

Hybrid classical-quantum approach to solve the heat equation using quantum annealersGiovani G. Pollachini,¹ Juan P. L. C. Salazar²,³ Caio B. D. Góes³,³ Thiago O. Maciel,³ and Eduardo I. Duzzioni^{1,3,*}¹*QuanBy Computação Quântica, Florianópolis, Santa Catarina, 88.030-902, Brazil*²*Engenharia Aeroespacial, Universidade Federal de Santa Catarina, Joinville, Santa Catarina, 89.219-600, Brazil*³*Departamento de Física, Universidade Federal de Santa Catarina, Florianópolis, Santa Catarina, 88.040-900, Brazil*

(Received 14 June 2021; accepted 13 September 2021; published 29 September 2021)

The numerical solution of partial differential equations by discretization techniques is ubiquitous in computational physics. In this work we investigate this approach in the quantum realm by solving the heat equation for a square plate subject to fixed temperatures at the edges and random heat sources and sinks within the domain. The hybrid classical-quantum approach consists of the solution on a quantum computer of the coupled linear system of equations that results from the discretization step. Owing to the limitations on the number of qubits and their connectivity, we use the Gauss-Seidel method to divide the full system of linear equations into subsystems, which are solved iteratively in block fashion. Each of the linear subsystems was solved using 2000Q and Advantage quantum computers developed by D-Wave Systems Inc. By comparing classical numerical and quantum solutions, we observe that the errors and chain break fraction are, on average, greater on the 2000Q system. Unlike in the classical Gauss-Seidel method, the errors of the quantum solutions level off after a few iterations of our algorithm. This is partly a result of the span of the real-number line available from the mapping of the chosen size of the set of qubit states. We verified this by using techniques to progressively shrink the range mapped by the set of qubit states at each iteration (increasing floating-point accuracy). As a result, no leveling off is observed. However, an increase in qubits does not translate to an overall lower error. This is believed to be indicative of the increasing length of chains required for the mapping to real numbers and the ensuing limitations of hardware.

DOI: [10.1103/PhysRevA.104.032426](https://doi.org/10.1103/PhysRevA.104.032426)**I. INTRODUCTION**

In computational physics, the mathematical description of the problem of interest relies on the solution of partial differential equations (PDEs) restricted to boundary conditions and initial conditions. Most of the techniques are based on discretization methods, in which the physical domain is divided into cells or volumes. When implicit methods are chosen, the discrete set of coupled equations to be solved is cast as a system of linear equations of the form $A\mathbf{x} = \mathbf{b}$, where A is generally an $N \times N$ sparse matrix and the vector \mathbf{b} is known, while the vector \mathbf{x} is the sought solution to the problem. The largest fraction of computational time is spent on the solution of the linear systems.

After the advent of the Harrow-Hassidim-Lloyd (HHL) algorithm [1] to solve the linear quantum problem, subsequent works implemented it as subroutine to solve ordinary or partial differential equations [2–9]. However, the solutions for the differential equations were presented in the form of a vector state, restricted to the calculation of global properties of the system of interest; otherwise, the cost to obtain each amplitude of the quantum state would render the quantum approach too expensive. Clader *et al.* [3] argue that an exponential speedup can be achieved by using the HHL algorithm to solve the electromagnetic scattering cross-section problem

via the finite-element method. However, Montanaro and Pallister [6] showed that the relation between the size of the system of linear equations and the accuracy of the solution was missed, which implies that the speedup can be reduced or even eliminated. A thorough analysis of the resources needed to solve the electromagnetic scattering cross section of a two-dimensional (2D) target was made by Scherer *et al.* [10], who concluded that the number of resources must be reduced by many orders of magnitude for the algorithm to become practical. There are other issues concerning quantum state preparation and the necessity of an oracle to encode the coefficients of the matrix A . For reasons like these, it is not yet clear whether exponential speedups have been achieved by quantum algorithms using the linear quantum problem as a subroutine. Recently, Linden *et al.* [11] showed that HHL-like algorithms used to solve the heat equation are never faster than the best classical algorithms. However, a quantum alternative which uses amplitude amplification can accelerate the process, achieving at most a quadratic speedup. Although this a very encouraging result, the gate model of quantum computing requires advances to approach large problems. Following this reasoning, quantum annealers can be more attractive at the current stage of quantum computing due to their large number of qubits and increased connectivity. For instance, a graph-coloring-based methodology to solve differential equations through quantum annealers was proposed in [12].

*duzzioni@gmail.com

An alternative method for solving the linear system of equations by quantum annealing was presented by O'Malley and Vesselinov [13] and Rogers and Singleton [14]. An analysis of the resources employed for this task was made by Borle and Lomonaco [15], who showed that in some circumstances it is possible to gain an advantage over traditional algorithms for the least-squares problem. The linear system is encoded in a quadratic unconstrained binary optimization (QUBO) problem [16] through a well-defined protocol, whose ground state is the solution to the problem. As opposed to a HHL-type algorithm, the final solution of the linear system of equations is a classical state codifying the vector \mathbf{x} .

In this work we solve the steady-state 2D heat equation with a hybrid classical-quantum algorithm. Applying the finite-difference method to the partial differential equation yields a system of coupled linear equations. In order to enable the solution of larger linear systems of equations, we use the block Gauss-Seidel method and solve the full system iteratively, where the smaller subsystems of equations are solved on the quantum processing units (QPUs). The quantum algorithm is implemented in the 2000Q and Advantage systems of D-Wave Systems Inc.

This paper is divided as follows. In Sec. II we introduce the D-Wave quantum annealing methodology, review the approach of Rogers and Singleton [14] to solve the linear system of equations, present the block Gauss-Seidel method, and elaborate on quantum resources required to implement the algorithm. The results are presented in Sec. III and are followed by conclusions and perspectives in Sec. IV.

II. METHODS

As described above, the basic idea of our hybrid approach is to replace a classical subroutine to solve the system of linear equations that results from the discretization of partial differential equations by a quantum algorithm. In the proposed method the system of linear equations is solved using quantum annealing, in which the final Hamiltonian encodes a QUBO problem [14]. In the following we briefly review the quantum annealing followed by the QUBO problem.

A. D-Wave quantum annealing

D-Wave quantum annealing is based on the adiabatic quantum evolution of the transverse Ising Hamiltonian with the system in thermal equilibrium at a very low temperature [17]. The goal of the computation is to minimize an objective function (energy) for which the solution to the desired problem is approximately encoded in its ground state [18]. In order to achieve the ground state of the system with a high success rate, the quantum fluctuations must be dominant over thermal fluctuations [17].

The idea behind this model is the preparation of the system as the ground state of an easy-to-prepare Hamiltonian, for instance ($\hbar\omega = 1$),

$$H(0) = - \sum_{i=1}^n \sigma_x^i, \quad (1)$$

where σ_x^i is the Pauli matrix in the x direction for the i th particle and its ground state is $|\psi(0)\rangle = |+\rangle^{\otimes n}$, which con-

tains the equally distributed superposition of all states in the computational basis. By choosing the final Hamiltonian $H(1)$ as the operator whose ground state contains the solution to the desired problem, it is possible to interpolate between these two Hamiltonians with a schedule function $f(s)$, subject to $f(0) = 0$ and $f(1) = 1$, resulting in

$$H(s) = [1 - f(s)]H(0) + f(s)H(1), \quad (2)$$

where $s = t/T$ ($0 \leq s \leq 1$) is the dimensionless time, t is the current time, and T is the total time of the computation. The final Hamiltonian implemented by D-Wave is the Ising Hamiltonian,

$$H(1) = H_{\text{Ising}} = \sum_i \alpha_i \sigma_z^i + \sum_{i,j} \beta_{ij} \sigma_z^i \sigma_z^j, \quad (3)$$

with the parameters α_i and β_{ij} being chosen to encode the solution to the problem and σ_z^i being the Pauli matrix in the z direction for particle i .

According to the adiabatic theorem [19], if the evolution is performed adiabatically, then the ground state of the problem Hamiltonian $H(1)$ is found with high probability, provided that the evolution time T from the initial Hamiltonian $H(0)$ to $H(1)$ is proportional to g_{min}^{-2} , where $g_{\text{min}} = E_1 - E_0$ is the minimum energy gap between the ground and first excited states [19,20]. It is important to observe that a suitable choice for the schedule function may result in completely different evolution times [20–22]. Therefore, the final Hamiltonian $H(1)$ must contain the solution to the linear system of equations [14]. Next, we show how to build $H(1)$, thus casting the system of linear equations into a QUBO problem.

B. QUBO algorithm to solve linear systems of equations

O'Malley and Vesselinov [13] and Rogers and Singleton [14] showed how to map the problem of solving a linear system of equations into a QUBO problem [16]. Also, in Chang *et al.* [23] the linear system of equations was obtained as a particular case of a nonlinear system of equations. Given a linear system $\mathbf{Ax} = \mathbf{b}$, the solution \mathbf{x} can be found by minimizing the function

$$H(\mathbf{x}) = (\mathbf{Ax} - \mathbf{b})^T (\mathbf{Ax} - \mathbf{b}). \quad (4)$$

We can approximate each component x_i ($i = 1, \dots, N$) of the vector \mathbf{x} by its binary representation in a given interval $[-d_i, 2c_i - d_i]$, with R bits, as

$$x_i = c_i \sum_{r=0}^{R-1} q_r^i 2^{-r} - d_i, \quad (5)$$

where each $q_r^i \in \{0, 1\}$ is the r th bit in the binary representation of x_i . If $d_i > 0$ and $c_i > d_i/2$, the domain of x_i contains positive and negative real values. This can easily be generalized to complex variables [14]. After substituting (5) into (4) and dropping the positive constant term, the equation becomes

$$H(q) = \sum_{r=0}^{R-1} \sum_{i=0}^{N-1} h_r^i q_r^i + \sum_{r,s=0}^{R-1} \sum_{i,j=0}^{N-1} J_{rs}^{ij} q_r^i q_s^j, \quad (6)$$

where the coefficients are given by

$$h_r^i = -2 \left(\sum_{j,k=0}^{N-1} A_{ki} A_{kj} c_i d_j + \sum_{j=0}^{N-1} A_{ji} c_i b_j \right) 2^{-r}, \quad (7)$$

$$J_{rs}^{ij} = \left(\sum_{k=0}^{N-1} A_{ki} A_{kj} c_i c_j \right) 2^{-(r+s)}. \quad (8)$$

Here, A_{ij} are coefficients of the matrix A , and b_i are components of the vector \mathbf{b} .

In quantum annealers, the physical qubits are indexed by a one-dimensional linear index ℓ [see Eq. (3)], while the logical qubits are indexed by two-dimensional indices i and r , where $i = 0, 1, \dots, N-1$ and $r = 0, 1, \dots, R-1$, as shown in Eq. (6). The mapping between the logical and physical indices is given by

$$\ell(i, r) = iR + r, \quad \ell = 0, \dots, NR - 1, \quad (9)$$

such that the inverse mapping is

$$\begin{aligned} i_\ell &= \lfloor \ell/R \rfloor, \\ r_\ell &= \ell \pmod{R}. \end{aligned} \quad (10)$$

The mapping from $q_\ell \rightarrow \sigma_\ell$ is $\sigma_\ell = 2q_\ell - 1$, with $\sigma_\ell = \{+1, -1\}$.

Finally, the quantization of the QUBO problem is made through $\hat{q}_\ell |q\rangle = q_\ell |q\rangle$, where the operators are such that $\hat{q}_\ell^2 = \hat{q}_\ell$, with eigenvalues $q_\ell \in \{0, 1\}$ and $|q\rangle = |q_0\rangle \otimes \dots \otimes |q_{NR-1}\rangle$. In order to solve the system of linear equations, the coefficients $h_r^i \rightarrow h_\ell$ and $J_{rs}^{ij} \rightarrow J_{\ell\kappa}$, described in Eqs. (7) and (8), must be provided to the D-Wave OCEAN software [24], the interface that communicates with the quantum hardware.

The readout of the quantum algorithm is a bit string $q_0 q_1 \dots q_{NR-1}$ of NR qubits that returns to the classical algorithm for further processing. In principle, the qubits must be fully connected. However, in situations where the matrix A is sparse, several coefficients J_{rs}^{ij} are set to zero; hence, not all connections between the qubits are necessary. With knowledge of the above, if the full linear system of equations is solved directly, the number of required qubits NR for equivalent floating-point accuracy of classical algorithms makes this implementation prohibitive at the current stage of technological development. However, the linear system can be divided into subsystems, and the full linear system can be solved iteratively in a hybrid quantum-classical approach, as detailed next.

C. Iterative QUBO method for solving linear systems

The block Gauss-Seidel method consists of solving the linear system $\mathbf{Ax} = \mathbf{b}$ iteratively by applying the Gauss-Seidel method to the original system divided into blocks [25]. Suppose, for simplicity, that the size of the system is an even number N and we divide it into two subsystems,

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}. \quad (11)$$

This translates to

$$A_{11}\mathbf{x}_1 + A_{12}\mathbf{x}_2 = \mathbf{b}_1, \quad (12)$$

$$A_{21}\mathbf{x}_1 + A_{22}\mathbf{x}_2 = \mathbf{b}_2. \quad (13)$$

In the iterative method we make an initial guess for \mathbf{x}_2 , such as $\mathbf{x}_2^{(0)} = 0$, where the upper index of the block vector $\mathbf{x}_2^{(0)}$ defines the step of the iteration process. Rearrangement of Eqs. (12) and (13) allow us to set up an iterative process where the initial guess is updated by solving two linear systems of size $N/2$:

$$A_{11}\mathbf{x}_1^{(1)} = \mathbf{b}_1 - A_{12}\mathbf{x}_2^{(0)}, \quad (14)$$

$$A_{22}\mathbf{x}_2^{(1)} = \mathbf{b}_2 - A_{21}\mathbf{x}_1^{(1)}. \quad (15)$$

At the k th step,

$$A_{11}\mathbf{x}_1^{(k)} = \mathbf{b}_1 - A_{12}\mathbf{x}_2^{(k-1)}, \quad (16)$$

$$A_{22}\mathbf{x}_2^{(k)} = \mathbf{b}_2 - A_{21}\mathbf{x}_1^{(k)}. \quad (17)$$

The iteration above can be written in the form

$$M\mathbf{x}^{(k)} = \mathbf{b} - U\mathbf{x}^{(k-1)}, \quad (18)$$

with

$$\begin{aligned} M &= \begin{bmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{bmatrix}, \quad U = \begin{bmatrix} 0 & A_{12} \\ 0 & 0 \end{bmatrix}, \\ \mathbf{b} &= \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}. \end{aligned}$$

By subtracting $M\mathbf{x} = \mathbf{b} - U\mathbf{x}$ from Eq. (18),

$$M\mathbf{e}^{(k)} = -U\mathbf{e}^{(k-1)}, \quad (19)$$

$$\mathbf{e}^{(k)} = (M^{-1}U)^{(k)}\mathbf{e}^{(0)}, \quad (20)$$

where $\mathbf{e}^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}$ is the error at the k th iteration. Hence, a sufficient condition for convergence is that the spectral radius of the iteration matrix $M^{-1}U$ is less than unity, i.e., $\rho(M^{-1}U) < 1$.

The generalization of the method is possible by dividing the original linear system into several blocks. If the system is divided into D blocks, each smaller system has size $N/D \times N/D$. The number of required iterations is not known beforehand but is defined by a convergence criterion. A typical criterion depends on the normalized residual $\hat{r}^{(k)}$, defined by

$$\hat{r}^{(k)} \equiv \frac{\|\mathbf{Ax}^{(k)} - \mathbf{b}\|}{\|\mathbf{b}\|}, \quad (21)$$

where $\mathbf{x}^{(k)}$ is the solution of the system of linear equations after k iteration steps. A solution is considered to be converged when $\hat{r}^{(k)}$ meets a specified tolerance τ , such that $\hat{r}^{(k)} \leq \tau$. Here, we instead monitor the relative error of the solution vector at the k th iteration $\mathbf{x}^{(k)}$, given as

$$\hat{e}^{(k)} \equiv \frac{\|\mathbf{x}^{(k)} - \mathbf{x}\|}{\|\mathbf{x}\|}, \quad (22)$$

where the reference solution \mathbf{x} is obtained by direct matrix inversion. It can be shown that $\hat{e}^{(k)}$ and $\hat{r}^{(k)}$ are related according

to

$$\hat{e}^{(k)} \leq \kappa(A) \hat{r}^{(k)}, \quad (23)$$

where $\kappa(A)$ is the condition number of the coefficient matrix A . Preconditioning techniques should be used when A is ill conditioned. In this work we do not pursue this issue further since it is relevant for classical and quantum algorithms alike.

Scaling of the method

The representation of a vector \mathbf{x} with N components requires NR qubits, where R is the number of qubits that gives the desired accuracy of floating-point numerical representation ϵ of each variable x_i . However, as stated above, it is possible to solve smaller subsystems of size N/D by partitioning the full system into D parts. In this case the number of qubits scales as $(N/D)R$. In order to reduce the number of qubits required in the computation for a given floating-point accuracy of x_i , one possibility is to shrink the interval $[-d_i, 2c_i - d_i]$ at each iteration, which we adopt here. Any reduction in the number of required qubits is significant in the era of noisy intermediate-scale quantum computers, where at most a few thousand qubits are available. Even more dramatic is the number of connections necessary to solve the QUBO problems. In the worst case, every logical qubit must be fully connected [see the coefficient J_{rs}^{ij} in Eq. (7)]. The block Gauss-Seidel method can reduce the number of connections among qubits from N^2R^2 to N^2R^2/D^2 . Additionally, when A is sparse, connectivity is reduced considerably.

If ϵ is a bound for the accuracy of the floating-point numerical representation of a physical variable x_i , for a solution represented with R qubits, then

$$\frac{2c_i}{2^R} \leq \epsilon$$

since $2c_i$ is the length of the open interval $[-d_i, 2c_i - d_i]$ where the solution is represented and 2^R is the number of different points that can be represented with R bits inside the aforementioned interval. From the expression above, the number of qubits required to represent one variable with floating-point accuracy ϵ scales as

$$R \geq \log_2(2c_i/\epsilon).$$

Since the total computing time \mathcal{T} depends on the energy gap between the ground and first excited states of the system, it is hard to obtain precise estimates of \mathcal{T} . As noted by Amin and Choi [26], the task of analytically extracting the minimum gap scaling has been extremely difficult in practice, except for a few special cases, such as for Deutsch-Jozsa [27] and Grover [21,28] algorithms.

In Table I we summarize the consuming quantum resources used in the iterative and noniterative QUBO algorithms to solve the linear system of equations.

III. RESULTS AND DISCUSSION: TWO-DIMENSIONAL HEAT EQUATION

In order to assess the viability and performance of the proposed method, we solve the heat equation for a square plate of length L subject to fixed temperature at its edges and multiple heat sources and sinks within the domain. The steady-state

TABLE I. Comparison between original QUBO and iterative QUBO methods. $\mathcal{T}_{N,R}$ indicates the time to run one QUBO algorithm for a linear system of size N and R qubits for representing the solution, while $\mathcal{T}_{N/D,R}$ is the time to solve a system of size N/D . n_{iter} is the number of iterations in the case of the iterative QUBO method. Both times $\mathcal{T}_{N,R}$ and $\mathcal{T}_{N/D,R}$ depend on the annealing time and the number of runs to sample solutions of the problem.

Method	No. qubits	Time
Original QUBO	NR	$\mathcal{T}_{N,R}$
Iterative QUBO	$(N/D)R$	$\mathcal{T}_{N/D,R} n_{\text{iter}}$

solution $T(x, y)$ is then determined by the following PDE and boundary conditions,

$$k \left[\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right] + \dot{q} = 0, \quad (24)$$

$$T(x, 0) = 0^\circ\text{C}, \quad x \in [0, L],$$

$$T(x, L) = \frac{x}{L} \times 100^\circ\text{C}, \quad x \in [0, L],$$

$$T(0, y) = 0^\circ\text{C}, \quad y \in [0, L],$$

$$T(L, y) = \frac{y}{L} \times 100^\circ\text{C}, \quad y \in [0, L], \quad (25)$$

where $\dot{q}(x, y)$ is the heat-source function and k is the thermal conductivity of the material (we use $k = 1$ W/m K).

The problem is solved numerically by the finite-difference method. The domain is divided into m segments of equal length, with nodes located at positions $x_i = (L/m)i$ and $y_j = (L/m)j$, where $i, j = 0, \dots, m$. The points (x_i, y_j) of the grid corresponding to $i = 0, i = m, j = 0$, or $j = m$ form the boundary of the plate and are determined by the boundary conditions (25). The points inside the grid are the unknowns, and the partial derivatives can be written as

$$\begin{aligned} \frac{\partial^2 T}{\partial x^2}(x_i, y_j) &= \frac{T(x_{i+1}, y_j) - 2T(x_i, y_j) + T(x_{i-1}, y_j)}{(L/m)^2} \\ &\quad + \frac{(L/m)^2}{12} \frac{\partial^4 T}{\partial x^4}(x_i^*, y_j), \\ \frac{\partial^2 T}{\partial y^2}(x_i, y_j) &= \frac{T(x_i, y_{j+1}) - 2T(x_i, y_j) + T(x_i, y_{j-1})}{(L/m)^2} \\ &\quad + \frac{(L/m)^2}{12} \frac{\partial^4 T}{\partial y^4}(x_i, y_j^*) \end{aligned} \quad (26)$$

for some $x_i^* \in [x_{i-1}, x_{i+1}]$ and $y_j^* \in [y_{j-1}, y_{j+1}]$. Ignoring terms proportional to $(L/m)^2$ or higher orders, the PDE is approximated by

$$\begin{aligned} k[T(x_{i+1}, y_j) - 4T(x_i, y_j) + T(x_{i-1}, y_j) + T(x_i, y_{j+1}) \\ + T(x_i, y_{j-1})] + Q(x_i, y_j) = 0, \end{aligned} \quad (27)$$

where $i, j = 1, 2, \dots, m-1$ and $Q = (L/m)^2 \dot{q}$. Equation (27) can be written in the form of a linear system $\mathbf{Ax} = \mathbf{b}$, where A is a sparse matrix with few nonzero entries and \mathbf{b} is defined by the boundary conditions and eventual source terms.

The spatial discretization forms a 9×9 matrix ($m = 11$), resulting in a system of coupled linear equations with 81 unknown variables, which is solved by the iterative QUBO

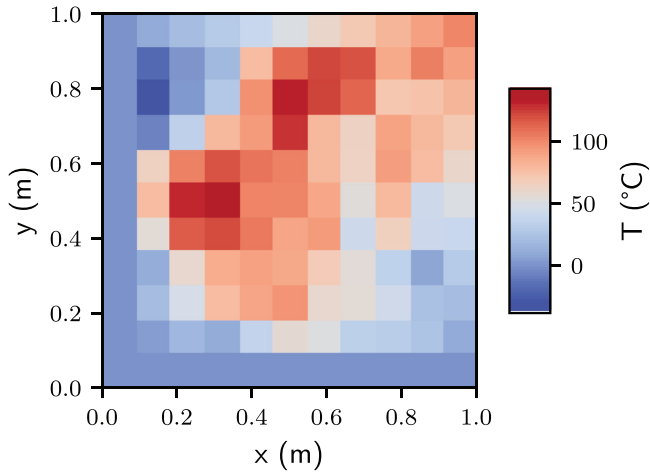


FIG. 1. Temperature distribution in the square plate.

algorithm on the Advantage (Advantage_system1.1) and D-Wave 2000Q (DW_2000Q_6) systems [29]. Each result comes from the lowest-energy state obtained after a user-defined number of 1000 repetitions (samples) of the annealing process on the QPUs. The parameters h_r^i (bias) and J_{rs}^{ij} (coupling) were computed in order to fit in the range of the physical system according to Eqs. (7) and (8). The minor embedding, the process through which the logical qubits given in Eq. (6) are mapped to physical qubits on quantum hardware, was obtained automatically through the EMBEDDINGCOMPOSITE function available from the D-Wave OCEAN software package. A sample solution of the problem is shown in Fig. 1, where a random distribution of heat sources and sinks was implemented according to Fig. 2. The PYTHON code used to generate all results presented here is publicly available [30].

This approach enables us to solve larger linear systems of equations, circumventing the limited number of qubits in the D-Wave quantum processing units. In this work we use at most 7 precision qubits and blocks of size 9×9 , which demand at most 63 logical qubits to describe all 9 unknown variables in the block Gauss-Seidel approach, while 567 would be required to solve the full system of linear equations

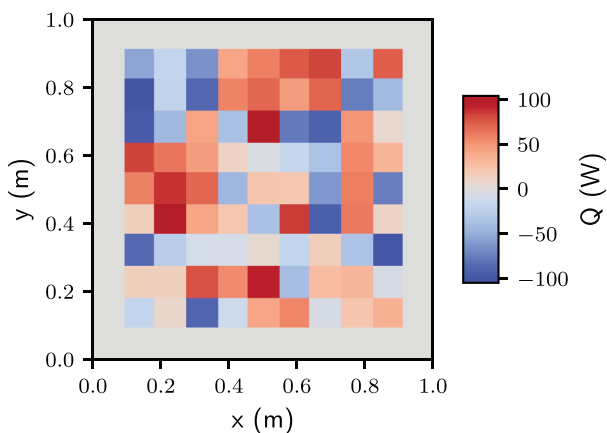


FIG. 2. Distribution of random heat sources and sinks.

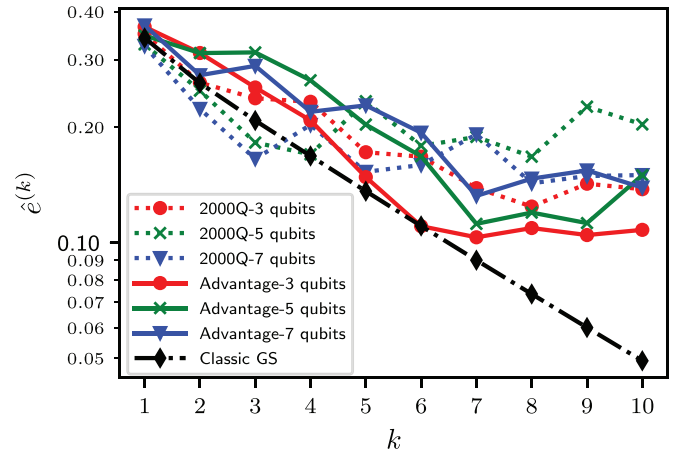


FIG. 3. The relative error $\hat{\epsilon}^{(k)}$ as a function of the number of iterations k of the block Gauss-Seidel method for different numbers of qubits used in the numerical representation of the solution \mathbf{x} . The solutions obtained by the 2000Q and Advantage systems are compared with the classic Gauss-Seidel implementation.

with 7 precision qubits. This choice of block size and number of qubits was a result of preliminary tests for which a possible minor embedding on the D-Wave quantum computers was found. We will see that the number of logical qubits is much smaller than the number of physical qubits in both D-Wave systems because of the limited connectivity between physical qubits.

In Fig. 3 the dependence of the relative error defined in Eq. (22) is shown as a function of the number of iterations of the block Gauss-Seidel method for different numbers of qubits used in the numerical representation of the unknown variables. In this example, the smallest error is found using three qubits in the Advantage system. We also notice that the error does not continually decrease, as would be expected in a classical algorithm. This occurs in part because of the limited floating-point accuracy of the variables x_i , such that after a certain number of iterations, an improvement of the solution is not possible. Another observation is that the increase in qubits does not necessarily translate into smaller errors. This result was not expected. However, it can be related to the appearance of more noise in the system since there are more qubits involved in the numerical representation of a given variable x_i , which in turn increases the probability of chain breaks. This subject will be discussed in the following. In general, the Advantage system performs marginally better than 2000Q.

In order to create a logical qubit, generally more than one physical qubit is necessary. This is a consequence of the limited number of couplers (connections) between qubits. In the D-Wave quantum processing units, a set of physical qubits that describe a vertex of a graph forms a chain. For this chain to behave as a basic quantum processing unit, all of its physical qubits must be in the same state. When this does not occur, the chain is broken, and the state is determined by a majority vote. In Fig. 4 we plot the average chain break fraction as a function of the number of iterations and the number of qubits used to represent each value of temperature. The average chain break fraction for each number of precision

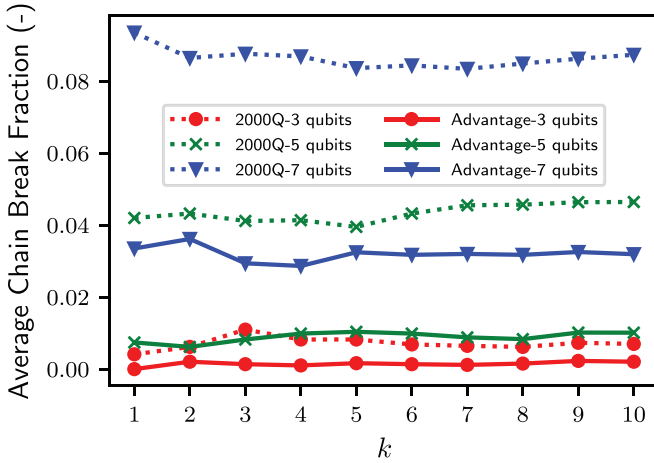


FIG. 4. Average chain break fraction as a function of the number of iterations k in the block Gauss-Seidel method for different numbers of precision qubits.

qubits is greater in the 2000Q than in the Advantage system and is, on average, independent of the iteration number. This result is expected since in the Chimera topology of the 2000Q system each qubit is connected to 6 other qubits, while in the Pegasus topology of the Advantage system, each qubit connects to 15 other qubits. Furthermore, the Advantage system has 5000+ qubits compared to 2000+ qubits in the 2000Q system. These features allow for more compact chains that are less susceptible to noise [31].

A significant improvement in the relative error of Fig. 3 can be obtained by reducing the length of the interval $2c_i$ represented by a given number of qubits at each iteration. The iterative QUBO algorithm is modified by shrinking the interval by factor of $\gamma = 0.8$ at each iteration; that is, the interval for estimating the solution at the k th iteration is given by $[x_i^{k-1} - c_i\gamma^{k-1}, x_i^{k-1} + c_i\gamma^{k-1}]$, where x_i^{k-1} is the estimate for x_i at the $(k-1)$ th iteration. This procedure increases the capacity of the QUBO solver to represent the solution of the system with a fixed number R of qubits, as can be seen in Fig. 5. The implementation of this updated search interval length brings the quantum solvers on par with the error obtained on classical hardware. It must be noted, however, that the update of the right-hand sides of Eqs. (16) and (17) during the Gauss-Seidel iterations is performed on a classical computer, with double precision, even in the quantum algorithm.

A comparison of Figs. 3 and 5 shows that the 2000Q performs slightly better than the Advantage system when the update of the search interval length is incorporated into the algorithm. There is not a clear interpretation why this occurs since the Advantage system is an improved version of the 2000Q system, where the chains are more compact and less susceptible to noise [31]. However, Zaborniak and de Sousa [32] suggest that the DW_2000Q_6 QPU presents a lower level of Hamiltonian noise compared to Advantage_system1.1. This noise originates mainly from the flux noise of the bias field. The results obtained from Figs. 3 and 5 also suggest that a greater number of precision qubits does not necessarily translate to a smaller error, as noted earlier.

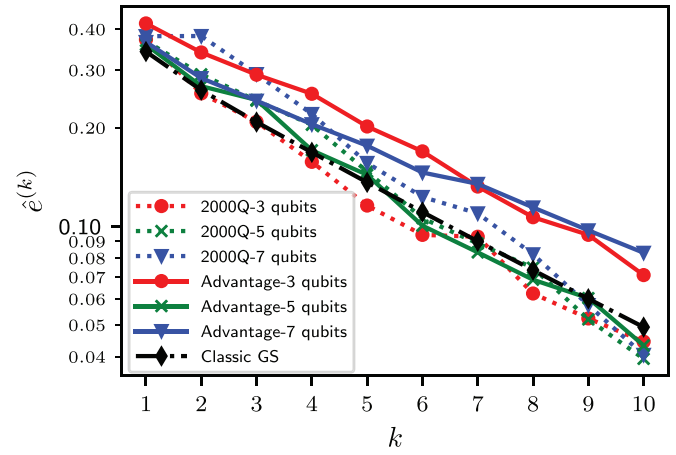


FIG. 5. The relative error $\hat{\epsilon}^{(k)}$ as a function of the number of iterations k of the block Gauss-Seidel method, with a shrink factor of 0.8 of the search domain at each iteration, for different numbers of qubits used in the numerical representation of the solution \mathbf{x} . The solutions obtained by the 2000Q and Advantage systems are compared with the classic Gauss-Seidel implementation.

IV. CONCLUSIONS AND PERSPECTIVES

The hybrid classical-quantum algorithm we have developed solves partial differential equations by the finite-difference method, yielding a system of coupled linear equations. This technique was tested with success for the 2D steady-state heat equation but, in principle, can be applied to any partial differential equation, encompassing a large array of applications for solution by quantum annealers. The method with updated search interval length allows us to obtain accuracy comparable to a classical algorithm with double precision using three qubits to numerically represent each variable. The main limitation of the quantum hardware is the reduced connectivity between qubits, although great strides have been made to increase the number of couplers for each qubit. The iterative algorithm allows for a significant reduction in the number of logical qubits, from NR to NR/D , at the expense of an increase in the number of required iterations for a given accuracy.

It is our understanding that there are good prospects for the iterative QUBO algorithm and that there is room for further improvements, such as robust heuristics to update the search interval length. Another improvement can be obtained by using an application-oriented hardware design, as many linear systems typical of finite-difference and finite-volume methods are characterized by sparseness of the coefficient matrix A . This could have the advantage of requiring fewer couplers than the present platforms. Owing to our limited access to the D-Wave QPUs, we believe that more tests are required to fully benchmark the performance of 2000Q and Advantage systems for the solution of systems of linear equations.

ACKNOWLEDGMENTS

The authors acknowledge financial support from the Brazilian funding agencies Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Conselho Nacional de Desenvolvimento Científico e Tecnológico

(CNPq), Fundação de Amparo à Pesquisa e Inovação do Estado de Santa Catarina (FAPESC), and the Instituto Na-

cional de Ciência e Tecnologia de Informação Quântica (INCT-IQ).

-
- [1] A. W. Harrow, A. Hassidim, and S. Lloyd, *Phys. Rev. Lett.* **103**, 150502 (2009).
- [2] S. K. Leyton and T. J. Osborne, [arXiv:0812.4423](https://arxiv.org/abs/0812.4423).
- [3] B. D. Clader, B. C. Jacobs, and C. R. Sprouse, *Phys. Rev. Lett.* **110**, 250504 (2013).
- [4] Y. Cao, A. Papageorgiou, I. Petras, J. Traub, and S. Kais, *New J. Phys.* **15**, 013021 (2013).
- [5] D. W. Berry, *J. Phys. A* **47**, 105301 (2014).
- [6] A. Montanaro and S. Pallister, *Phys. Rev. A* **93**, 032324 (2016).
- [7] D. W. Berry, A. M. Childs, A. Ostrander, and G. Wang, *Commun. Math. Phys.* **356**, 1057 (2017).
- [8] P. C. S. Costa, S. Jordan, and A. Ostrander, *Phys. Rev. A* **99**, 012323 (2019).
- [9] S. Wang, Z. Wang, W. Li, L. Fan, Z. Wei, and Y. Gu, *Quantum Inf. Process.* **19**, 170 (2020).
- [10] A. Scherer, B. Valiron, S.-C. Mau, S. Alexander, E. Van den Berg, and T. E. Chapuran, *Quantum Inf. Process.* **16**, 60 (2017).
- [11] N. Linden, A. Montanaro, and C. Shao, [arXiv:2004.06516](https://arxiv.org/abs/2004.06516).
- [12] S. Srivastava and V. Sundararaghavan, *Phys. Rev. A* **99**, 052355 (2019).
- [13] D. O'Malley and V. V. Vesselinov, in *2016 IEEE High Performance Extreme Computing Conference (HPEC)* (IEEE, Waltham, MA, 2016), pp. 1–7.
- [14] M. L. Rogers and R. L. Singleton, Jr., *Front. Phys.* **8**, 265 (2020).
- [15] A. Borle and S. J. Lomonaco, in *WALCOM: Algorithms and Computation, WALCOM 2019*, Lecture Notes in Computer Science, Vol. 11355, edited by G. Das, P. Mandal, K. Mukhopadhyaya, and S. Nakano (Springer, Cham, 2019).
- [16] F. Glover, G. Kochenberger, and Y. Du, *4OR-Q J Oper Res* **17**, 335 (2019).
- [17] T. Kadowaki and H. Nishimori, *Phys. Rev. E* **58**, 5355 (1998).
- [18] B. Apolloni, C. Carvalho, and D. De Falco, *Stochastic Processes Their Appl.* **33**, 233 (1989).
- [19] S. Jansen, M.-B. Ruskai, and R. Seiler, *J. Math. Phys.* **48**, 102111 (2007).
- [20] T. Albash and D. A. Lidar, *Rev. Mod. Phys.* **90**, 015002 (2018).
- [21] J. Roland and N. J. Cerf, *Phys. Rev. A* **65**, 042308 (2002).
- [22] D. An and L. Lin, [arXiv:1909.05500](https://arxiv.org/abs/1909.05500).
- [23] C. C. Chang, A. Gambhir, T. S. Humble, and S. Sota, *Sci. Rep.* **9**, 10258 (2019).
- [24] D-Wave Systems Inc., <https://docs.ocean.dwavesys.com/en/stable/index.html>.
- [25] G. H. Golub and C. F. V. Loan, *Matrix Computations*, 3rd ed., Johns Hopkins Studies in the Mathematical Sciences (Johns Hopkins University Press, Baltimore, MD, 1996).
- [26] M. H. S. Amin and V. Choi, *Phys. Rev. A* **80**, 062326 (2009).
- [27] S. Das, R. Kobes, and G. Kunstatter, *Phys. Rev. A* **65**, 062310 (2002).
- [28] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, [arXiv:quant-ph/0001106](https://arxiv.org/abs/quant-ph/0001106).
- [29] D-Wave Systems Inc., D-Wave User Manual 09-1109B-B, 2021 (unpublished), https://docs.dwavesys.com/docs/latest/doc_qpu.html.
- [30] <https://github.com/croquemadame/quantumGS.git>.
- [31] D-Wave Systems Inc., Technical Report No. 14-1049a-a, 2020 (unpublished), <https://dwavesys.com>.
- [32] T. Zaborniak and R. de Sousa, *IEEE Trans. Quantum Eng.* **2**, 1 (2021).