the component of a vector along  $|b\rangle$ , leaving only the part orthogonal to  $|b\rangle$ . First of all, let us analyze the eigenvalues of  $\hat{M}$ , by calling  $\lambda_k$  the  $k$ -th eigenvalue and  $|\phi_k\rangle$  the corresponding eigenstate, namely  $\hat{M} |\phi_k\rangle = \lambda_k |\phi_k\rangle$ . The  $k$ -th eigenvalue can be computed as

$$\lambda_k = \langle \phi_k | \hat{M} | \phi_k \rangle = 1 - \langle \phi_k | b \rangle^2. \quad (56)$$

Recalling the Cauchy–Schwarz inequality, namely

$$\langle \phi_k | b \rangle \leq \sqrt{\langle \phi_k | \phi_k \rangle} \sqrt{\langle b | b \rangle} = 1, \quad (57)$$

it is possible to find out that

$$\lambda_k = 1 - \langle \phi_k | b \rangle^2 \geq 0. \quad (58)$$

This means that all eigenvalues of  $\hat{M}$  are larger than or equal to zero, i.e. the core operator  $\hat{M}$  is positive semi-definite. Using the same orthonormal basis, it is possible to express  $\hat{M}$  by spectral decomposition

$$\hat{M} = \sum_{k=0}^{N-1} \lambda_k |\phi_k\rangle \langle \phi_k|. \quad (59)$$

Now, coming back to the main observable  $\hat{O}$ , let us consider the expectation value of the observable  $\hat{O}$  with regards to the generic state vector  $|\psi\rangle$ , namely

$$\langle\psi|\hat{O}|\psi\rangle = \langle\psi|\hat{C}^T\left(\sum_{k=0}^{N-1}\lambda_k|\phi_k\rangle\langle\phi_k|\right)\hat{C}|\psi\rangle = \sum_{k=0}^{N-1}\lambda_k\langle\phi_k|\hat{C}|\psi\rangle^2, \quad (60)$$

where we used  $\hat{C}^T = \hat{C}$ . Using the result given by Eq. (58) yields

$$\langle\psi|\hat{O}|\psi\rangle \geq 0, \quad (61)$$

which proves that the main observable  $\hat{O}$  is also positive semi-definite, i.e. all eigenvalues of  $\hat{O}$  are positive or equal to zero. Clearly  $|x\rangle$  is the eigenvector of  $\hat{O}$  corresponding to the zero eigenvalue, namely

$$\hat{O}|x\rangle = \hat{O}\hat{A}^{-1}|b\rangle = f^{-1}\hat{C}^T(I - |b\rangle\langle b|)|b\rangle = 0, \quad (62)$$

which proves that  $|x\rangle$  is the ground state of the operator  $\hat{O}$ , as expected by design. The linear algebra task given by Eq. (53) is converted to the task of finding the ground state of the Hamiltonian  $\hat{H} \equiv \hat{O}$  [5].

### 3.4. Quantum circuit ansatz

The third step is the quantum circuit ansatz, i.e. a parameterized trial solution which should be able to approximate the ground state  $|x\rangle$ . The key point is that, in order to prove the quantum supremacy, there are too many elements in the vector  $|x\rangle$  to work on them directly by a classical computer. Let us recall that VQE is a hybrid algorithm, where the optimization is supposed to be done by a classical computer [2]. Hence let us introduce a vector of parameters  $\vec{\theta}$ , which are fewer such that they can be handled by a classical computer. The quantum circuit ansatz enforces a parameterized state  $|x(\vec{\theta})\rangle$  which makes possible to map these parameters on a generic quantum state. The parameterized state is the output of a unitary transformation (quantum gate), namely

$$|x(\vec{\theta})\rangle = U(\vec{\theta})|0\rangle^{\otimes n}, \quad (63)$$

where  $|0\rangle^{\otimes n}$  stands for  $|0\rangle \otimes |0\rangle \otimes \dots \otimes |0\rangle = |0\rangle|0\rangle \dots |0\rangle = |00\dots 0\rangle$  (the computational basis is always separable) and  $\vec{\theta}$  is a vector of tunable parameters. The parameters  $\vec{\theta}$  are typically generic ‘rotations’ of qubits which are optimized during the minimization step of the VQE algorithm (see next section 3.5). In order to preserve the quantum advantage, the number of parameters to optimize over in the ansatz circuit must be much less than the size of the computational basis of the quantum states, because the former is handled by a classical optimizer/minimizer, while the latter exploits the full capability of the quantum computer. With other words, the number of parameters must grow as a polynomial in the number  $n$  of qubits, while the size of the full vector  $|x\rangle$  is exponential in the number of qubits [4]. It is not important which polynomial describes the growth of the number of parameters, because any polynomial cannot compete with the growth of the exponential function  $2^n$  for large  $n$ . In the following sections, for example, we will see  $8n$  parameters in the ansatz depicted in Fig. (3) and  $4n$  parameters in the simplified ansatz depicted in Fig. (6).

### 3.5. Optimization

The fourth step is the actual minimization of the loss function. A loss function quantifies the difference (“loss”) between a quantum state predicted by the ansatz for a given input and the ground state. Taking into account Eq. (61) and the ansatz given by Eq. (63), the loss function can be defined as

$$L(\vec{\theta}) = \langle x(\vec{\theta}) | \hat{O} | x(\vec{\theta}) \rangle \geq 0. \quad (64)$$

The optimal set of parameters can be formally defined by the argument of the minimization problem with regards to the loss function, namely

$$\vec{\theta}_{\min} = \arg \min_{\vec{\theta}} L(\vec{\theta}) = \arg \min_{\vec{\theta}} \langle x(\vec{\theta}) | \hat{O} | x(\vec{\theta}) \rangle. \quad (65)$$

Let us define  $|x_{\min}\rangle$  as

$$|x_{\min}\rangle = |x(\vec{\theta}_{\min})\rangle. \quad (66)$$

Because of numerical errors,  $|x_{\min}\rangle$  is different from the theoretical ground state, i.e.  $|x_{\min}\rangle \neq |x\rangle$ , but it is usually close enough.

### 3.6. De-normalization

The final step is to de-normalize the numerical quantum approximation  $|x_{\min}\rangle$  for coming back to the original quantity of interest, i.e. the temperature. Let us start with the initial temperature profile. Let us define the auxiliary quantity

$$\theta = \sqrt{\vec{T} \cdot \vec{T}}, \quad (67)$$

which can be used to express Eq. (49) as  $\vec{T} = \theta |b\rangle$ . Let us compute this auxiliary quantity  $\theta$  by the spatial average of the initial temperature profile, namely

$$\theta = \frac{\sum_l T_l}{\sum_j \langle j|b\rangle}. \quad (68)$$

where  $\sum_j \langle j|b\rangle$  is the sum of all real amplitudes in  $|b\rangle$ . Please remember that  $|j\rangle \in \{0, 1\}^{\otimes n}$  involves the binary representation of integer  $l$ . Similarly, recalling Eq. (50), let us define

$$\theta^+ = \sqrt{\vec{T}^+ \cdot \vec{T}^+}. \quad (69)$$

which can now be computed by using the quantum numerical approximation  $|x_{\min}\rangle$ . Taking advantage of the energy conservation, which implies

$$\sum_l T_l^+ = \sum_l T_l, \quad (70)$$

the quantity  $\theta^+$  can be computed by the following formula

$$\theta^+ = \frac{\sum_l T_l}{\sum_j \langle j|x_{\min}\rangle}. \quad (71)$$

It is worth to note that, in case of accurate minimization, all terms in the summation at the denominator in Eq. (71) are positive because they correspond to the nodal values of the normalized new temperatures. Clearly  $f = \theta^+/\theta$ . The quantity  $\theta^+$  is essential to de-normalize the quantum solution and to come back to the updated temperature profile, namely

$$\vec{T}^+ = \theta^+ |x_{\min}\rangle. \quad (72)$$

So far we presented the straightforward implementation of the VQE approach which demonstrates a fundamental possibility to solve linear algebra problems, and in particular the discretized conduction equation on a quantum processor. However, it has one essential disadvantage: this algorithm requires to decompose the observable  $\hat{O}$  in terms of a sequence of Pauli matrices (see next section for details). Usually the number of Pauli matrices in this decomposition is exponential in the number of qubits and hence it spoils the potential quantum speedup [6]. More sophisticated variational methods have been already proposed in the literature, which are more promising from a practical point of view [6]. One possibility consists in evaluating the loss function by an adaption of a fundamental quantum circuit, the so-called Hadamard test [7]. An even more effective implementation consists in combining the Hadamard test with the quantum Fourier transform [6]. A more advanced approach for near-term algorithms, namely algorithms suitable for near-term quantum hardware, is represented by the so-called ansatz tree [8], which has been already applied to the discretized conduction equation [6]. These techniques will be explored and compared in a future work.

### 3.7. Practical details of implementation

In this section, we need to complete the algorithm presented in previous section by adding more details about the actual implementation of the algorithm in a quantum computer. Even though these details are general, we will focus on Qiskit [4] by IBM as an example open-source software development kit. Qiskit [4] is an open-source framework for quantum computing that allows users to design, simulate, and run quantum programs on real hardware. It provides an intuitive way to build quantum circuits, optimize them for execution, and simulate their behavior before running on actual quantum processors. Qiskit also includes tools for error mitigation and circuit optimization, making it more practical for real-world use.

#### 3.7.1. Decomposition in Pauli matrices

In the presented algorithm, the loss function to be minimized  $L(\vec{\theta})$  is defined by the expectation value of the observable  $\hat{O}$  defined by Eq. (55). To measure the observable  $\hat{O}$  given by Eq. (55) on a quantum computer by Qiskit [4], one must represent it as a sum of tensor products of Pauli matrices, that is

$$\hat{H} \equiv \hat{O} = \sum_{p=0}^{N_p-1} \gamma_p \hat{P}_p, \quad (73)$$

where  $N_p$  is the number of terms in the Pauli decomposition of the Hamiltonian,  $\gamma_p \in \mathbb{R}$  because  $\hat{O} = \hat{O}^\dagger$  is Hermitian (actually it is real and symmetric in this case),  $\hat{P}_p \in$

$\{I, X, Y, Z\}^{\otimes n}$ , and the Pauli matrices are

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (74)$$

We will clarify soon the physical meaning of this decomposition in terms of Pauli matrices, but it is important to first understand the tensor product between matrices. As an example, let us consider again a composite system made of two qubits. In this case, the generic  $p$ -th element of the decomposition looks like

$$\begin{aligned} \hat{P}_p &= \sigma^{p0} \otimes \sigma^{p1} = \begin{bmatrix} \sigma_{11}^{p0} \sigma^{p1} & \sigma_{12}^{p0} \sigma^{p1} \\ \sigma_{21}^{p0} \sigma^{p1} & \sigma_{22}^{p0} \sigma^{p1} \end{bmatrix} = \\ &= \begin{bmatrix} \sigma_{11}^{p0} \begin{pmatrix} \sigma_{11}^{p1} & \sigma_{12}^{p1} \\ \sigma_{21}^{p1} & \sigma_{22}^{p1} \end{pmatrix} & \sigma_{12}^{p0} \begin{pmatrix} \sigma_{11}^{p1} & \sigma_{12}^{p1} \\ \sigma_{21}^{p1} & \sigma_{22}^{p1} \end{pmatrix} \\ \sigma_{21}^{p0} \begin{pmatrix} \sigma_{11}^{p1} & \sigma_{12}^{p1} \\ \sigma_{21}^{p1} & \sigma_{22}^{p1} \end{pmatrix} & \sigma_{22}^{p0} \begin{pmatrix} \sigma_{11}^{p1} & \sigma_{12}^{p1} \\ \sigma_{21}^{p1} & \sigma_{22}^{p1} \end{pmatrix} \end{bmatrix} = \\ &= \begin{bmatrix} \sigma_{11}^{p0} \sigma_{11}^{p1} & \sigma_{11}^{p0} \sigma_{12}^{p1} & \sigma_{12}^{p0} \sigma_{11}^{p1} & \sigma_{12}^{p0} \sigma_{12}^{p1} \\ \sigma_{11}^{p0} \sigma_{21}^{p1} & \sigma_{11}^{p0} \sigma_{22}^{p1} & \sigma_{12}^{p0} \sigma_{21}^{p1} & \sigma_{12}^{p0} \sigma_{22}^{p1} \\ \sigma_{21}^{p0} \sigma_{11}^{p1} & \sigma_{21}^{p0} \sigma_{12}^{p1} & \sigma_{22}^{p0} \sigma_{11}^{p1} & \sigma_{22}^{p0} \sigma_{12}^{p1} \\ \sigma_{21}^{p0} \sigma_{21}^{p1} & \sigma_{21}^{p0} \sigma_{22}^{p1} & \sigma_{22}^{p0} \sigma_{21}^{p1} & \sigma_{22}^{p0} \sigma_{22}^{p1} \end{bmatrix}, \end{aligned} \quad (75)$$

where  $\sigma^{p0}$  and  $\sigma^{p1}$  are matrices which can be  $I, X, Y$  or  $Z$  and they refer to the first and the second qubit, respectively. After becoming more familiar with this nomenclature, let us come back to the case with  $n$  qubits, which can become pretty complicated, namely

$$\hat{P}_p = \sigma^{p0} \otimes \sigma^{p1} \otimes \dots \sigma^{p(n-1)}, \quad (76)$$

where the detailed expressions are omitted for the sake of simplicity. Fortunately, the tensor product among matrices has a fundamental property [2], which only applies to separable states but can help in understanding this decomposition. Let us suppose to apply the  $p$ -th element of the decomposition  $\hat{P}_p$  to a separable vector state  $|\psi_{\text{sep}}\rangle = |\psi_0\rangle \otimes |\psi_1\rangle \otimes \dots \otimes |\psi_{n-1}\rangle$ , which yields

$$\begin{aligned} \hat{P}_p |\psi_{\text{sep}}\rangle &= [\sigma^{p0} \otimes \sigma^{p1} \otimes \dots \otimes \sigma^{p(n-1)}] (|\psi_0\rangle \otimes |\psi_1\rangle \otimes \dots \otimes |\psi_{n-1}\rangle) = \\ &= \sigma^{p0} |\psi_0\rangle \otimes \sigma^{p1} |\psi_1\rangle \otimes \dots \otimes \sigma^{p(n-1)} |\psi_{n-1}\rangle. \end{aligned} \quad (77)$$

The previous formula means that the result of  $\hat{P}_p |\psi_{\text{sep}}\rangle$  is simply the tensor product of the individual calculation  $\sigma^{pq} |\psi_q\rangle$  for all the qubits. This implies

$$\langle \psi_{\text{sep}} | \hat{P}_p | \psi_{\text{sep}} \rangle = \prod_{q=0}^{n-1} \langle \psi_q | \sigma^{pq} | \psi_q \rangle. \quad (78)$$

and consequently

$$\langle \psi_{\text{sep}} | \hat{O} | \psi_{\text{sep}} \rangle = \sum_{p=0}^{N_p-1} \gamma_p \prod_{q=0}^{n-1} \langle \psi_q | \sigma^{pq} | \psi_q \rangle, \quad (79)$$

which means that the expectation value of the observable  $\hat{O}$  with regards to separable states can be computed by a sequence of measurements on one qubit at a time (but it is crucial to change the measurement basis for the  $q$ -th qubit corresponding the  $\sigma^{pq}$  matrix). In case of non separable states, i.e. in case of entanglement, the previous simplification does not hold.

Even though the previous formula is a special case, it allows one to appreciate that there is a computational problem at this point, namely  $N_p$  grows pretty fast with  $n$  (exponentially, see next). Let us do an example. In case of a system with 3 qubits, the decomposition in Pauli matrices given by Eq. (73) is the following

$$\begin{aligned}\hat{O} = & \gamma_0 III + \gamma_1 IIX + \gamma_2 IXI + \gamma_3 IXX + \gamma_4 I X Z + \gamma_5 IYY + \\ & \gamma_6 IZI + \gamma_7 IZX + \gamma_8 IZZ + \gamma_9 XII + \gamma_{10} XIX + \gamma_{11} XXI + \\ & \gamma_{12} XXX + \gamma_{13} XXZ + \gamma_{14} XYY + \gamma_{15} XZI + \gamma_{16} XZX + \gamma_{17} XZZ + \\ & \gamma_{18} YIY + \gamma_{19} YXY + \gamma_{20} YYI + \gamma_{21} YYX + \gamma_{22} YYZ + \gamma_{23} YZY + \\ & \gamma_{24} ZII + \gamma_{25} ZIX + \gamma_{26} ZIZ + \gamma_{27} ZXI + \gamma_{28} ZXX + \gamma_{29} ZXZ + \\ & \gamma_{30} ZYY + \gamma_{31} ZZI + \gamma_{32} ZZX + \gamma_{33} ZZZ,\end{aligned}\tag{80}$$

where, for example,  $III$  means  $I \otimes I \otimes I$  and similarly for all remaining terms. In this example, for  $n = 3$  the Pauli decomposition requires  $N_p = 34$ . For the sake of comparison, for  $n = 4$  the Pauli decomposition requires  $N_p = 120 \sim \exp(k_p n)$ , where  $k_p$  is a proper constant. Therefore, the present application appears to be a case where the number of Pauli products in the Hamiltonian decomposition grows exponentially with the number of qubits, as suggested in [6]. However, the reference does not explicitly analyze this scaling, and as noted, leveraging the distribution of Pauli string weights can potentially reduce the complexity. While a brute-force or naive approach would indeed be impractical, the literature suggests alternative methods – such as truncation, grouping of Pauli strings, and other techniques – that might be applicable in this context. It remains unclear whether these approaches could be effectively employed in this specific case. More details about the Pauli decomposition can be found in Ref. [9].

### 3.7.2. Efficient circuit ansatz

In the presented algorithm, the loss function to be minimized  $L(\vec{\theta})$  is defined with regards to a parameterized trial solution  $|x(\vec{\theta})\rangle$ , which is called the ansatz. The ansatz given by Eq. (63) is a parameterized trial solution  $|x(\vec{\theta})\rangle$  which should be able to approximate the ground state  $|x\rangle$ . The parametrized solution is the output of a unitary transformation (quantum gate)  $U(\vec{\theta})$ , which depends on a vector  $\vec{\theta}$  of  $N_\theta$  tunable parameters. Naively one would like to have a procedure for correlating the parameters in  $\vec{\theta}$  with the real amplitudes in  $|x(\vec{\theta})\rangle$  by means of some analytical formulas. This approach is usually called real data loading or better encoding, and some algorithms have been proposed in literature [10]. For optimization problems – and for VQE in particular – real data loading/encoding is not strictly necessary and it will be omitted here in favor of a more efficient approach, namely an approach with less tunable parameters (see also Appendix D).

In Qiskit [4], let us consider a unitary transformation  $U(\vec{\theta})$  which consists of two ingredients: (i) four layers of single-qubit operations, (ii) spanned by controlled *NOT* gates

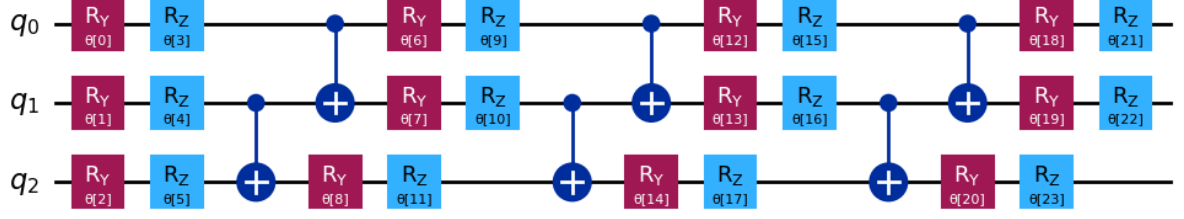


Figure 3: Efficient ansatz (3 qubit, 24 parameters = four layers with six parameters each or, equivalently, eight parameters per qubit). For clarity, horizontal lines represent quantum wires which correspond to qubits in the circuit, red squares are the  $R_Y$  gates (see Eq. (81)), blue squares are the  $R_Z$  gates (see Eq. (82)), blue dots represent control points in controlled gates,  $\oplus$  symbol is used for a controlled- $X$  ( $CNOT$ ) gate. The latter gate explicitly guarantees the desired entanglement.

(also called controlled- $X$  gates) for ensuring some degree of entanglement [2]. This is a heuristic pattern that can be used to prepare trial states for variational quantum algorithms or classification circuit for machine learning [4]. The single-qubit operations consist of the sequential application  $R_Y R_Z$  (in this case, there is no tensor product implied because both apply to the same qubit) of a  $R_Y$  gate and a  $R_Z$  gate, defined as

$$R_Y(\theta_Y) = \exp\left(-i \frac{\theta_Y}{2} Y\right) = \begin{pmatrix} \cos(\theta_Y/2) & -\sin(\theta_Y/2) \\ \sin(\theta_Y/2) & \cos(\theta_Y/2) \end{pmatrix}, \quad (81)$$

and

$$R_Z(\theta_Z) = \exp\left(-i \frac{\theta_Z}{2} Z\right) = \begin{pmatrix} \exp(-i\theta_Z/2) & 0 \\ 0 & \exp(i\theta_Z/2) \end{pmatrix}. \quad (82)$$

The previous definitions can be thought as derived from the same generic formula

$$\exp(-i\theta_\sigma \sigma) = \cos \theta_\sigma I - i \sin \theta_\sigma \sigma, \quad (83)$$

where  $\theta_\sigma$  is a parameter and  $\sigma$  is a matrix which can be the Pauli matrix  $Y$  or  $Z$ . The previous generic formula derives from the property of Pauli matrices (after multiplication by  $i = \sqrt{-1}$  to make them anti-Hermitian) to generate transformations in the sense of Lie algebras [2]. This formula is analogous for Pauli matrices to the Euler's formula of complex analysis.

This ansatz is called “*EfficientSU2*” circuit in Qiskit [4] and it is plotted in Fig. (3) for a system with 3 qubits. For a system with  $n = 3$  qubits,  $N_\theta = 24$  because there are four layers with six parameters each (two gates  $R_Y$  and  $R_Z$  for each qubit) or, equivalently, eight parameters per qubit. For the sake of comparison, for  $n = 4$  the number of parameters in this ansatz becomes  $N_\theta = 8n = 32$ . It is essential that the number of ansatz parameters to optimize over is linear as in this case (or polynomial at worst) in the number of qubits, in order to ensure a potential quantum supremacy (because  $N_\theta \sim k_\theta n \ll 2^n = N$ , where  $k_\theta$  is a proper constant).



### 3.7.3. Minimization of the loss function

VQE is a hybrid algorithm that combines (i) classical operations for the converging iterations and (ii) loss function evaluations by quantum operations, to find the ground state of the target quantum system, which is designed in our case to update the one-dimensional temperature profile consistently with the heat conduction equation.

For the converging iterations by classical operations, one can use the “*minimize*” function of the “*scipy.optimize*” library in the SciPy platform [3]. It is recommended to focus on Jacobian-free methods: for example, the “*COBYLA*” solver and “*L-BFGS-B*” solver, which produce similar performance according to our preliminary experiments. The goal is to minimize the number of evaluations of loss functions, which requires limiting the tolerance for termination in the range  $1\text{--}10 \times 10^{-3}$ .

## 3.8. Simulated results

### 3.8.1. Simulated results by Qiskit

For quantum computers in the NISQ era, the discussed algorithm for real applications is still very challenging, mainly because of qubit decoherence. For this reason, in order to perform some preliminary experiments, let us use the “*BaseEstimatorV2*” simulator available in Qiskit [4], which estimates expectation values for provided quantum circuit and observable combinations. An example implementation of the VQE in Qiskit is reported in Appendix F.

Let us compute the outcome of applying the time-progress operator  $\hat{C}^{-1}$  given by Eq. (8) to an initial temperature profile. With other words, let us perform one time step to update the temperature profile of our target problem. In Fig. (4), the results for one time-step update of the temperature profile according to the heat conduction equation are reported in case of  $n = 3$  qubits ( $N = 8$ ), *BaseEstimatorV2* quantum simulator and *COBYLA* classical minimizer with tolerance for termination  $1 \times 10^{-3}$ . This minimization required 839 evaluations of the loss function, which are still too many for most existing quantum computers to compete with classical computers. Similarly, in Fig. (5), the results for the same problem in case of  $n = 4$  qubits ( $N = 16$ ) are reported (the quantum simulator and the classical minimizer are the same as before). In this second case, even though the results look acceptable, we performed  $10^6$  evaluations of the loss function, hitting the upper maximum limit which we set in advance.

### 3.8.2. Simulated results by Qrisp

While the physics side of quantum computing makes significant progress, the support for high-level quantum programming abstractions is still in its infancy compared to modern classical languages and frameworks [11]. An interesting example is provided by Qrisp, which is a high-level programming language developed by Fraunhofer for creating and compiling quantum algorithms [11]. Its structured programming model enables scalable development and maintenance [11]. An example implementation of the VQE in Qrisp is reported in Appendix F.

For simplicity, the Qrisp example code employs the simplified ansatz shown in Fig. 6, which consists solely of  $R_Y$  and  $CNOT$  gates. Among the relevant parameters for the

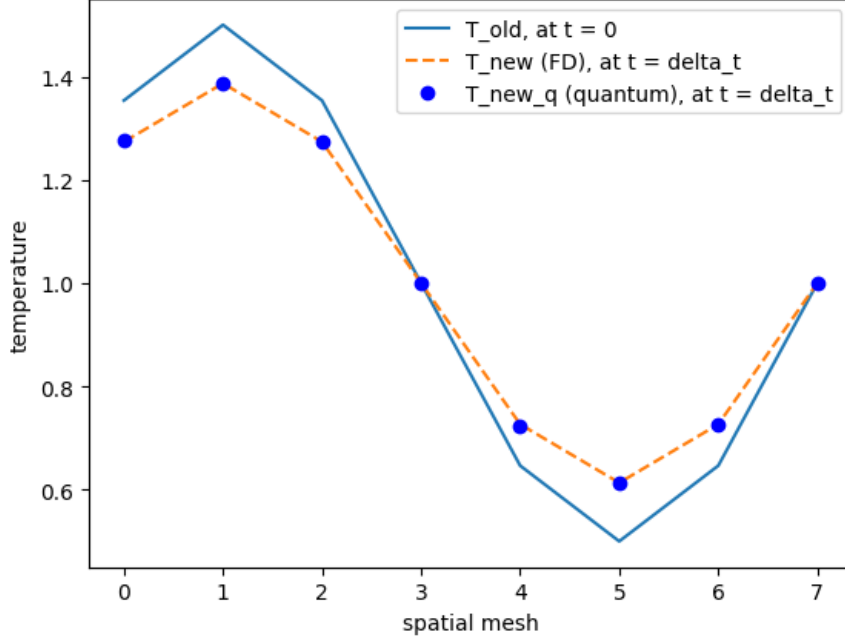


Figure 4: One time-step update of the temperature profile according to heat conduction equation by quantum computing (3 qubits, *BaseEstimatorV2* quantum simulator, *COBYLA* classical minimizer with tolerance for termination  $1 \times 10^{-3}$ ). The blue line is the initial temperature profile (with mean equal to 1), the orange dashed line is the new temperature profile at time  $\Delta t$ , computed by finite-difference method. The blue dots are the mesh node temperatures computed by the quantum simulator.

*vqe.run* method, used to compute the system’s energy in the Qrisp example code (see Fig. Appendix F), the most important is ‘*precision*’, as it determines the number of shots during execution on real hardware. In quantum computing, ‘shots’ refer to the number of times a quantum circuit is executed to collect measurement statistics. Since quantum measurements are probabilistic, multiple shots are required to estimate expectation values with sufficient accuracy. Hence precision refers to how accurately the Hamiltonian is evaluated. The number of shots the real quantum hardware performs per iteration scales quadratically with the inverse precision. Therefore it is important to estimate properly the required precision in order to assess the feasibility of running a VQE algorithm on real quantum hardware.

In Fig. 7 the impact of parameter ‘*precision*’ on simulated results is investigated by 30 repetitions of the VQE algorithm for collecting some relevant statistics (mean and standard deviation) of the performed measurements. In particular, ‘*precision*’: 0.025 and ‘*precision*’: 0.001 are considered. In the first case, the uncertainty is larger than the temperature differences due to the heat conduction time step, making the simulation practically useless. This proves that actual precision, or equivalently the maximum number of shots, are limiting factors for successfully implementing VQE algorithms on real quantum hardware.

### 3.9. VQE based on diagonalizing the measurement

As already mentioned, the key for modeling an irreversible phenomenon by an ideal quantum computer is to properly design the measurement. Unfortunately, the naive VQE

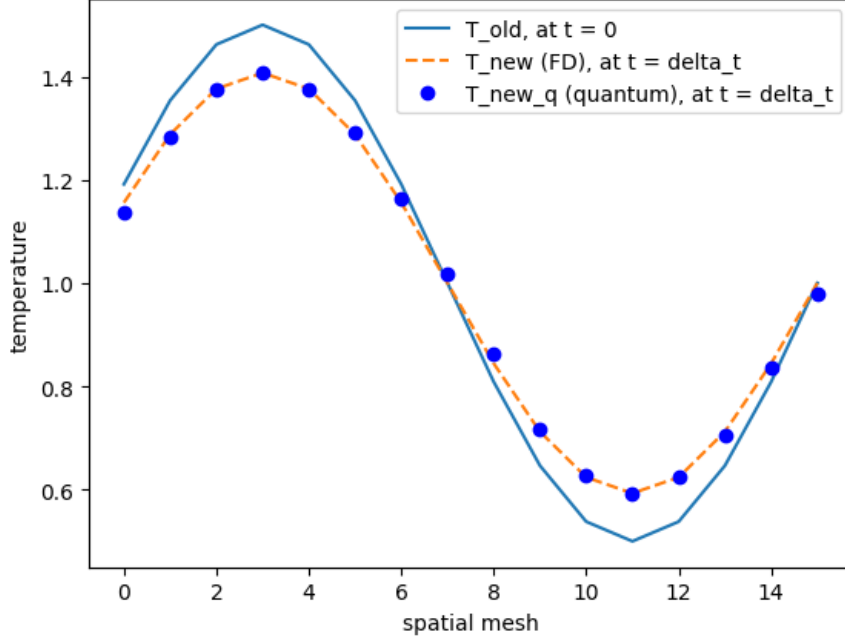


Figure 5: One time-step update of the temperature profile according to heat conduction equation by quantum computing (4 qubits, *BaseEstimatorV2* quantum simulator, *COBYLA* classical minimizer with tolerance for termination  $1 \times 10^{-3}$ ). The blue line is the initial temperature profile (with mean equal to 1), the orange dashed line is the new temperature profile at time  $\Delta t$ , computed by finite-difference method. The blue dots are the mesh node temperatures computed by the quantum simulator.

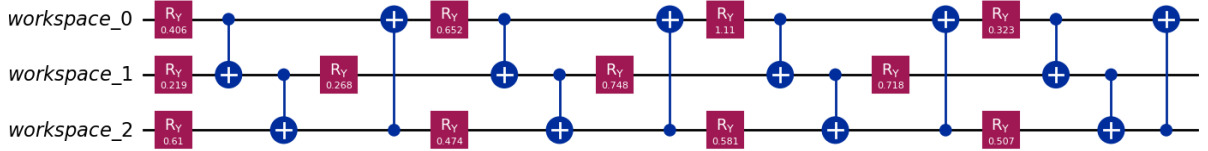


Figure 6: Simplified ansatz (3 qubit, 12 parameters = four layers with three parameters each or, equivalently, four parameters per qubit). For clarity, horizontal lines represent quantum wires which correspond to qubits in the circuit, red squares are the  $R_Y$  gates (see Eq. (81)), blue dots represent control points in controlled gates,  $\oplus$  symbol is used for a controlled- $X$  ( $CNOT$ ) gate.

approach discussed in the previous section presents a challenge: in the general case, the observable  $\hat{O}$  may require an expansion involving an exponential number of Pauli matrices [6]. See the expansion given by Eq. (80) for  $n = 3$  and the discussion afterwards. For this reason, we discuss here a better approach based on diagonalizing the measurement [6]. The key idea is to simplify the observable by transferring relevant information about the problem in the preparation of the state by a proper circuit. In order to do so, we need to introduce first the Quantum Fourier Transform, which is a unitary transformation by construction.

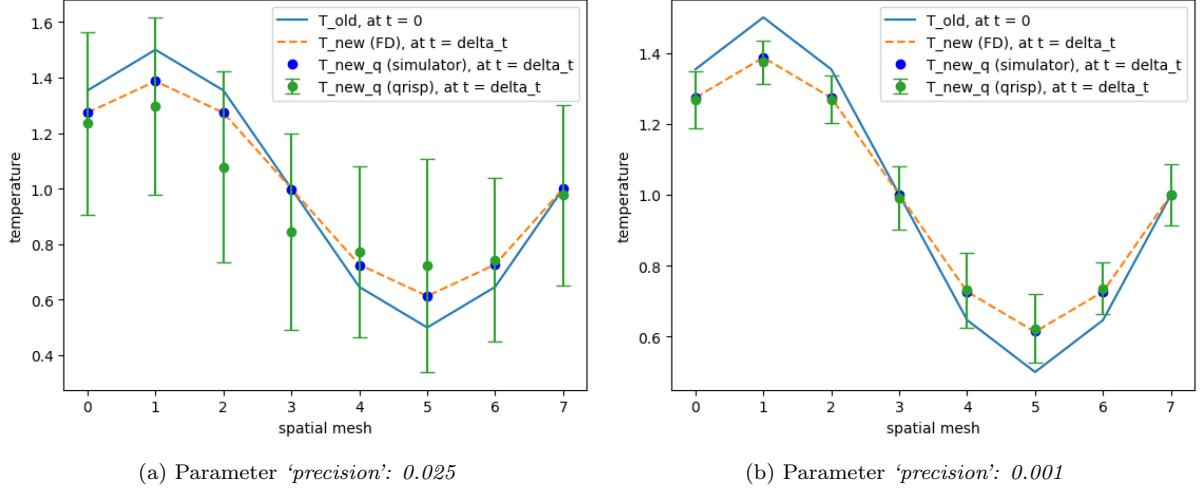


Figure 7: Simulated results obtained by Qrisp example code for implementing the VQE (see Appendix F). In both cases, 30 repetitions are considered for statistics (mean and standard deviation), but with *'precision': 0.025* and *'precision': 0.001* of the performed measurements, respectively. In the first case, the uncertainty is larger than the temperature differences due to the heat conduction time step.

### 3.9.1. Quantum Fourier Transform (QFT)

Before deriving the QFT, let us generalize our nomenclature about the normalization needed to pass from a temperature vector and the corresponding quantum state. A generic quantum state  $|\psi\rangle$  is related to the corresponding temperature vector by a proper scaling factor. For example, according to Eq. (49) and Eq. (67), the following scaling holds  $|b\rangle = (1/\theta)\vec{T}$ , which means that state  $|b\rangle$  is obtained by normalizing  $\vec{T}$  by the scaling factor  $\theta$ . Similarly,  $|x\rangle = (1/\theta^+)\vec{T}^+$ , where it is important to highlight that  $\theta^+ \neq \theta$ . More specifically, the scaling factor  $N/\theta^2$  changes during the simulation. In general, let us define the linear mapping as  $|\psi\rangle = (1/\theta_\psi)\vec{T}^\psi$ . This generalized mapping will be used in the rest of this section.

The Fourier transformation is defined in this document in such a way so as to realize a unitary transformation by construction. Consequently  $\hat{U}_{FT}$  is a unitary matrix, which can be automatically implemented by means of a unitary quantum circuit [2]. For the sake of clarity, let us use in the following  $\hat{U}_{QFT}$ , where  $\hat{U}_{QFT} \equiv \hat{U}_{FT}$ . Because the FT is a linear transformation,  $|\tilde{\psi}\rangle = \hat{U}_{QFT} |\psi\rangle = (1/\theta_\psi)\vec{T}^\psi$  holds too. Consequently,

$$|\tilde{\psi}\rangle = \hat{U}_{QFT} |\psi\rangle, \quad (84)$$

where, for example,  $\psi$  can be  $|b\rangle$  or  $|x\rangle$ . The quantum states  $|\psi\rangle$  and  $|\tilde{\psi}\rangle$  are defined as

$$|\psi\rangle = \sum_{|j\rangle \in \{0,1\}^{\otimes n}} \psi_j |j\rangle, \quad (85)$$

$$|\tilde{\psi}\rangle = \sum_{|j\rangle \in \{0,1\}^{\otimes n}} \tilde{\psi}_j |j\rangle, \quad (86)$$

where the computational basis is the same, as it happens also for the classical case given by Eq. (10) and Eq. (11). Taking into account Eq. (12), the quantum FT of the generic  $l'$ -th amplitude of the transformed state is given by

$$\tilde{b}_{j'} = \frac{1}{\sqrt{N}} \sum_{|j\rangle \in \{0,1\}^{\otimes n}} b_j \omega_N^{j'j}. \quad (87)$$

The problem arises with Eq. (14) because the wavenumber spectrum is not linear with regards to the transformed field. This means that the wavenumber spectrum of a quantum state can not be a quantum state. Therefore, let us define the vector  $\vec{p}$  as the wavenumber spectrum, namely

$$\vec{p}_\psi = \text{Diag}(|\tilde{\psi}\rangle \langle \tilde{\psi}|), \quad (88)$$

where  $\text{Diag}(\cdot)$  is the diagonal extraction operator and  $|\tilde{\psi}\rangle \langle \tilde{\psi}|$  is the density matrix, which – in case of pure states – represents the classical probability distribution over measurement outcomes in the standard basis. It is possible to prove that the wavenumber spectrum defined by Eq. (88) is automatically normalized, namely

$$\sum_{|j'\rangle \in \{0,1\}^{\otimes n}} p_{j'}^\psi = \langle \tilde{\psi} | \tilde{\psi} \rangle = \langle \tilde{\psi} | \hat{U}_{QFT}^\dagger \hat{U}_{QFT} | \tilde{\psi} \rangle = \langle \psi | \psi \rangle = 1. \quad (89)$$

By comparing Eq. (88) with Eq. (14), it is easy to prove that  $\vec{p}_\psi = |\tilde{\psi}\rangle \odot |\tilde{\psi}^*\rangle = (N/\theta_\psi^2) \vec{p}^c$ , where the superscript  $*$  means the complex conjugate. The scaling factor  $N/\theta^2$  changes during the simulation. For example, for the temperature profile reported in Appendix E,  $N/\theta^2 = 8/9$  initially, but it tends to unity when the solution approaches the steady-state temperature profile.

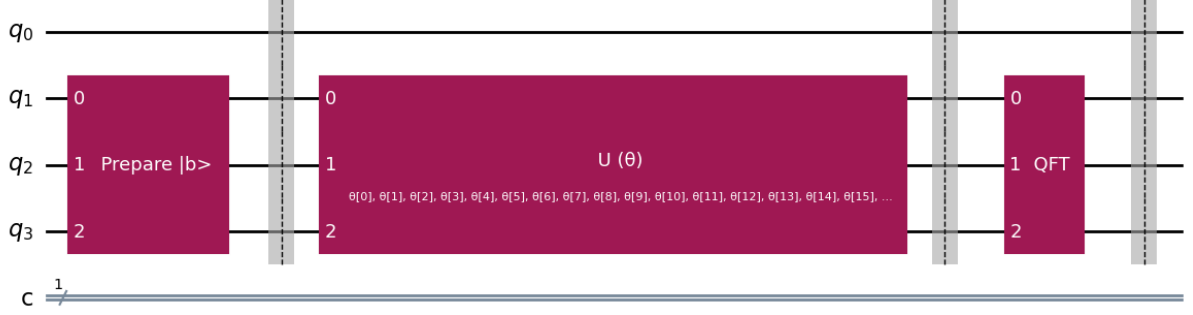
### 3.9.2. Hadamard test approach

Before proceeding with the methodology proposed in Ref. [6], let us first clarify how the loss function given by Eq. (64) is actually computed by a quantum algorithm. First of all, the observable  $\hat{O}$  is represented as a sum of  $N_p$  tensor products of Pauli matrices, as reported in Eq. (73). Secondly, Eq. (63) means that the parametrized solution  $|x(\vec{\theta})\rangle$  is actually computed by means of a unitary transformation  $U(\vec{\theta})$ , coded by the selected ansatz. Let us highlight these implementation features in the definition of the loss function given by Eq. (64), namely

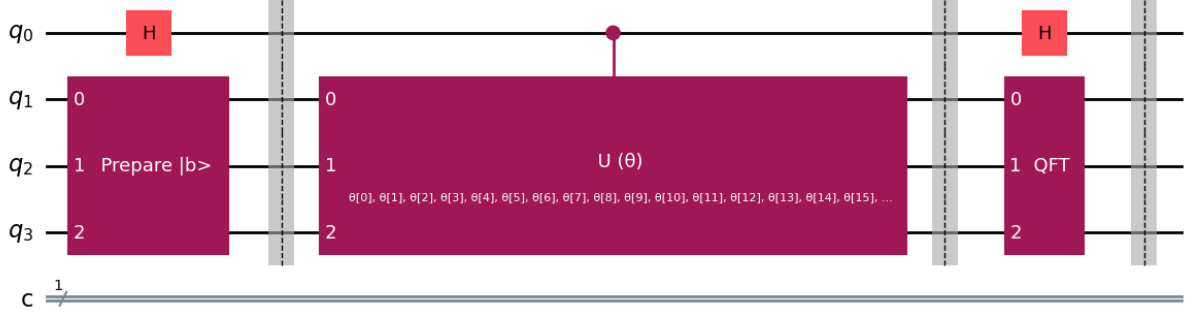
$$L(\vec{\theta}) = \langle x(\vec{\theta}) | \hat{O} | x(\vec{\theta}) \rangle = \mathcal{M}(\hat{O}, U(\vec{\theta})), \quad (90)$$

where  $\mathcal{M}(\hat{O}, U)$  is the measurement protocol for estimating the expectation value of the observable  $\hat{O}$  by means of the circuit  $U$ . Eq. (90) means that the naive implementation, discussed in the previous section, consists in computing the loss function by a direct measurement protocol. Unfortunately, the latter requires large  $N_p$ , i.e. too many tensor products to represent  $\hat{O}$ .

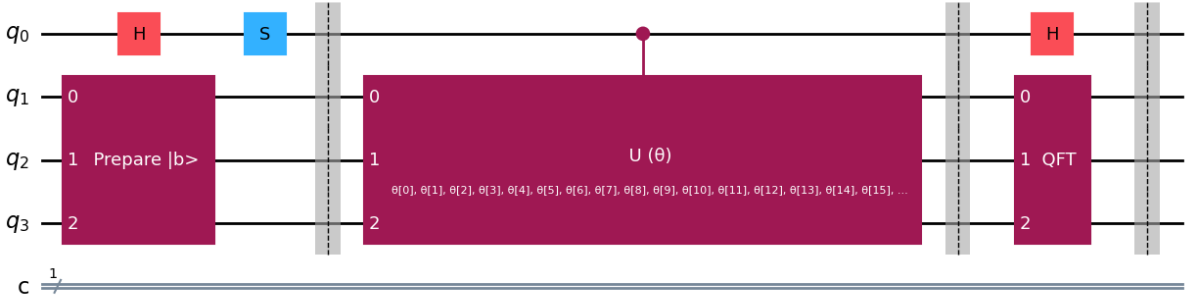
On the other hand, the key idea here is to use the Quantum Fourier Transform (QFT) to simplify the observable  $\hat{O}$  by encoding the relevant information about the problem into



(a) Circuit, which can be described as  $\hat{U}_{QFT} U U_b$ , for preparing the quantum state  $|\tilde{x}\rangle$  (with  $n$  qubits), which is used by the measurement protocol  $\mathcal{M}_{\hat{D}^2}$  given by Eq. (95).



(b) Circuit, which can be described as  $U_{HI}$ , for preparing the quantum state  $|\xi\rangle$  (with  $n + 1$  qubits), which is used by the measurement protocol  $\mathcal{M}_{\hat{D}, \text{Re}}$  given by Eq. (98).



(c) Circuit, which can be described as  $U_{HS}$ , for preparing the quantum state  $|\xi'\rangle$  (with  $n + 1$  qubits), which is used by the measurement protocol  $\mathcal{M}_{\hat{D}, \text{Im}}$  given by Eq. (99).

Figure 8: Quantum circuits used by the measurement protocols  $\mathcal{M}_{\hat{D}^2}$ ,  $\mathcal{M}_{\hat{D}, \text{Re}}$  and  $\mathcal{M}_{\hat{D}, \text{Im}}$ , needed for computing the loss function according to Eq. (100). Note that unitary transformations are graphically represented from left to right, but they apply in the reverse order in the computational formulas [2].

some state preparations via appropriate quantum circuits. This approach avoids the issue of exponential growth in the number of Pauli matrices required for the decomposition of the observable. Let us diagonalize the loss function by recalling the definition of observable given by Eq. (55) and by using the definition of QFT given by Eq. (84), namely,

$$\begin{aligned}
L(\vec{\theta}) &= \langle x(\vec{\theta}) | \hat{C}^T (I - |b\rangle \langle b|) \hat{C} | x(\vec{\theta}) \rangle \\
&= \langle x | \hat{C}^T \hat{C} | x \rangle - \langle x | \hat{C}^T | b \rangle \langle b | \hat{C} | x \rangle \\
&= \langle \tilde{x} | \hat{D}^2 | \tilde{x} \rangle - \langle x | \hat{C}^T | b \rangle \left( \langle x | \hat{C}^T | b \rangle \right)^* \\
&= \langle \tilde{x} | \hat{D}^2 | \tilde{x} \rangle - \left| \langle \tilde{x} | \hat{D} | \tilde{b} \rangle \right|^2
\end{aligned} \tag{91}$$

where the dependence of  $|x(\vec{\theta})\rangle$  on  $\vec{\theta}$  was dropped for the sake of simplicity and  $\hat{D} = \hat{U}_{QFT} \hat{C} \hat{U}_{QFT}^\dagger$ , which means that QFT diagonalizes the conduction operator  $\hat{C}$ , as already discussed in the previous sections. In deriving the last formula, the following property was used:  $\hat{U}_{QFT} \hat{C}^T \hat{U}_{QFT}^\dagger = \hat{D}^T = \hat{D}$ . Moreover, it is worth to recall that  $\langle b | \hat{C} | x \rangle$  is a complex number and that  $\langle b | \hat{C} | x \rangle = (\langle x | \hat{C}^T | b \rangle)^*$ , where  $(\cdot)^*$  is the complex conjugate. The QFT clearly simplifies the observables which are now  $\hat{D}$  and  $\hat{D}^2$ , but it introduces the challenge of efficiently preparing non-trivial states necessary for measuring the terms involved in computing the loss function [6].

Let us identify the measurement protocols for estimating both terms in Eq. (91). Let us start with the first term  $\langle \tilde{x} | \hat{D}^2 | \tilde{x} \rangle$ . Since  $\hat{D}^2$  has real matrix elements and is Hermitian (as it arises from a symmetry), its expectation value is necessarily real [2]. We need to use the selected ansatz in a way which is different from what was done before. We use it to pass from state  $|b\rangle$  to state  $|x\rangle$  for some parameter vector  $\vec{\theta}$ , namely

$$|x(\vec{\theta})\rangle = U(\vec{\theta}) |b\rangle. \tag{92}$$

The previous formula may appear incompatible with Eq. (63). Actually there is no contradiction between Eq. (63) and Eq. (92) because the ansatz is just a sort-of numerical spline, which can realize different transformations by different parameter vectors  $\vec{\theta}$ . Consequently, another transformation must be used to prepare the state  $|b\rangle$ , namely  $|b\rangle = U_b |0\rangle^{\otimes n}$ , which leads to

$$|x(\vec{\theta})\rangle = U(\vec{\theta}) |b\rangle = U(\vec{\theta}) U_b |0\rangle^{\otimes n}. \tag{93}$$

Finally

$$|\tilde{x}\rangle = |\tilde{x}(\vec{\theta})\rangle = \hat{U}_{QFT} U(\vec{\theta}) |b\rangle = \hat{U}_{QFT} U(\vec{\theta}) U_b |0\rangle^{\otimes n}. \tag{94}$$

Consequently the measurement protocol for estimating the first term in Eq. (91) becomes

$$\langle \tilde{x} | \hat{D}^2 | \tilde{x} \rangle = \mathcal{M} \left( \hat{D}^2, \hat{U}_{QFT} U U_b \right) = \mathcal{M}_{\hat{D}^2}, \tag{95}$$

which is depicted in Fig. (8a).

The remaining term  $\langle \tilde{x} | \hat{D} | \tilde{b} \rangle$  in Eq. (91) is more difficult to compute because it is asymmetric with regards to the states which must collapse on the observable  $\hat{D}$ . In this

case, even though  $\hat{D}$  is Hermitian,  $\langle \tilde{x} | \hat{D} | \tilde{b} \rangle$  is not necessarily real. Therefore, we can use two circuits for computing its real and imaginary parts. Following the procedure suggested in Ref. [6], an additional ancilla qubit is added, and some modifications of the standard Hadamard test [2] are properly designed. In particular, the modified Hadamard tests, depicted in Fig. 8b and in Fig. 8c, are used to prepare two states  $|\xi\rangle$  and  $|\xi'\rangle$ , respectively. These quantum states are defined with regards to an enlarged system made of  $n+1$  qubits, where the ancilla qubit is added to the original  $n$  qubits. The quantum circuits preparing these states act as unitary gates, namely

$$|\xi\rangle = |\xi(\vec{\theta})\rangle = U_{HI}(\vec{\theta}) |0\rangle^{\otimes(n+1)}, \quad (96)$$

$$|\xi'\rangle = |\xi'(\vec{\theta})\rangle = U_{HS}(\vec{\theta}) |0\rangle^{\otimes(n+1)}. \quad (97)$$

These quantum states are used in the following measurement protocols (which are proved in the following):

$$\text{Re} \left( \langle \tilde{x} | \hat{D} | \tilde{b} \rangle \right) = \langle \xi | Z \otimes \hat{D} | \xi \rangle = \mathcal{M} \left( Z \otimes \hat{D}, U_{HI} \right) = \mathcal{M}_{\hat{D}, \text{Re}}, \quad (98)$$

$$\text{Im} \left( \langle \tilde{x} | \hat{D} | \tilde{b} \rangle \right) = \langle \xi' | Z \otimes \hat{D} | \xi' \rangle = \mathcal{M} \left( Z \otimes \hat{D}, U_{HS} \right) = \mathcal{M}_{\hat{D}, \text{Im}}, \quad (99)$$

where  $Z$  is one of the Pauli matrices reported in Eq. (74). Before proving the above measurement protocols, it is worth to realize that they can be used to compute the loss function given by Eq. (91), namely

$$\begin{aligned} L(\vec{\theta}) &= \langle \tilde{x} | \hat{D}^2 | \tilde{x} \rangle - \left| \langle \tilde{x} | \hat{D} | \tilde{b} \rangle \right|^2 \\ &= \langle \tilde{x} | \hat{D}^2 | \tilde{x} \rangle - \left[ \text{Re} \left( \langle \tilde{x} | \hat{D} | \tilde{b} \rangle \right) \right]^2 - \left[ \text{Im} \left( \langle \tilde{x} | \hat{D} | \tilde{b} \rangle \right) \right]^2 \\ &= \mathcal{M}_{\hat{D}^2} - \mathcal{M}_{\hat{D}, \text{Re}}^2 - \mathcal{M}_{\hat{D}, \text{Im}}^2. \end{aligned} \quad (100)$$

This is the novel methodology proposed in Ref. [6], which reduces significantly the number of Pauli matrices required for the decomposition of the observable, and hence substitutes the naive measurement protocol reported in Eq. (90). The key idea is to simplify the observables, at the price of making the circuits for generating the measurement states more complex.

The novel methodology given by Eq. (100) is based essentially on the measurement protocols given by Eq. (98) and Eq. (99). In order to prove them, one needs to derive explicitly the states  $|\xi\rangle$  and  $|\xi'\rangle$ . Let us start with  $|\xi\rangle$ , which is prepared by the circuit reported in Fig. 8b. Recalling that the Hadamard gate [2] is given by

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \quad (101)$$

Hence,  $H |0\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$  and  $H |1\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$  because, by convention,  $|0\rangle = (1, 0)^T$  and  $|1\rangle = (0, 1)^T$ . Analyzing the circuit depicted in Fig. 8b yields

$$|\xi\rangle_{\text{1st barrier}} = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \otimes |\tilde{b}\rangle. \quad (102)$$



In this case, the controlled U gate is only applied to the target if the controlled qubit(s) is in the  $|1\rangle$  state, namely

$$U^c = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & U_{11} & U_{12} \\ 0 & 0 & U_{21} & U_{22} \end{pmatrix}. \quad (103)$$

Consequently, according to equation 92, the state at the second barrier in this case becomes:

$$|\xi\rangle|_{\text{2nd barrier}} = \frac{1}{\sqrt{2}} \left( |0\rangle \otimes |\tilde{b}\rangle + |1\rangle \otimes |\tilde{x}\rangle \right). \quad (104)$$

Finally, applying the Hadamard gate again to the previous state yields:

$$\begin{aligned} |\xi\rangle|_{\text{3rd barrier}} = |\xi\rangle &= \frac{1}{\sqrt{2}} \left[ \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \otimes |\tilde{b}\rangle + \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) \otimes |\tilde{x}\rangle \right] = \\ &= \frac{1}{2} |0\rangle \otimes (|\tilde{b}\rangle + |\tilde{x}\rangle) + \frac{1}{2} |1\rangle \otimes (|\tilde{b}\rangle - |\tilde{x}\rangle). \end{aligned} \quad (105)$$

This is a particularly interesting quantum state: (i) because it represents a superposition between the vector of known terms of the linear system  $|\tilde{b}\rangle$  and the solution vector  $|\tilde{x}\rangle$ ; (ii) furthermore, the presence of the ancilla qubit allows us to distinguish between two linear combinations,  $|\tilde{b}\rangle + |\tilde{x}\rangle$  and  $|\tilde{b}\rangle - |\tilde{x}\rangle$ , thereby broadening the range of computations that can be performed with this state. The state exhibits quantum entanglement between the ancilla qubit and the register containing  $|\tilde{b}\rangle$  and  $|\tilde{x}\rangle$ , meaning that measurement of the ancilla directly affects the state of the second register. Having simultaneous access to both  $|\tilde{b}\rangle + |\tilde{x}\rangle$  and  $|\tilde{b}\rangle - |\tilde{x}\rangle$  enables the use of quantum interference to extract global features of the solution, such as inner products or similarity tests. Recalling that  $Z|0\rangle = |0\rangle$  and  $Z|1\rangle = -|1\rangle$ , applying the observable  $Z \otimes \hat{D}$  to  $|\xi\rangle$  yields

$$\begin{aligned} (Z \otimes \hat{D})|\xi\rangle &= \frac{1}{2} Z|0\rangle \otimes \hat{D}(|\tilde{b}\rangle + |\tilde{x}\rangle) + \frac{1}{2} Z|1\rangle \otimes \hat{D}(|\tilde{b}\rangle - |\tilde{x}\rangle) \\ &= \frac{1}{2} |0\rangle \otimes \hat{D}(|\tilde{b}\rangle + |\tilde{x}\rangle) - \frac{1}{2} |1\rangle \otimes \hat{D}(|\tilde{b}\rangle - |\tilde{x}\rangle). \end{aligned} \quad (106)$$

Next, we want to compute  $\langle\xi|(Z \otimes \hat{D})|\xi\rangle$ . The complex conjugate state  $\langle\xi|$  is given by

$$\langle\xi| = \frac{1}{2} \langle 0| \otimes (\langle\tilde{b}| + \langle\tilde{x}|) + \frac{1}{2} \langle 1| \otimes (\langle\tilde{b}| - \langle\tilde{x}|). \quad (107)$$

Consequently the expectation value  $\langle\xi|(Z \otimes \hat{D})|\xi\rangle$  is given by

$$\begin{aligned} \langle\xi|(Z \otimes \hat{D})|\xi\rangle &= \frac{1}{4} \left[ \langle 0| \otimes (\langle\tilde{b}| + \langle\tilde{x}|) \right] \left[ |0\rangle \otimes \hat{D}(|\tilde{b}\rangle + |\tilde{x}\rangle) \right] \\ &\quad - \frac{1}{4} \left[ \langle 0| \otimes (\langle\tilde{b}| + \langle\tilde{x}|) \right] \left[ |1\rangle \otimes \hat{D}(|\tilde{b}\rangle - |\tilde{x}\rangle) \right] \\ &\quad + \frac{1}{4} \left[ \langle 1| \otimes (\langle\tilde{b}| - \langle\tilde{x}|) \right] \left[ |0\rangle \otimes \hat{D}(|\tilde{b}\rangle + |\tilde{x}\rangle) \right] \\ &\quad - \frac{1}{4} \left[ \langle 1| \otimes (\langle\tilde{b}| - \langle\tilde{x}|) \right] \left[ |1\rangle \otimes \hat{D}(|\tilde{b}\rangle - |\tilde{x}\rangle) \right]. \end{aligned} \quad (108)$$

The second and the third term in the previous expression are null, because  $\langle 0|1\rangle = 0$  and  $\langle 1|0\rangle = 0$ . Taking into account that  $\langle 0|0\rangle = 1$  and  $\langle 1|1\rangle = 1$  yields

$$\begin{aligned}\langle \xi | (Z \otimes \hat{D}) | \xi \rangle &= \frac{1}{4} \left( \langle \tilde{b} | \hat{D} | \tilde{b} \rangle + \langle \tilde{b} | \hat{D} | \tilde{x} \rangle + \langle \tilde{x} | \hat{D} | \tilde{b} \rangle + \langle \tilde{x} | \hat{D} | \tilde{x} \rangle \right) \\ &\quad - \frac{1}{4} \left( \langle \tilde{b} | \hat{D} | \tilde{b} \rangle - \langle \tilde{b} | \hat{D} | \tilde{x} \rangle - \langle \tilde{x} | \hat{D} | \tilde{b} \rangle + \langle \tilde{x} | \hat{D} | \tilde{x} \rangle \right) \\ &= \frac{1}{2} \left( \langle \tilde{b} | \hat{D} | \tilde{x} \rangle + \langle \tilde{x} | \hat{D} | \tilde{b} \rangle \right).\end{aligned}\tag{109}$$

Taking into account that

$$(\langle \tilde{x} | \hat{D} | \tilde{b} \rangle)^* = \langle \tilde{b} | \hat{D}^\dagger | \tilde{x} \rangle = \langle \tilde{b} | \hat{D} | \tilde{x} \rangle,\tag{110}$$

it is possible to derive the following property

$$\langle \tilde{b} | \hat{D} | \tilde{x} \rangle + \langle \tilde{x} | \hat{D} | \tilde{b} \rangle = 2 \operatorname{Re} \left( \langle \tilde{x} | \hat{D} | \tilde{b} \rangle \right).\tag{111}$$

Consequently, using Eq. (111) into Eq. (109) yields

$$\langle \xi | (Z \otimes \hat{D}) | \xi \rangle = \operatorname{Re} \left( \langle \tilde{x} | \hat{D} | \tilde{b} \rangle \right) = \mathcal{M}_{\hat{D}, \operatorname{Re}},\tag{112}$$

which is the desired result for the measurement protocol  $\mathcal{M}_{\hat{D}, \operatorname{Re}}$ .

On the other hand, let us proceed with  $|\xi'\rangle$ , which is prepared by the circuit reported in Fig. 8c. Recalling that the phase gate [2] is given by

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}.\tag{113}$$

Hence,  $S|0\rangle = |0\rangle$  and  $S|1\rangle = i|1\rangle$ . Analyzing the circuit depicted in Fig. 8c yields

$$|\xi'\rangle|_{\text{1st barrier}} = \frac{1}{\sqrt{2}} (|0\rangle + i|1\rangle) \otimes |\tilde{b}\rangle,\tag{114}$$

$$|\xi'\rangle|_{\text{2nd barrier}} = \frac{1}{\sqrt{2}} \left( |0\rangle \otimes |\tilde{b}\rangle + i|1\rangle \otimes |\tilde{x}\rangle \right),\tag{115}$$

$$\begin{aligned}|\xi'\rangle|_{\text{3rd barrier}} = |\xi'\rangle &= \frac{1}{\sqrt{2}} \left[ \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \otimes |\tilde{b}\rangle + \frac{i}{\sqrt{2}} (|0\rangle - |1\rangle) \otimes |\tilde{x}\rangle \right] = \\ &= \frac{1}{2} |0\rangle \otimes (|\tilde{b}\rangle + i|\tilde{x}\rangle) + \frac{1}{2} |1\rangle \otimes (|\tilde{b}\rangle - i|\tilde{x}\rangle).\end{aligned}\tag{116}$$

The same considerations about the unique properties of the state  $|\xi\rangle$  hold as well for the state  $|\xi'\rangle$ . Proceeding in the same way discussed above, the expectation value  $\langle \xi' | Z \otimes \hat{D} | \xi' \rangle$  is given by

$$\langle \xi' | Z \otimes \hat{D} | \xi' \rangle = \frac{i}{2} \left( \langle \tilde{b} | \hat{D} | \tilde{x} \rangle - \langle \tilde{x} | \hat{D} | \tilde{b} \rangle \right).\tag{117}$$

Taking into account again Eq. (110), it is possible to derive the following property

$$\langle \tilde{b} | \hat{D} | \tilde{x} \rangle - \langle \tilde{x} | \hat{D} | \tilde{b} \rangle = \left( \langle \tilde{x} | \hat{D} | \tilde{b} \rangle \right)^* - \langle \tilde{x} | \hat{D} | \tilde{b} \rangle = -2i \operatorname{Im} \left( \langle \tilde{x} | \hat{D} | \tilde{b} \rangle \right). \quad (118)$$

Consequently, using Eq. (118) into Eq. (117) yields

$$\langle \xi' | Z \otimes \hat{D} | \xi' \rangle = \operatorname{Im} \left( \langle \tilde{x} | \hat{D} | \tilde{b} \rangle \right) = \mathcal{M}_{\hat{D}, \operatorname{Im}}, \quad (119)$$

which is again the desired result for the measurement protocol  $\mathcal{M}_{\hat{D}, \operatorname{Im}}$ .

## 4. Conclusions

At the current stage of technological development, predicting the potential impact of quantum computing on Thermal Science remains extremely challenging, as it depends on future advancements. As a paradigmatic case, we focused on solving the heat conduction equation, with the starting point being the development of algorithms that leverage quantum computing most effectively for this application.

In these notes, we began by analyzing the Variational Quantum Eigensolver (VQE) algorithm, as it establishes a crucial connection between solving linear systems of equations – common in Thermal Science – and finding the ground state of quantum systems, a fundamental problem that provides deeper insight into quantum mechanics. While VQE faces practical challenges for implementation on real quantum computers, the complexity of decomposing the target observable into Pauli matrices depends on the specific problem. For instance, in molecular Hamiltonian functions, the number of Pauli strings typically scales as  $\sim n^4$ , which is still computationally demanding but not exponential. Despite these challenges, with appropriate techniques, VQE may still be applicable [12].

More advanced algorithms will also need to be analyzed. Promising alternatives include the Quantum Approximate Optimization Algorithm (QAOA) for combinatorial optimization tasks, the Quantum Linear System Algorithm (QLSA) (such as HHL) for efficiently solving linear equations, and hybrid quantum-classical methods that combine classical preconditioning techniques with quantum solvers to mitigate computational bottlenecks.

## Acknowledgments

Portions of this work were developed with the assistance of ChatGPT, an AI language model by OpenAI, in accordance with the CC-BY 4.0 license.

## Appendix A. Two-athlete strategy

Let us imagine that two athletes must both participate in a preliminary qualifying tournament to advance to the final stage of a sports competition. Unfortunately, the first athlete has not had enough time to train properly and therefore has a 20% chance of qualifying, while the second athlete has prepared adequately and thus has a 80% chance of achieving the same result. Let us indicate by  $|1\rangle$  the qualified state for the final stage and by  $|0\rangle$  the unqualified state, after the measurement certified by the preliminary tournament. In this regard, the state of the two athletes, before the preliminary tournament, can be expressed by the superposition of unqualified state  $|0\rangle$  and qualified state  $|1\rangle$ , namely

$$|\psi_0\rangle = \sqrt{0.8} |0\rangle + \sqrt{0.2} |1\rangle, \quad (\text{A.1})$$

$$|\psi_1\rangle = \sqrt{0.2} |0\rangle + \sqrt{0.8} |1\rangle. \quad (\text{A.2})$$

If the two athletes compete in the qualifying tournament independently, the expected outcome will be

$$|\psi_0\rangle \otimes |\psi_1\rangle = \sqrt{0.16} |00\rangle + \sqrt{0.64} |01\rangle + \sqrt{0.04} |10\rangle + \sqrt{0.16} |11\rangle. \quad (\text{A.3})$$

The previous formula means that there is a 64% probability that the under-prepared athlete does not qualify while the well-trained athlete qualifies, which is the most probable outcome of the tournament. The opposite outcome changing both predictions at the same time is quite unlikely (4% probability). Mixed outcomes, where one event is aligned with the most likely expectation and the other one changing the expected outcome, have the same (intermediate) 16% probability. This outcome can be obtained also by sampling properly a purposely-designed quantum circuit. Let us assign a qubit for each athlete, i.e.  $q_0$  and  $q_1$  respectively. For each athlete, let us design a proper gate representing the training, which realizes a single-qubit rotation about the  $Y$ -axis and ensures the expected performance probability according to Eq. (31). The obtained quantum circuit and the corresponding simulated results are reported in Fig. (A.9). It is possible to prove that the previous predictions are correct by recalling the general formula for combining the probabilities of uncorrelated events by tensor product, namely

$$|\psi_0\rangle \otimes |\psi_1\rangle = \sqrt{p_0^{[0]} p_1^{[0]}} |00\rangle + \sqrt{p_0^{[0]} p_1^{[1]}} |01\rangle + \sqrt{p_0^{[1]} p_1^{[0]}} |10\rangle + \sqrt{p_0^{[1]} p_1^{[1]}} |11\rangle. \quad (\text{A.4})$$

Putting aside the sports for a while, the previous formula is consistent with the kinetic theory of gases and, in particular, with the assumption of molecular chaos in deriving the Boltzmann equation (also known as the *Stosszahlansatz*), which states that before a collision occurs, the velocities of two colliding particles are uncorrelated.

Coming back to the example, let us suppose now that the better-prepared athlete decides to help the less-prepared one by sharing advice on how to tackle the various challenges of the qualification tournament and perhaps provides some insights about their opponents. This time, the chances of success for the less-prepared athlete increase significantly. However, there is also a price to pay: in some cases, the better-prepared athlete may provide misleading advice, leading to failures that would not have occurred otherwise. Now the state

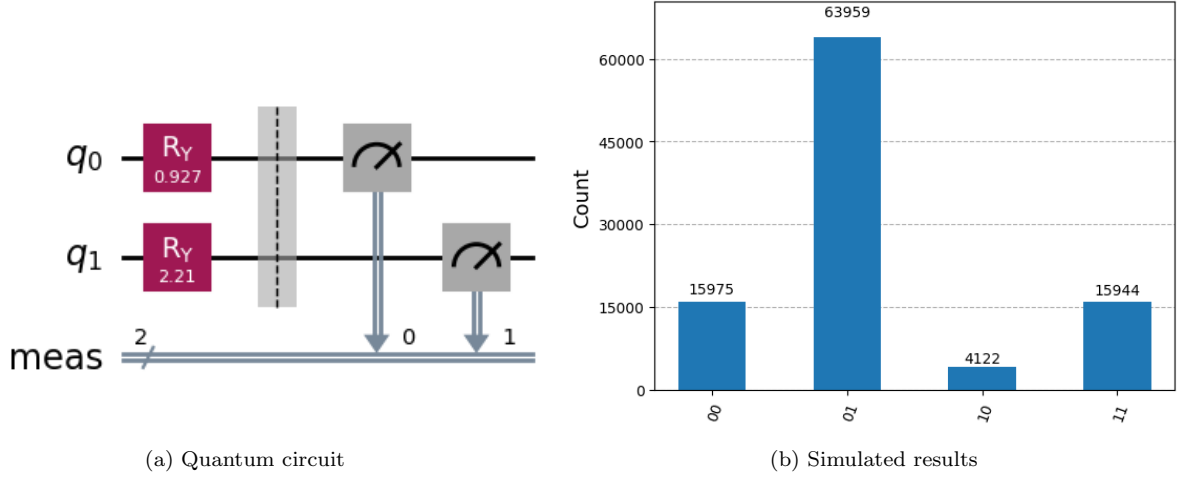


Figure A.9: Uncorrelated athletes/qubits  $q_0$  and  $q_1$ . For each athlete, a proper gate represents the training, which realizes a single-qubit rotation about the  $Y$ -axis ( $R_Y$  gate) and ensures the expected performance probability according to Eq. (31). The dashed vertical line is used to demarcate logical gates from measurement.

representing the tournament outcome for the two athletes becomes correlated (i.e. entangled). Because of entanglement, when the well-trained athlete qualifies (i.e. the second qubit is equal to  $|1\rangle$ ) then the athlete is convincing enough to flip the other athlete's performance outcome (from 0 to 1 but also vice versa). This means that the probabilities of the outcome  $|01\rangle$  and  $|11\rangle$  are swapped, namely

$$|\psi_0\psi_1\rangle = \sqrt{0.16} |00\rangle + \sqrt{0.16} |01\rangle + \sqrt{0.04} |10\rangle + \sqrt{0.64} |11\rangle. \quad (\text{A.5})$$

Now the probability that both athletes qualify is increased to 64%, meaning that this strategy is anyway advantageous. In this second case, the quantum circuit must be updated by adding a controlled *NOT* gate (also called controlled- $X$  gate or *CNOT* gate): as already pointed out, the *CNOT* gate implies that, whenever  $q_1$  equals one, the athlete is persuasive enough to reverse the other's performance outcome (switching between 0 and 1 but also vice versa). The quantum circuit and the corresponding simulated results in this second case are reported in Fig. (A.10). The key point is that, in presence of correlation – also called entanglement –, the outcome state is not separable, which means that it cannot be expressed as the tensor product of two independent states, namely  $|\psi_0\psi_1\rangle \neq |\psi_0^? \rangle \otimes |\psi_1^? \rangle$ , where  $|\psi_0^? \rangle$  and  $|\psi_1^? \rangle$  are hypothetical states which do not exist. In order to prove that such separated states do not exist, let us compare Eq. (A.5) with the tensor product definition given by Eq. (A.4), namely

$$p_0^{|0\rangle?} p_1^{|0\rangle?} = 0.16, \quad (\text{A.6})$$

$$p_0^{|0\rangle?} p_1^{|1\rangle?} = 0.16, \quad (\text{A.7})$$

$$p_0^{|1\rangle?} p_1^{|0\rangle?} = 0.04, \quad (\text{A.8})$$

$$p_0^{|1\rangle?} p_1^{|1\rangle?} = 0.64. \quad (\text{A.9})$$

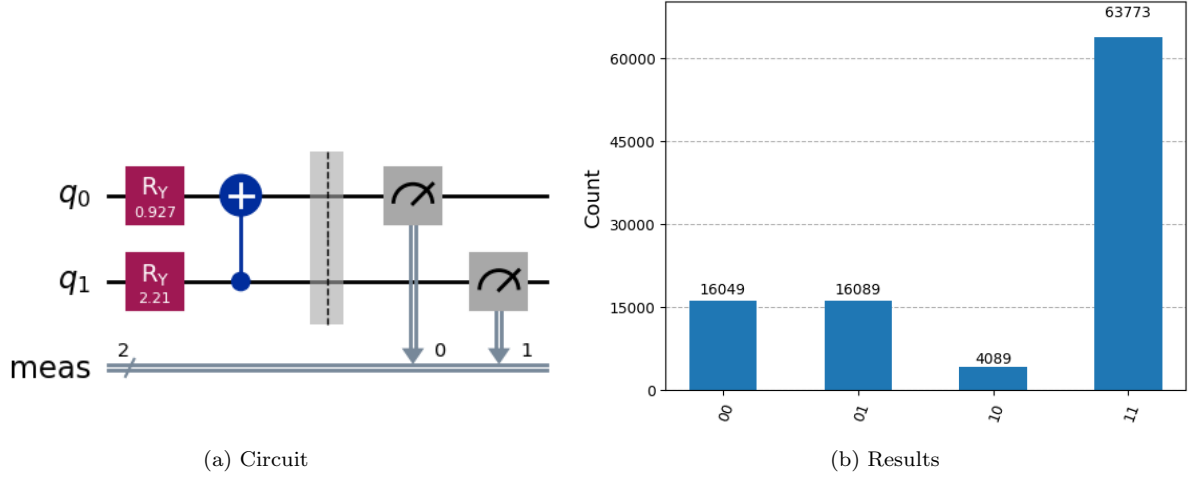


Figure A.10: Correlated athletes/qubits due to entanglement. In this second case, the quantum circuit was updated by adding a controlled *NOT* gate (also called controlled-*X* gate or *CNOT* gate): every time that  $q_1$  is equal to one, the athlete is convincing enough to flip the second athlete's performance outcome (from 0 to 1 but also vice versa).

Let us combine Eq. (A.6) and Eq. (A.7), which yields  $p_1^{[0]?} = p_1^{[1]?}$ , where  $p_0^{[0]?}$  was simplified. The last relation can be used in Eq. (A.8) to yield

$$p_0^{[1]?} p_1^{[1]?} = 0.04, \quad (\text{A.10})$$

$$p_0^{[1]?} p_1^{[1]?} = 0.64, \quad (\text{A.11})$$

which are two relations clearly incompatible with each other. Hence, the state given by Eq. (A.5) is not separable because the two athletes are entangled. This metaphor of the two-athlete strategy aligns well with the numerical example shown in Fig. 2, which can therefore also be interpreted as a visual representation of the current example.

## Appendix B. Prisoner's dilemma

In the classical prisoner's dilemma, two players (Alice and Bob) must independently decide whether to Cooperate (C) or Defect (D). If both cooperate, they receive a moderate penalty (e.g., 1 year in prison each). If one defects while the other cooperates, the defector goes free (0 years) while the cooperator gets the maximum penalty (3 years in prison). If both defect, they receive a higher penalty (typically 2 years in prison each). In classical game theory, defection is the dominant strategy, leading to a situation where both players receive a worse outcome than if they had cooperated.

Now, suppose Alice and Bob behave as an entangled Bell state [2]. This state introduces non-classical correlations between their choices, namely

$$|\psi\rangle = \frac{1}{\sqrt{2}} (|CC\rangle + |DD\rangle). \quad (\text{B.1})$$

In this state, their decisions are no longer independent: if Alice is measured and goes for cooperation, Bob's measurement in the same basis will necessarily yield cooperation as well, and the same holds for defection. By leveraging quantum operations, Alice and Bob can reach a new equilibrium where cooperation becomes as likely as defection, leading to a better outcome than in the classical case. Quantum entanglement thus enhances cooperation and offers a possible resolution to the prisoner's dilemma beyond classical strategies.

### Appendix C. Hilbert space versus Bloch sphere

Here, we aim to intuitively explain the construction of the Bloch sphere representation of a single qubit, and its relationship to the Hilbert space representation. As previously discussed, a single qubit state vector can be expressed using Eq. (30):

$$|\psi_q\rangle = \delta_q^{[0]} |0\rangle + \delta_q^{[1]} |1\rangle,$$

where  $\delta_q^{[0]}, \delta_q^{[1]} \in \mathbb{C}$  are complex numbers. These coefficients define a state in a two-dimensional Hilbert space over the complex numbers, which can be thought of as having four real parameters; thus, even in this simple case, it is difficult to visualize the state. Fortunately, we can overcome this limitation by using the polar (Euler) representation of complex numbers and applying the normalization condition,  $|\delta_q^{[0]}|^2 + |\delta_q^{[1]}|^2 = 1$ . We first express the complex amplitudes as  $\delta_q^{[0]} = Ae^{i\alpha}$  and  $\delta_q^{[1]} = Be^{i\beta}$ . Substituting into Eq. (30), we obtain:

$$|\psi_q\rangle = Ae^{i\alpha} |0\rangle + Be^{i\beta} |1\rangle. \quad (\text{C.1})$$

The normalization condition constrains the amplitudes to lie on the unit circle in real two-dimensional space:  $A^2 + B^2 = 1$ .

Therefore, we can parameterize the amplitudes using a single *polar* angle  $\theta$ :

$$|\psi_q\rangle = \cos \theta e^{i\alpha} |0\rangle + \sin \theta e^{i\beta} |1\rangle.$$

Since an overall (global) phase factor has no physical effect, we can factor out  $e^{i\alpha}$  and ignore it, defining a relative phase  $\zeta = \beta - \alpha$ . We then obtain:

$$|\psi_q\rangle = e^{i\alpha} (\cos \theta |0\rangle + \sin \theta e^{i\zeta} |1\rangle),$$

where the term  $e^{i\alpha}$  can be safely omitted, since it represents a phase shift, and does not affect measurement outcomes [2], yielding the simplified and physically equivalent expression:

$$|\psi_q\rangle = \cos \theta |0\rangle + \sin \theta e^{i\zeta} |1\rangle. \quad (\text{C.2})$$

Comparing Eq. (C.2) with Eq. (31) in the main text yields  $\theta = \varphi_q/2$  and  $\zeta = \zeta_q$ , which will be clearer at the end of this appendix and is the main point of this derivation. Now, using Euler's formula  $e^{i\zeta} = \cos \zeta + i \sin \zeta$ , we can rewrite the qubit state:

$$|\psi_q\rangle = \cos \theta |0\rangle + \sin \theta \cos \zeta |1\rangle + i \sin \theta \sin \zeta |1\rangle,$$

and comparing to the spherical coordinate vector  $\vec{r} = (\sin \theta \cos \zeta, \sin \theta \sin \zeta, \cos \theta)^T$ . We recognize that it is possible to visualize the state  $|\psi_q\rangle$  in a tridimensional Hilbert space with the bases  $|1\rangle = (1, 0, 0)^T$ ,  $i|1\rangle = (0, i, 0)^T$  (the generic axis of the Hilbert space may be complex, but the inner product must be still non-commutative<sup>3</sup>),  $|0\rangle = (0, 0, 1)^T$ , as shown in Fig. C.11(a). Using the full sphere in Hilbert space has the problem that multiple states

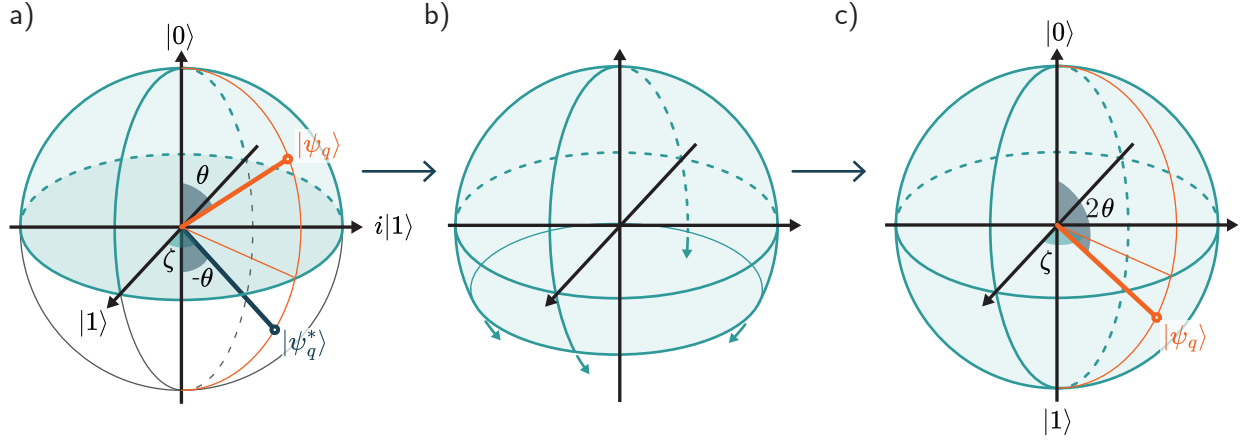


Figure C.11: Intuitive construction of the Bloch sphere. (a) The qubit vector state  $|\psi_q\rangle$  (and the reflected state  $|\psi_q^*\rangle$ ) represented in Hilbert space. (b) The conceptual stretching of the upper hemisphere onto a sphere (and the equator of the upper hemisphere in a point). (c) The final representation of  $|\psi_q\rangle$  as a vector state on the Bloch sphere.

can result in the same measurement outcomes. For example, consider the state  $|\psi_q\rangle$  and the reflected state through the equatorial plane  $|\psi_q^*\rangle = \cos(-\theta)|0\rangle + \sin(-\theta)e^{i\varphi}|1\rangle$ , depicted in Fig. C.11(a), which yield the same measurement probabilities:

$$\begin{aligned} p_0 &= \langle \psi_q | P_0 | \psi_q \rangle = \langle \psi_q^* | P_0 | \psi_q^* \rangle = \cos^2 \theta, \\ p_1 &= \langle \psi_q | P_1 | \psi_q \rangle = \langle \psi_q^* | P_1 | \psi_q^* \rangle = \sin^2 \theta, \end{aligned}$$

where the projectors operators of the measurement are defined as  $P_0 = |0\rangle\langle 0|$  and  $P_1 = |1\rangle\langle 1|$ . Thus, we see that  $|\psi_q\rangle$  and  $|\psi_q^*\rangle$  are physically indistinguishable by measurement. As a result, only the *upper hemisphere* of the sphere in the Hilbert space is needed to uniquely describe a qubit state, which corresponds to restricting the angle  $\theta$  to the interval  $\theta \in [0, \pi/2]$ .

Furthermore, all the states that lie along the equator ( $\theta = \pi/2$  in the Hilbert space) represent the same measurement outcome (i.e., for  $\theta = \pi/2$  we have  $|\psi'_q\rangle = \cos \zeta |1\rangle + \sin \zeta i |1\rangle$ , so the measurement probability is  $p_1 = \langle \psi'_q | P_1 | \psi'_q \rangle = \cos^2 \zeta + \sin^2 \zeta = 1$ ). Therefore, we can conceptually imagine “pulling” this circumference downward until it collapses into a single

<sup>3</sup>In a two-dimensional complex space, the inner product is non-commutative because it is conjugate symmetric, meaning  $\langle \psi | \phi \rangle = \langle \phi | \psi \rangle^*$  (complex conjugate), so swapping the vectors changes the result unless the inner product is real. For example, if  $|a\rangle = |1\rangle$  and  $|b\rangle = i|1\rangle$ , then  $\langle a | b \rangle = i \neq \langle b | a \rangle = -i$ , therefore  $\langle a | b \rangle = \langle b | a \rangle^*$ .



point. This transformation simplifies the visualization and results in the familiar Bloch sphere representation, as illustrated in Fig. C.11(b) and Fig. C.11(c). In this representation, the polar angle  $\theta$  from the Hilbert space mapping is effectively *doubled* on the Bloch sphere. To maintain consistency between the two representations, we can introduce the Bloch polar angle  $\varphi = 2\theta \Leftrightarrow \theta = \varphi/2$ , with  $\varphi \in [0, \pi]$ , as used in Eq. (31).

It is important to note that this is an illustrative non-rigorous explanation of the Bloch sphere construction. Mathematically, the Bloch sphere corresponds to the complex projective line of the two-dimensional Hilbert space, constructed using a stereographic projection of the qubit state onto a plane and topologically represented as the *Riemann sphere*.

## Appendix D. Real data loading/encoding

A fundamental aspect of quantum computing is the ability to efficiently load real-world data into a quantum system and extract results back to classical computing. Naively one would like to have a procedure for correlating the parameters in  $\vec{\theta}$  with the real amplitudes in  $|x(\vec{\theta})\rangle$  by means of some analytical formulas. This approach is usually called real data loading or better encoding, and some algorithms have been proposed in literature, e.g. a divide-and-conquer algorithm for quantum state preparation [10]. Even though, for optimization problems (including VQE), real data loading/encoding is not strictly necessary, it is a good way to understand how a quantum computer works and hence it will be discussed here.

Loading real-world data into a quantum system requires a quantum state preparation. Many algorithms to create arbitrary quantum states require quantum circuits with depth  $O(N)$  to load an  $N$ -dimensional vector [10]. In the context of quantum circuits, depth refers to the number of sequential (time-ordered) layers of quantum gates that must be applied to execute an algorithm. It measures how many steps a quantum circuit takes to process information. Some algorithms have been proposed in the literature based on a divide-and-conquer strategy to load a  $N$ -dimensional vector using a quantum circuit with poly-logarithmic depth [10]. The problem is that these algorithms usually require a large number of parameters to compute and therefore are less suitable for variational problems as VQE.

The divide-and-conquer paradigm is used in efficient algorithms for sorting, computing the discrete Fourier transform, and others [10]. The main idea is to divide a problem into subproblems of the same class and combine the solutions of the subproblems, in a recursive way, to obtain the solution of the original problem [10]. In particular, one of the standard methods for loading information in a quantum device is based on using controlled rotations [10]. These controlled rotations can be designed by means of a proper binary tree data structure for the data to be loaded and consequently, by means of Eq. (31), for the expected rotations. In the case of  $n = 3$  qubits, the resulting quantum circuit, which consists solely of rotations about the  $Y$ -axis ( $R_Y$  gates), is shown in Fig. D.12. As an example of its application, let us suppose to load the following real data

$$L_l = \frac{1}{N} + \frac{1}{2N} \sin\left(\frac{2\pi}{N}l + \frac{2\pi}{N}\right), \quad l = 0, 1, \dots, N-1, \quad (\text{D.1})$$

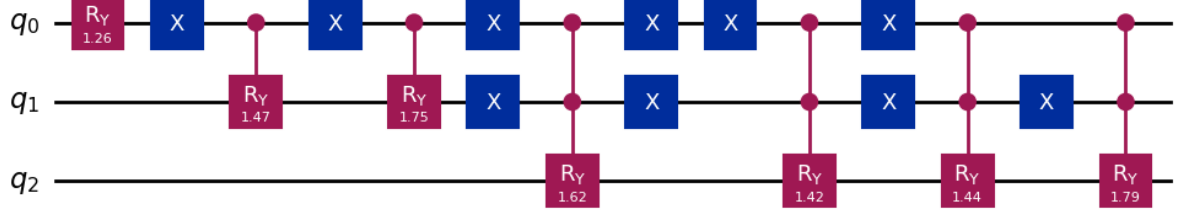


Figure D.12: Quantum circuit for loading real data in a quantum system state  $|x\rangle$  based on a divide-and-conquer algorithm proposed in Ref. [10]. Note that the proposed circuit uses only rotations about the Y-axis ( $R_Y$  gates), which are designed by means of a proper binary tree data structure [10]. For clarity, horizontal lines represent quantum wires which correspond to qubits in the circuit, red squares are the  $R_Y$  gates (see Eq. (81)), blue squares are the  $X$  gates ( $NOT$  gates), red dots represent control points in controlled gates.

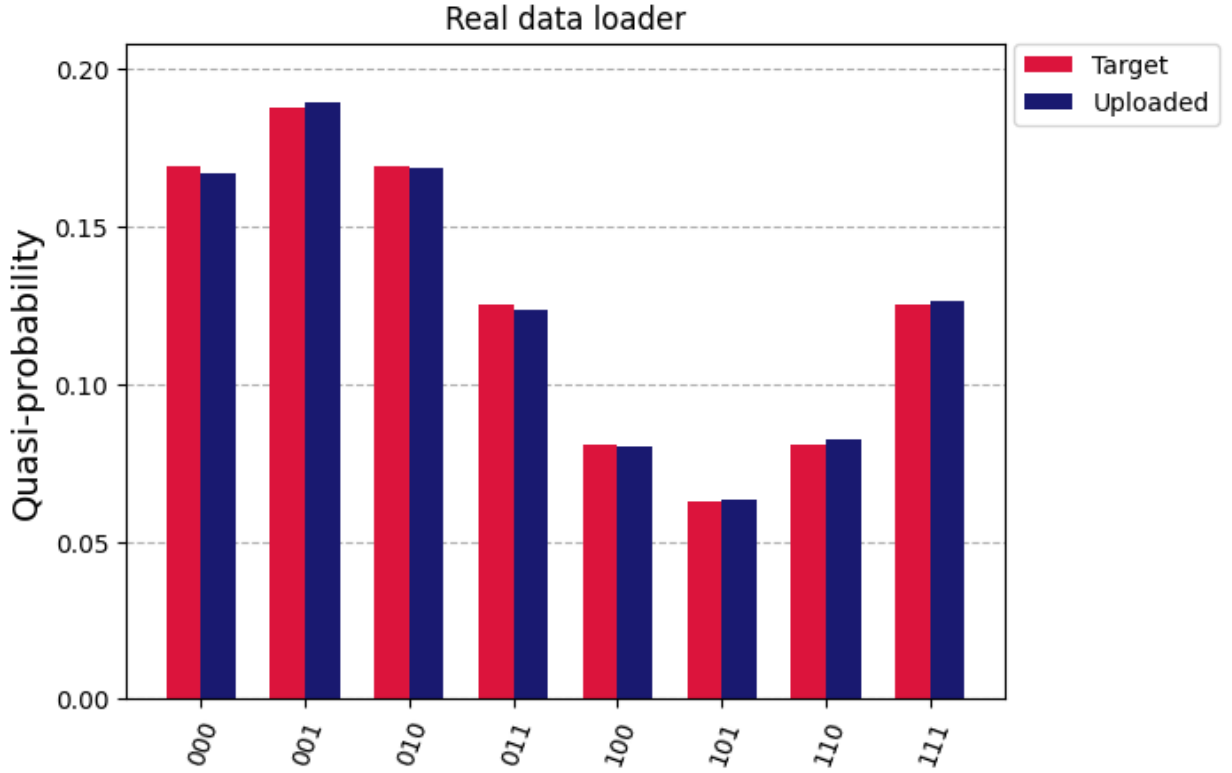


Figure D.13: Example of real data loading / encoding (simulated). The red bars represent the real data to be loaded. The blue bar represents the actual loaded data by a divide-and-conquer algorithm proposed in Ref. [10]. These results are based on a quantum simulator provided in the Qiskit package.

where the elements  $L_l$  are designed such that  $\sum_l L_l = 1$ . The last condition makes possible to load the previous data as (quasi) probabilities of a quantum state, namely

$$|x_L\rangle = \sum_{|j\rangle \in \{0,1\}^{\otimes n}} \sqrt{L_j} |j\rangle. \quad (\text{D.2})$$

Please note that this is different from what is done in the main text (see section 3.2), where the elements of vector  $\vec{T}$  are loaded as amplitudes of a quantum state  $|x_T\rangle = \sum_j (T_j / \sqrt{\vec{T} \cdot \vec{T}}) |j\rangle$ , e.g. see Eq. (50). In quantum computing, real data can be loaded into quantum states either as probabilities or amplitudes, each with distinct implications. Probability encoding represents data as a quantum probability distribution, where measurement outcomes follow predefined likelihoods, making it useful for probabilistic modeling and quantum sampling. In contrast, amplitude encoding directly maps data values into quantum state amplitudes, enabling powerful applications in quantum machine learning and linear algebra but requiring more complex state reconstruction. While probability encoding is more intuitive and measurement-friendly, amplitude encoding offers greater expressive power for quantum algorithms. Coming back to the original data loading/encoding problem, in case of  $N = 8$ , the values  $L_l$  can be encoded into a three-qubit quantum state  $|L\rangle$  using the proposed circuit by means of  $N_\theta^{d\&c} = 7$  parameters. The quasi-probabilities obtained from simulating this circuit, based on 100,000 measurements of system replicas, are presented in Fig. D.13, demonstrating the circuit’s effectiveness in accurately encoding the target real data.

In the divide-and-conquer approach, the good point is that the parameters  $\vec{\theta}$  have a clear physical interpretation, thanks to the binary tree data structure, and they can be analytically computed to recover the target data to be loaded. The problem is that the number of parameters in this ansatz grows as  $N_\theta^{d\&c} = 2^n - 1$ , which is as large as the number of real data to be loaded (recall that  $\langle x|x \rangle = 1$ ). For comparison, the efficient ansatz discussed in section 3.7.2, called “*EfficientSU2*” circuit in Qiskit [4], requires instead a number of parameters which grows only linearly with the number of qubits, namely  $N_\theta = 8n$ . It is clear that, in case of a large number of qubits, the divide-and-conquer approach is less efficient than “*EfficientSU2*” – and therefore impractical – for variational problems because  $N_\theta^{d\&c} \gg N_\theta$ .

Clearly, the problem of data loading and encoding is quite general and extends far beyond variational applications. In particular, all software development kits have efficient routines for performing this task. For example, Qiskit [4] has a state preparation routine based on the decomposition of arbitrary isometries into a sequence of single-qubit and controlled-not (*CNOT*) gates [13]. This approach is tested here for loading the data given by Eq. (D.1) in case of  $N = 8$  (3 qubits) and  $N = 16$  (4 qubits) on a real hardware. In particular, let us use the IQM Garnet machine developed by IQM, a Finnish-German quantum computer manufacturer. The experimental results (based on 20,000 shots) of the state preparation algorithm proposed in Ref. [13] on IQM Garnet are reported in Fig. D.14. These results should be considered indicative, as real hardware in the NISQ era is influenced by environmental conditions, causing actual outcomes to vary slightly from run to run.

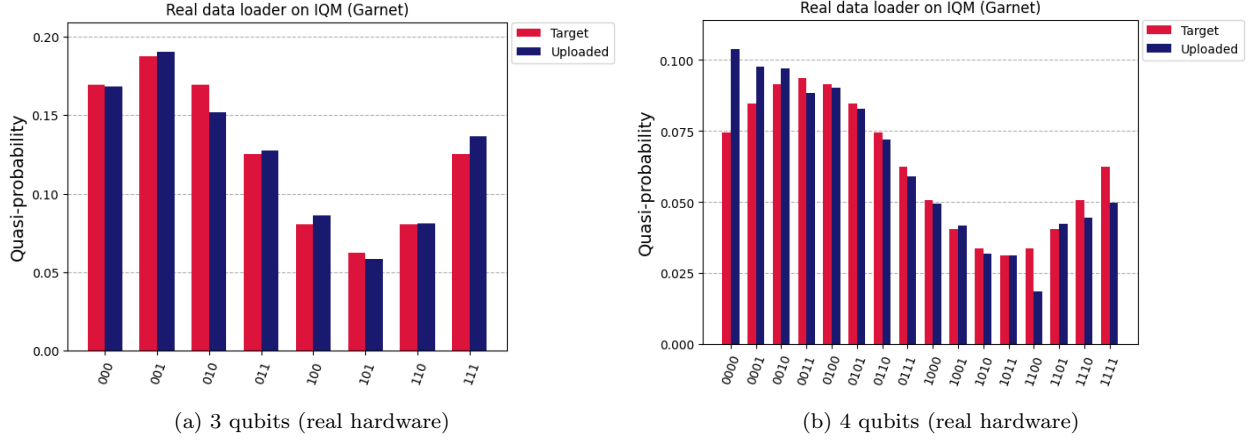


Figure D.14: Example of real data loading / encoding (real hardware). The red bars represent the real data to be loaded. The blue bar represents the actual loaded data by a divide-and-conquer algorithm proposed in Ref. [10]. These results (based on 20,000 shots) are obtained by IQM Garnet machine developed by IQM, a Finnish-German quantum computer manufacturer.

## Appendix E. How discrete FT works

In this Appendix, let us try to clarify the meaning of the discrete Fourier transform with complex output. First of all, let us introduce the Euler's formula which establishes the fundamental relationship between the trigonometric functions and the complex exponential function, namely

$$e^{i\varphi} = \cos \varphi + i \sin \varphi, \quad (\text{E.1})$$

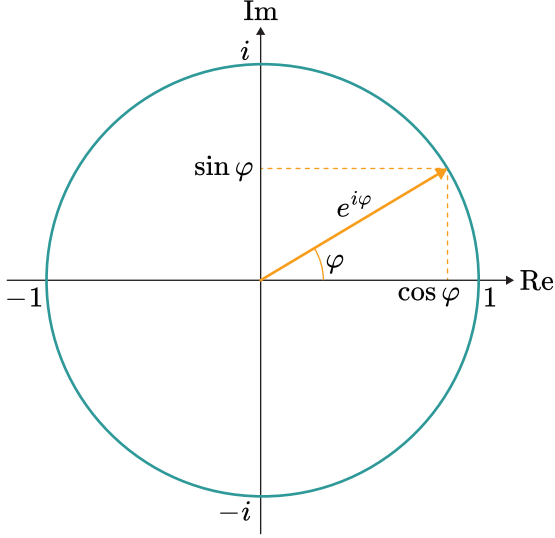
where  $i = \sqrt{-1}$  is the imaginary unit and  $\varphi$  is a generic angle between the line connecting the unitary complex number and the real axis on the complex plane in Fig. E.15(a). Next, we need to understand the meaning of  $\omega_N = e^{i2\pi/N}$  given by Eq. (13). Dividing the full angle  $2\pi$  into  $N$  parts,  $\omega_N$  is obtained by performing, in the complex plane, a rotation corresponding to a slice of  $2\pi/N$ . In polar form,  $\omega_N$  identifies one of the possible complex roots of unity.

The  $l$ -th power of  $\omega_N$ , i.e.  $\omega_N^l$ , corresponds to a rotation equal to  $l$  slices, i.e.  $2\pi l/N$ . Also the power  $\omega_N^l$  is a complex root of unity. This property allows to define a discrete geometrical series, namely  $1, \omega_N^1, \omega_N^2, \dots, \omega_N^{N-1}$ , which is represented as a series of vectors in Fig. E.15(b). The linear combination of these vectors implies

$$\sum_{l=0}^{N-1} \omega_N^l = 0. \quad (\text{E.2})$$

The intuitive meaning is that the net effect of a sum over all discrete angular vectors (with norm one), which are angularly equally spaced is equal to zero, because they are obtained by adding progressively the slice  $\omega_N$  until covering the full angle  $2\pi$ . It is possible to generalize

a)  $e^{i\varphi} = \cos \varphi + i \sin \varphi$



b)  $e^{i 2\pi l/8}$  for  $l = 0, 1, \dots, 7$

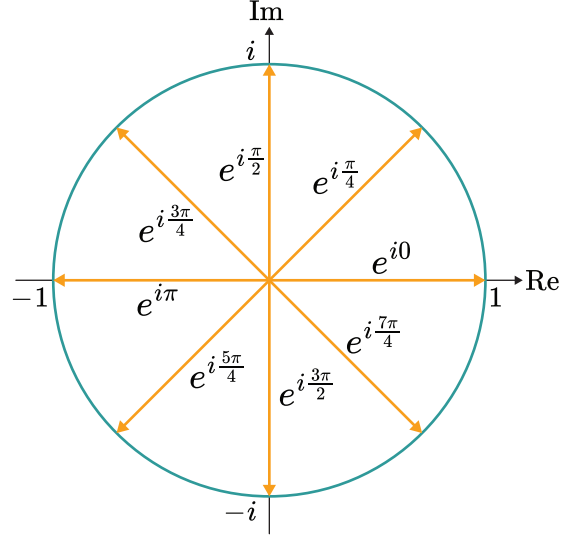


Figure E.15: Euler's formula for representing complex roots of unity in the polar form on complex plane.

the previous result by a generic parameter  $m$  as

$$\sum_{l=0}^{N-1} \omega_N^{lm} = \sum_{l=0}^{N-1} e^{2\pi i l m/N} = \begin{cases} N, & m = 0 \\ 0, & m \neq 0 \end{cases}. \quad (\text{E.3})$$

In order to understand the role of the parameter  $m$  and hence the meaning of the previous result, let us consider the examples reported in Fig. E.16. In top of those cases where the net effect is zero, one has also to add the trivial cases where the exponent is equal to zero and therefore all terms in the summation become equal to unity. Some powers of  $\omega_N$  appear also in the definition of operator  $\hat{U}_{FT}$  given by Eq. (16). The sum over all the components of the generic  $m$ -th row of the operator  $\hat{U}_{FT}$  is given by Eq. (E.3) multiplied by the normalization factor  $1/\sqrt{N}$ . Eq. (E.3) is also an immediate consequence of Vieta's formulas.

The property given by Eq. (E.3) is particularly useful in the discrete FT to select the harmonic components of a generic field (in our case, a temperature field  $\vec{T}$ ). Let us suppose that the temperature field  $\vec{T}$  is defined by a proper orthonormal basis  $\vec{e}_0, \vec{e}_1, \vec{e}_2, \dots, \vec{e}_{N-1}$ , as reported in Eq. (10), where  $T_l$  is the nodal value for the  $l$ -th mesh node. Let us suppose to decompose each nodal value  $T_l$  in harmonic components  $\tilde{T}_k$  where  $k$  goes from 0 to  $N-1$ , as prescribed by the inverse transform given by Eq. (15). These components altogether makes the transformed field, which is a column vector of complex numbers  $\vec{\tilde{T}}$  defined with regards to the same orthonormal basis by Eq. (11). As it will be clear by the following example, Eq. (E.3) allows one to project the vector  $\vec{\tilde{T}}$  on those components of the transformed vector with wavenumbers  $k$  such that  $m = f(k) = 0$ . As an example, let us consider a normalized (dimensionless) temperature profile, where the  $l$ -th generic nodal value is given

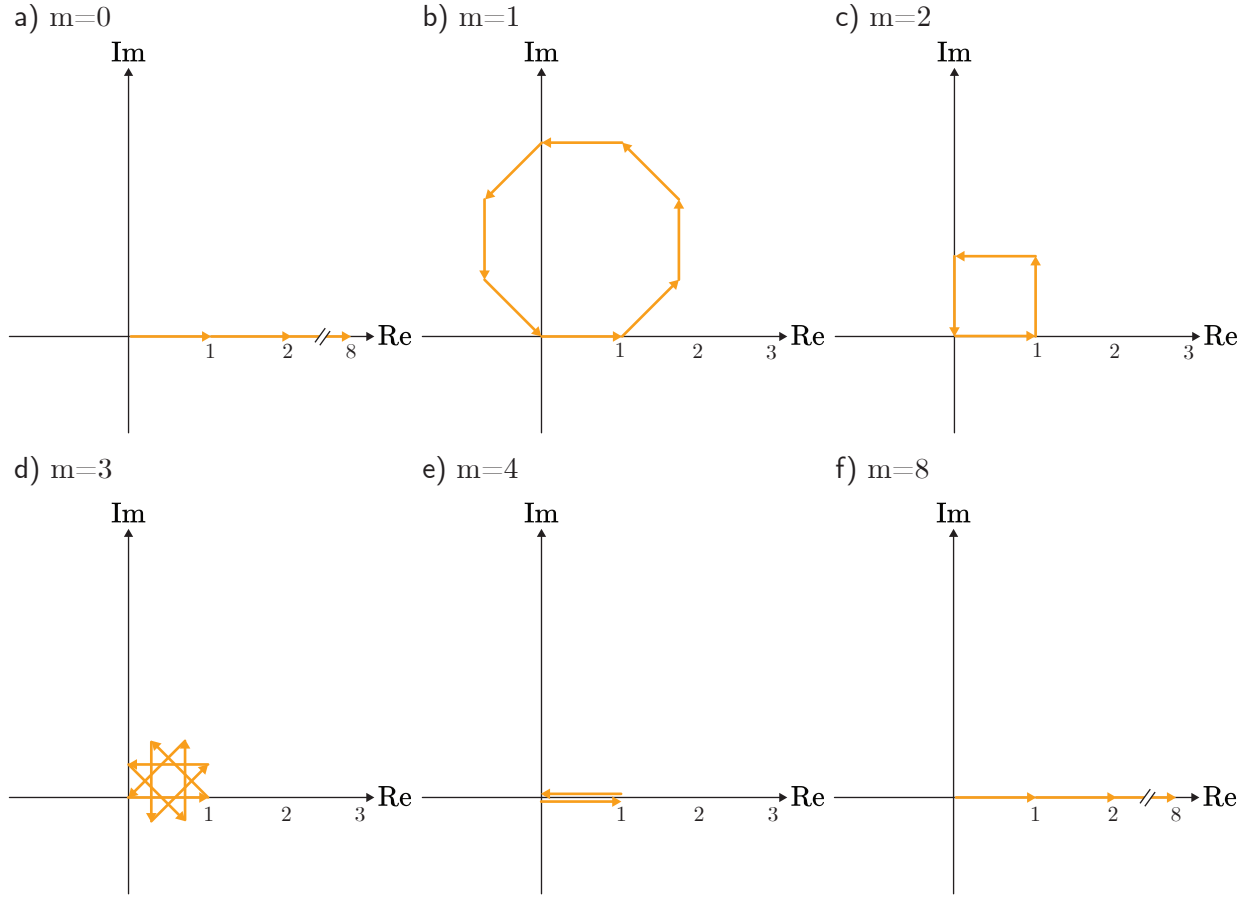


Figure E.16: Summation of geometric series of complex roots of unity, i.e.  $\sum_{l=0}^{N-1} e^{i 2\pi l m / N}$ . In case of  $N = 8$ , the sub-figures report two examples where the summation is non-zero, i.e.  $m = 0$  and  $m = 8$ , as well as three examples where the summation is zero, i.e.  $m = 1$ ,  $m = 2$  and  $m = 3$ .

by the following expression

$$T_l = 1 + \frac{1}{2} \sin \left[ \frac{2\pi}{N} (l + 1) \right]. \quad (\text{E.4})$$

Again from the Euler's formula, namely  $e^{i\theta} = \cos \theta + i \sin(\theta)$ , it is possible to express the sine function in the previous example as  $\sin(\theta) = (e^{i\theta} - e^{-i\theta}) / (2i)$ , which yields equivalently

$$T_l = 1 + \frac{1}{4i} \left( -\omega_N^{-1-l} + \omega_N^{1+l} \right). \quad (\text{E.5})$$

Let us apply the discrete FT given by Eq. (12), namely

$$\tilde{T}_k = \frac{1}{\sqrt{N}} \sum_{l=0}^{N-1} \left[ \omega_N^{kl} - \frac{\omega_N^{-1}}{4i} \omega_N^{l(k-1)} + \frac{\omega_N}{4i} \omega_N^{l(k+1)} \right]. \quad (\text{E.6})$$

Because the previous result is invariant under cyclic shifts, as evident from Fig. E.15(b),

substituting the equivalence  $\omega_N^{k(l+1)} = \omega_N^{k(l+1-N)}$  in the previous expression yields

$$\tilde{T}_k = \frac{1}{\sqrt{N}} \sum_{l=0}^{N-1} \left[ \omega_N^{kl} - \frac{\omega_N^{-1}}{4i} \omega_N^{l(k-1)} + \frac{\omega_N}{4i} \omega_N^{l(k+1-N)} \right]. \quad (\text{E.7})$$

Using the property for geometric series given by Eq. (E.3) and combining the results by the orthonormal basis used in Eq. (11) yields

$$\vec{T} = \sqrt{N} \left( \vec{e}_0 - \frac{\omega_N^{-1}}{4i} \vec{e}_1 + \frac{\omega_N}{4i} \vec{e}_{N-1} \right). \quad (\text{E.8})$$

Substituting the definition of  $\omega_N$  yields

$$\vec{T} = \sqrt{N} \vec{e}_0 + \left( \frac{\sqrt{2N}}{8} + i \frac{\sqrt{2N}}{8} \right) \vec{e}_1 + \left( \frac{\sqrt{2N}}{8} - i \frac{\sqrt{2N}}{8} \right) \vec{e}_{N-1}. \quad (\text{E.9})$$

From the application point of view, let us introduce the shift operator  $S$ , which is the circulant matrix defined as

$$\hat{S} : \vec{e}_l \rightarrow \vec{e}_{l+N/2 \bmod N}, \quad (\text{E.10})$$

where mod is the modulo operation, which returns the remainder of a division. By means of the shifted operator, it is possible to define the result in the standard (shifted) form, namely

$$\vec{T}_s = \hat{S} \vec{T} = \left( \frac{\sqrt{2N}}{8} - i \frac{\sqrt{2N}}{8} \right) \vec{e}_{N/2-1} + \sqrt{N} \vec{e}_{N/2} + \left( \frac{\sqrt{2N}}{8} + i \frac{\sqrt{2N}}{8} \right) \vec{e}_{N/2+1}. \quad (\text{E.11})$$

It is also useful to compute the wavenumber spectrum, which describes how the variance of the temperature field is distributed over different harmonic components, by means of Eq. (14). The (shifted) wavenumber spectrum  $\vec{p}_s^c$  is defined as

$$\vec{p}_s^c = \frac{1}{N} \vec{T}_s \odot \vec{T}_s^* = \frac{1}{16} \vec{e}_{N/2-1} + \vec{e}_{N/2} + \frac{1}{16} \vec{e}_{N/2+1}. \quad (\text{E.12})$$

where  $\odot$  represents the Hadamard (element-wise) product and the superscript  $*$  means the complex conjugate.

It may be interesting to compute the (shifted) wavenumber spectrum  $\vec{p}_s^c$  by the quantum spectrum  $\vec{p}_s$ . Taking into account Eq. (49) and Eq. (67) yields  $|b\rangle = (1/\theta) \vec{T}$ , which means that state  $|b\rangle$  is obtained by normalizing  $\vec{T}$  by the scaling factor  $\theta$ . Because the FT is a linear transformation, then  $|\tilde{b}\rangle = (1/\theta) \vec{\tilde{T}}$  and consequently  $\vec{\tilde{T}} = \theta |\tilde{b}\rangle$ . Using the latter relation into Eq. (E.12) yields

$$\vec{p}_s^c = \frac{1}{N} \vec{T}_s \odot \vec{T}_s^* = \frac{\theta^2}{N} \vec{\tilde{b}}_s \odot \vec{\tilde{b}}_s^* = \frac{\theta^2}{N} \vec{p}_s, \quad (\text{E.13})$$

where shifted  $\vec{\tilde{b}}_s = \hat{S} \vec{\tilde{b}}$ , shifted  $\vec{p}_s = \hat{S} \vec{p}$  and  $\vec{p}$  is the quantum spectrum given by Eq. (88). For our example temperature profile, given by Eq. (E.4),  $\theta^2/N = 9/8$  holds, which makes

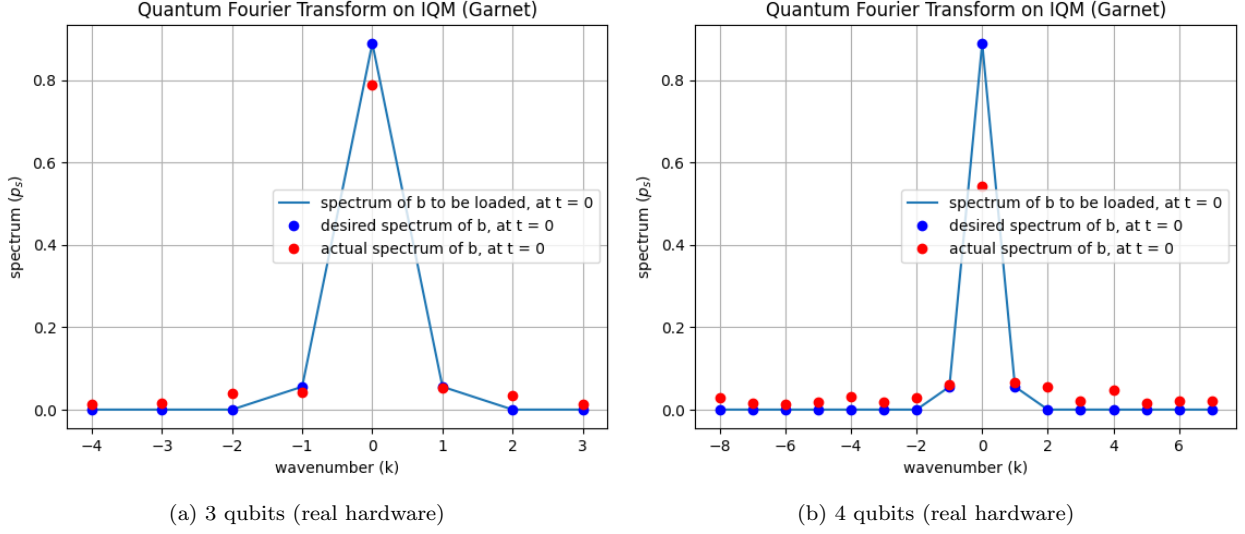


Figure E.17: Wavenumber spectrum, given by Eq. (88), computed by QFT routine provided in Qiskit [4]. These results (based on 20,000 shots) are obtained by IQM Garnet machine developed by IQM, a Finnish-German quantum computer manufacturer.

possible to compute  $\vec{p}_s^c$  by  $\vec{p}_s$ . In particular, for the example considered in this Appendix,  $\vec{p}_s$  is computed in case of  $N = 8$  (3 qubits) and  $N = 16$  (4 qubits) on a real hardware. In particular, let us use the IQM Garnet machine developed by IQM, a Finnish-German quantum computer manufacturer. The experimental results (based on 20,000 shots) of the QFT routine provided in Qiskit by IBM [4] on IQM Garnet are reported in Fig. E.17. These results should be considered indicative, as real hardware in the NISQ era is influenced by environmental conditions, causing actual outcomes to vary slightly from run to run.

## Appendix F. Example codes for VQE

In this Appendix, we provide some example codes for implementing the Variational Quantum Eigensolver (VQE) algorithm. Let us start with the Qiskit language [4] by IBM as a popular open-source software development kit. In Qiskit, the VQE can be implemented as follows.

```
import numpy as np
from qiskit.circuit.library import EfficientSU2
from qiskit.quantum_info import SparsePauliOp
from qiskit.primitives import StatevectorEstimator as Estimator
from scipy.optimize import minimize

num_qubits = 3 # number of qubits
N = pow(2,num_qubits) # number of mesh nodes
for i in range(N):
    T_old[i] = 1 + (1/2)*np.sin(2*np.pi*(i+1)/N)
```



```

TT_old = np.sum(T_old**2)
b0 = np.sqrt(TT_old)
b = T_old/b0 # initial profile

# (1) ANSATZ
raw_ansatz = EfficientSU2(num_qubits)
# Initial (arbitrary) set of parameter
theta0 = np.ones(raw_ansatz.num_parameters)

# (2) OBSERVABLE = HAMILTONIAN = "ENERGY"
# Conduction matrix
r = 0.5 # = delta_t*alpha/(delta_x**2) = Fo, Fourier number
d = np.ones(N)*(1+2*r)
od = np.ones(N-1)*(-r)
C = np.diag(d, 0) + np.diag(od, -1) + np.diag(od, 1)
C[0,N-1] = -r
C[N-1,0] = -r
O = np.identity(N)-np.outer(b,b)
O = np.matmul(O,C)
C_dag = np.transpose(C)
O = np.matmul(C_dag,O)
observable = SparsePauliOp.from_operator(O)

# (3) ESTIMATOR, quantum simulator
estimator = Estimator()

# LOSS FUNCTION
def cost_func_vqe(params, ansatz, hamiltonian, estimator):
    """Return estimate of energy from estimator

    Parameters:
        params (ndarray): Array of ansatz parameters
        ansatz (QuantumCircuit): Parameterized ansatz circuit
        hamiltonian (SparsePauliOp): Operator representation of
            Hamiltonian
        estimator (Estimator): Estimator primitive instance

    Returns:
        float: Energy estimate
    """
    pub = (ansatz, hamiltonian, params)
    cost = estimator.run([pub]).result()[0].data.evs

    return cost

```

```

# MINIMIZATION STEP
result = minimize(cost_func_vqe, theta0, args=(raw_ansatz.decompose
    ()), observable, estimator),
                method="COBYLA", # minimization method
                tol = 1e-3, # which affects iterations/time
                options={'maxiter': 1000, 'disp': True})

```

While the physics side of quantum computing makes significant progress, the support for high-level quantum programming abstractions is still in its infancy compared to modern classical languages and frameworks [11]. An interesting example is provided by Qrisp, which is a high-level programming language developed by Fraunhofer for creating and compiling quantum algorithms [11]. Its structured programming model enables scalable development and maintenance [11]. In Qrisp, the VQE can be implemented as follows.

```

import numpy as np
from qrisp import *
from qrisp.operators import QubitOperator
from qrisp.vqe.vqe_problem import *

num_qubits = 3 # number of qubits
N = pow(2,num_qubits) # number of mesh nodes

# (1) ANSATZ
def ansatz(qv,theta):
    for i in range(num_qubits):
        ry(theta[i],qv[i])
    for i in range(num_qubits-1):
        cx(qv[i],qv[i+1])
    cx(qv[num_qubits-1],qv[0])

# (2) OBSERVABLE = HAMILTONIAN = "ENERGY"
H = QubitOperator.from_matrix(0).to_pauli()

# (3) ESTIMATOR, quantum simulator
# Default, if 'backend' is not specified

# VQE PROBLEM
vqe = VQEProblem(hamiltonian = H,
                 ansatz_function = ansatz,
                 num_params = 3,
                 callback = True)

# MINIMIZATION STEP
qarg = QuantumVariable(num_qubits)
energy = vqe.run(qarg,

```

```
depth = 4,  
max_iter = 1000,  
mes_kwargs={'precision':0.1,'diagonalisation_method':  
            'commuting'})
```

## References

- [1] Yong-Xian Wang, Li-Lun Zhang, Wei Liu, Xing-Hua Cheng, Yu Zhuang, and Anthony T. Chronopoulos. Performance optimizations for scalable cfd applications on hybrid cpu+mic heterogeneous computing system with millions of cores. *Computers & Fluids*, 173:226–236, 2018. ISSN 0045-7930. doi: <https://doi.org/10.1016/j.compfluid.2018.03.005>. URL <https://www.sciencedirect.com/science/article/pii/S0045793018301038>.
- [2] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, USA, 10th edition, 2011. ISBN 1107002176.
- [3] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.
- [4] Ali Javadi-Abhari, Matthew Treinish, Kevin Krsulich, Christopher J. Wood, Jake Lishman, Julien Gacon, Simon Martiel, Paul D. Nation, Lev S. Bishop, Andrew W. Cross, Blake R. Johnson, and Jay M. Gambetta. Quantum computing with qiskit, 2024. URL <https://arxiv.org/abs/2405.08810>.
- [5] Xiaosi Xu, Jinzhao Sun, Suguru Endo, Ying Li, Simon C. Benjamin, and Xiao Yuan. Variational algorithms for linear algebra. *Science Bulletin*, 66(21):2181–2188, 2021. ISSN 2095-9273. doi: <https://doi.org/10.1016/j.scib.2021.06.023>. URL <https://www.sciencedirect.com/science/article/pii/S2095927321004631>.
- [6] N. M. Guseynov, A. A. Zhukov, W. V. Pogosov, and A. V. Lebedev. Depth analysis of variational quantum algorithms for the heat equation. *Phys. Rev. A*, 107:052422, May 2023. doi: 10.1103/PhysRevA.107.052422. URL <https://link.aps.org/doi/10.1103/PhysRevA.107.052422>.
- [7] Julia Ingelmann, Sachin S. Bharadwaj, Philipp Pfeffer, Katepalli R. Sreenivasan, and Jörg Schumacher. Two quantum algorithms for solving the one-dimensional advection–diffusion equation. *Computers & Fluids*, 281:106369, 2024. ISSN 0045-7930. doi: <https://doi.org/10.1016/j.compfluid.2024.106369>. URL <https://www.sciencedirect.com/science/article/pii/S0045793024002019>.
- [8] Hsin-Yuan Huang, Kishor Bharti, and Patrick Rebentrost. Near-term quantum algorithms for linear systems of equations with regression loss functions. *New Journal of Physics*, 23(11):113021, nov 2021. doi: 10.1088/1367-2630/ac325f. URL <https://dx.doi.org/10.1088/1367-2630/ac325f>.
- [9] Lukas Hantzko, Lennart Binkowski, and Sabhyata Gupta. Tensorized pauli decomposition algorithm. *Physica Scripta*, 99(8):085128, jul 2024. doi: 10.1088/1402-4896/ad6499. URL <https://dx.doi.org/10.1088/1402-4896/ad6499>.
- [10] Israel F. Araujo, Daniel K. Park, Francesco Petruccione, and Adenilton J. da Silva. A divide-and-conquer algorithm for quantum state preparation. *Scientific Reports*, 11(1), 2021. doi: 10.1038/s41598-021-85474-1. URL <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85102793192&doi=10.1038%2fs41598-021-85474-1&partnerID=40&md5=a5ea36704c5a5ebd2aa1af98374df7d9>.
- [11] Raphael Seidel, Sebastian Bock, Nikolay Vassilev Tcholtchev, and Manfred Hauswirth. Qrisp: a framework for compilable high-level programming of gate-based quantum computers, 2022. URL <https://publica.fraunhofer.de/handle/publica/445649>.
- [12] Zoltán Zimborás, Bálint Koczor, Zoë Holmes, Elsi-Mari Borrelli, András Gilyén, Hsin-Yuan Huang, Zhenyu Cai, Antonio Acín, Leandro Aolita, Leonardo Banchi, Fernando G. S. L. Brandão, Daniel Cavalcanti, Toby Cubitt, Sergey N. Filippov, Guillermo García-Pérez, John Goold, Orsolya Kálmán, Elica Kyoseva, Matteo A. C. Rossi, Boris Sokolov, Ivano Tavernelli, and Sabrina Maniscalco. Myths around quantum computation before full fault tolerance: What no-go theorems rule out and what they don’t, 2025. URL <https://arxiv.org/abs/2501.05694>.
- [13] Raban Iten, Roger Colbeck, Ivan Kukuljan, Jonathan Home, and Matthias Christandl. Quantum

circuits for isometries. *Phys. Rev. A*, 93:032318, Mar 2016. doi: 10.1103/PhysRevA.93.032318. URL <https://link.aps.org/doi/10.1103/PhysRevA.93.032318>.