Box algorithm for the solution of differential equations on a quantum annealer

Siddhartha Srivastava* and Veera Sundararaghavan†
Department of Aerospace Engineering, University of Michigan, Ann Arbor, Michigan 48109, USA

(Received 7 January 2019; published 31 May 2019)

Differential equations are ubiquitous in models of physical phenomena. Applications like steady-state analysis of heat flow and deflection in elastic bars often admit to a second-order differential equation. In this paper, we discuss the use of a quantum annealer to solve such differential equations by recasting a finite element model in the form of an Ising Hamiltonian. The discrete variables involved in the Ising model introduce complications when defining differential quantities, for instance, gradients involved in scientific computations of solid and fluid mechanics. To address this issue, a graph-coloring-based methodology is proposed which searches iteratively for solutions in a subspace of weak solutions defined over a graph, hereafter called the "box algorithm." The box algorithm is demonstrated by solving a truss mechanics problem on a D-Wave quantum computer.

DOI: 10.1103/PhysRevA.99.052355

I. INTRODUCTION

Computational methods are rapidly emerging as an essential tool to understand and solve complex engineering problems, complementing the traditional means of experimentation and theory. Feynman's statement [1]—"with a suitable class of quantum machines you could imitate any quantum system, including the physical world"—has driven our vision towards a machine that can solve computational problems inaccessible to classical computers. Early versions of such quantum computers have already appeared. Mirroring gate-based classical computers, gate-based quantum computers with a small number of qubits have been demonstrated and promise an eventual path towards universal quantum computation. However, noise limits the number of gate operations that can be enforced before the quantum states decohere. In parallel, quantum annealers have been developed that provide a significant number of qubits for solving a class of combinatorial optimization problems. In these machines, an Ising Hamiltonian is engineered such that the solution to the computational problem is encoded in its ground state. The system evolves adiabatically to the ground state as governed by the Schrödinger equation for the time-dependent Hamiltonian.

The D-Wave system is a quantum annealer that currently provides more than 2000 qubits modeling a transverse Ising Hamiltonian whose ground state is NP-complete. The Hamiltonian with q_i describing the state of the ith qubit is given by

$$E(\mathbf{q}) = \sum_{i\in\text{sites}} H_i q_i + \sum_{(i,j)\in\text{links}} J_{ij} q_i q_j. \tag{1}$$

The Hamiltonian includes self-interaction (H_i is the on-site energy of qubit q_i) and site-site interaction terms (J_{ij} are the interaction energies of two connected qubits q_i and q_j), with the qubits connected in a chimera graph. The system is first initialized in the ground state of a Hamiltonian which is known and easy to prepare. Then, the Hamiltonian is changed such that the system equilibrates to the ground state of the Ising Hamiltonian $E(\mathbf{q})$ according to the adiabatic theorem. NP-hard combinatorial optimization problems can be encoded through the field and site interaction strengths. The system particularly holds promise for solving graph coloring problems with large sizes (N) where classical polynomial time algorithms cannot be devised. Many engineering problems in airline scheduling, image segmentation, and pattern recognition have been encoded as graph coloring problems solvable on quantum annealers.

While differential equations are ubiquitous in models of physical phenomena, the use of quantum annealers for scientific computing in solid and fluid mechanics has not yet been explored. Scientific computing mostly involves solving a linear system of equations $Ax = b$ defined on a continuum domain discretized with finite elements. The matrix A, generally is a sparse, structured, and positive-definite matrix obtained by assembling element-level stiffness matrices. In the past, gate-based quantum computing algorithms have been devised to solve the system of linear equations using quantum linear system algorithms (QLSAs) (Harrow-Hassidim-Lloyd algorithm [2]) and its variants [3–6]. This algorithm, unlike a classical solver, does not give a direct solution x but rather allows sampling from the solution vector. Nevertheless, this has spawned several works in differential equation modeling on quantum computers [7–15]. The sampling task by itself requires solving $Ax = b$. In the classical setting, the complexity scales with the size of the problem and goes as $O(Nsk\log(1/\epsilon))$ for the conjugate gradient method, where N is the number of unknowns, k is the condition number, s is the sparsity of A, and ϵ is the precision of the solution. On the other hand, the QLSA [2] has a favorable running time of $O(\log(N)k^2s^2/\epsilon)$ which scales logarithmically with the size of the problem. Quantum annealers are especially attractive for scientific computing with the ability to scale up the simulations to a more significant number of qubits. However, algorithms for the solution

*sidsriva@umich.edu
†veeras@umich.edu

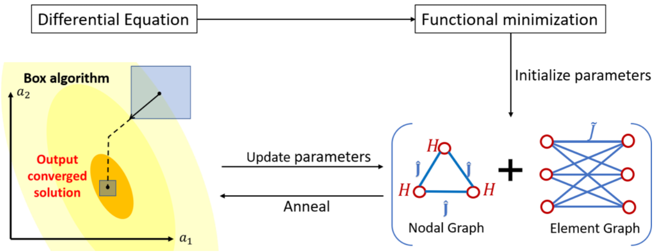


FIG. 1. Illustration of the procedure for solving a differential equation.

of differential equations have not been devised yet on these systems [16]. The merit of the paper is the algorithm to solve a differential equation on an annealer.

Here, we note that the solution to $Ax = b$ can be encoded in an equivalent minimization problem min $(\frac{1}{2}x^TAx - x^Tb)$, which contains field and interaction terms similar to an Ising model. Thus, in this paper, we explore mapping of this energy to an Ising Hamiltonian on the D-Wave machine recognizing that the graph in the D-Wave chip by itself models a finite element meshlike topology. The element level stiffness and force vectors are then encoded in the Ising Hamiltonian as interaction weights and field variables. Dirichlet boundary conditions are enforced by modifying field terms to favor one qubit state over another. An illustration of this procedure is presented in Fig. 1. A discretized version of the differential equation is solved using energy minimization on a graph. Direct minimization of energy may hold advantages over the conventional finite element approach in systems which lead to a matrix with large condition numbers, and zero or negative eigenvalues leading to bifurcation events such as buckling in shells and phase transitions [17].

In the solution to a differential equation, the qubits must encode a rational number. However, the qubit encoding the Ising lattice point carries two discrete levels (up and down spin) in the ground state. In classical computers, with similar binary (0 or 1) encoding, anywhere from 32 bits (float) to 80 bits (long double) of memory can be used to encode more than 12 million high-precision variables in 1 GB of memory. In contrast, currently available quantum annealers have a limited number of physical qubits. This restriction makes the representation of solutions of double precision similar to a classical computer extremely expensive. In Refs. [18,19], the problem of minimizing $||Ax - b||$ in the least-squares sense was posed by encoding physical qubits to represent rational numbers using a radix 2 representation. This format requires a significant number of physical qubits and connections to represent positive rational numbers and an additional qubit to represent the sign of the number [18]. In comparison, the box algorithm searches within a small discrete set of up and down qubit values with each element of the set mapped to a double precision value, thereby eliminating the need for additional qubits to achieve higher precision.

In this paper, we consider a self-adjoint form of a second-order differential equation as the model problem. The problem statement and the relevant mathematical details are presented in Sec. II. The graph representation of the problem is formulated in Sec. III. The iterative procedure, referred to as the "box algorithm," is presented in Sec. IV. All procedures are accompanied by numerical examples for elucidation. This algorithm is demonstrated by solving a truss mechanics problem on a D-Wave quantum computer in Sec. V.

II. MATHEMATICAL PRELIMINARIES

A self-adjoint form of a second-order differential equation on an interval (x_l, x_r) is defined as

$$-(p(x)u'(x))' + q(x)u(x) = f(x), \quad x_l < x < x_r. \tag{2}$$

Dirichlet boundary conditions are considered at both ends, i.e., $u(x_l) = u_l$ and $u(x_r) = u_r$. A well-posed problem requires $p(x) \geqslant p_{\min} > 0$ and $q(x) \geqslant q_{\min} \geqslant 0$. Furthermore, for convenience, it is assumed that $p, q \in C([x_l, x_r])$ and $f \in L^2([x_l, x_r])$. These conditions are sufficient to show the existence of a unique solution to the weak form [20].

A. Functional minimization

Motivated by the intractability of direct integration of differential equation (2), it is often convenient to employ functional minimization techniques. Calculus of variations can be used to observe that the minimization of functional (3) leads to the strong form described in Eq. (2):

$$\Pi[u] = \int_{x_l}^{x_r} \left(\frac{1}{2}pu'^2 + \frac{1}{2}qu^2 - fu\right)dx. \tag{3}$$

Square integrability of u and its first derivative are required in this definition of $\Pi[u]$. The implication is that the minimizing solution, u, lies in the Sobolov space $H^1([x_l, x_r])$. A discrete problem is obtained by using a finite basis for the solution defined in Eq. (4), which satisfies the Dirichlet boundary conditions:

$$u_N(x) = \sum_{i=0}^{N} a_i\phi_i(x) \tag{4}$$

The admissible choices of $\mathbf{a} = (a_0, a_1, \dots, a_N)$ satisfy $u_N(x_d) = u_d$, where x_d is a Dirichlet boundary and u_d is the prescribed value at that point. This approximation reduces the infinite-dimensional functional minimization problem to finite dimensions. The approximated functional Π_N is entirely determined by the representation of u in the finite basis as shown in Eq. (5). It is worth observing that the choice of $\phi_i(x)$ is such that $\phi_i \in H^1([x_l, x_r])$, i.e., for any $u_N \in V_N = \text{span}\{\phi_1, \phi_2, \dots, \phi_N\} \subseteq H^1([x_l, x_r])$. Additionally the proper inclusion, $V_i \subseteq V_{i+1}$, guarantees convergence of the solution with increasing N:

$$\Pi_N[a_0, \dots, a_r, \dots, a_N] = \int_{x_l}^{x_r} \frac{p}{2}\left(\sum_{i=0}^{N} a_i\phi_i'\right)^2 + \frac{q}{2}\left(\sum_{i=0}^{N} a_i\phi_i\right)^2 - f\left(\sum_{i=0}^{N} a_i\phi_i\right)dx. \tag{5}$$

As the solution is completely determined by the variable $\mathbf{a}$, the functional minimization of Eq. (5) is reformalized as Eq. (6), where $\mathbf{a}^{b.a.}$ refers to the coefficients of best approximation of solution, u_N, in the subspace V_N:

$$\mathbf{a}^{b.a.} = \arg\min_{\mathbf{a}} \Pi_N(\mathbf{a}). \tag{6}$$

B. Finite element approximation

The finite element basis provides a popular choice of compactly supported shape functions. For the purpose of simplicity, "tent" or "hat" functions [defined in Eqs. (7)] are used in this work. The domain is split into N elements with $N+1$ nodes. The generalization to higher order families of piecewise-continuous basis functions is immediate but is omitted for brevity.

$$\phi_i(x) = \begin{cases} 0, & x < x_{i-1} \\ (x - x_{i-1})/(x_i - x_{i-1}), & x_{i-1} \leqslant x < x_i \\ 1 - (x - x_i)/(x_{i+1} - x_i), & x_i \leqslant x < x_{i+1} \\ 0, & x \geqslant x_{i+1}\,. \end{cases} \tag{7}$$

The usage of a compact basis further reduces the complexity by reducing the integration over the whole domain to a summation of integration over smaller elements. It is shown in Sec. III that this choice of shape functions lead to a relatively sparse graph. It simplifies the computation by reducing the size of the graph optimization problem. The simplified form of Π specialized for the hat functions is presented in Eq. (8):

$$\begin{aligned}\Pi_N(\mathbf{a}) = \sum_{i=1}^{N} & a_{i-1}^2\left(\int_{x_{i-1}}^{x_i} \frac{p}{2}\phi_{i-1}'^2 + \frac{q}{2}\phi_{i-1}^2 dx\right) \\ & + a_i^2\left(\int_{x_{i-1}}^{x_i} \frac{p}{2}\phi_i'^2 + \frac{q}{2}\phi_i^2 dx\right) \\ & + a_{i-1}a_i\left(\int_{x_{i-1}}^{x_i} p\phi_{i-1}'\phi_i' + q\phi_{i-1}\phi_i dx\right) \\ & - a_{i-1}\left(\int_{x_{i-1}}^{x_i} f\phi_{i-1}dx\right) - a_i\left(\int_{x_{i-1}}^{x_i} f\phi_i dx\right).\end{aligned} \tag{8}$$

This form of Π promotes modularity in computation and allows expressing the functional as

$$\Pi_N = \sum_{i=1}^{N} \mathbf{A_i}.\mathbf{S_i} \tag{9}$$

where vectors $\mathbf{A_i} \equiv \mathbf{A_i}(a_{i-1}, a_i)$ and $\mathbf{S_i} \equiv \mathbf{S_i}(p, q, f)$ are defined for each element in Eqs. (10):

$$\begin{aligned}\mathbf{A_i} &= \left[a_{i-1}^2, \quad a_i^2, \quad a_{i-1}a_i, \quad a_{i-1}, \quad a_i\right]^T, \\ \mathbf{S_i} &= \left[\int_{x_{i-1}}^{x_i} \frac{p}{2}\phi_{i-1}'^2 + \frac{q}{2}\phi_{i-1}^2 dx, \quad \int_{x_{i-1}}^{x_i} \frac{p}{2}\phi_i'^2 + \frac{q}{2}\phi_i^2 dx,\right. \\ &\quad \times \int_{x_{i-1}}^{x_i} p\phi_{i-1}'\phi_i' + q\phi_{i-1}\phi_i dx, \quad -\int_{x_{i-1}}^{x_i} f\phi_{i-1}dx, \\ &\quad \left. -\int_{x_{i-1}}^{x_i} f\phi_i dx\right]^T.\end{aligned} \tag{10}$$

The vector $\mathbf{S_i}$ is independent of state $\mathbf{a}$ and is therefore only computed once in the whole procedure.

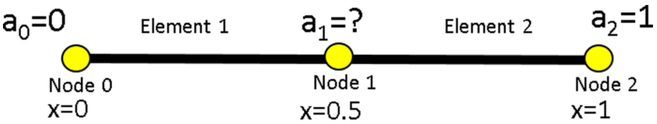
Example 1. Consider the differential equation with boundary conditions $u(0) = 0$ and $u(1) = 1$:

$$\frac{d^2u}{dx^2} = 0, \qquad 0 < x < 1.$$

The functional is

$$\Pi[u] = \frac{1}{2}\int_0^1 u'^2 dx.$$

For simplicity, consider a grid with a uniform mesh of two elements and three nodes:



Using linear interpolants for the elements,

$$u(x) = \begin{cases} a_0(1-2x) + a_1(2x), & 0 < x \leqslant 0.5 \\ a_1(2-2x) + a_2(2x-1), & 0.5 < x \leqslant 1. \end{cases}$$

The functional with the finite element (FE) discretization is

$$\Pi_N(\mathbf{a}) = (a_0 - a_1)^2 + (a_1 - a_2)^2.$$

Modular representation of the functional ($\Pi_N = \mathbf{A_1}.\mathbf{S_1} + \mathbf{A_2}.\mathbf{S_2}$) is

$$\begin{aligned}\mathbf{A_1} &= \left[a_0^2, a_1^2, a_0a_1, a_0, a_1\right]^T, \\ \mathbf{A_2} &= \left[a_1^2, a_2^2, a_1a_2, a_1, a_2\right]^T, \\ \mathbf{S_1} &= \mathbf{S_2} = [1, 1, -2, 0, 0]^T.\end{aligned}$$

Examples 2-4 will illustrate key steps of the algorithm using the problem defined in *Examples 1* ■

III. GRAPH COLORING PROBLEM

Quantum annealing methods are tailored to find the lowest energy states in an Ising system defined in Eq. (1). The Ising Hamiltonian defines a binary graph coloring problem with each vertex of graph or qubit labeled as +1 or −1. The value of the qubits determines the free variable, in this case **a**. The parameters H_i and J_{ij} are defined such that the Ising Hamiltonian, for a labeling representing the state **a**, corresponds to the functional $\Pi_N(\mathbf{a})$. These problems, namely, the representation of state and estimation of parameters, are addressed in this section.

A. Representation of state

Representation of a functional in terms of continuous variables is not feasible on quantum architectures. Due to this limitation, the values of each a_i (ith component of **a**) are chosen from a finite set of values based on the labeling of qubits. The representation presented here permits three possible values of a_i at each node. In particular, for each node (indexed "i"), the state (a_i) is exactly determined by the labeling of qubits q_1^i, q_2^i, and q_3^i with the ith node taking values in the set $\{v_{i_1}, v_{i_2}, v_{i_3}\}$. Equation (11) defines a mapping $(q_1^i, q_2^i, q_3^i) \rightarrow a_i$ as shown in Table I:

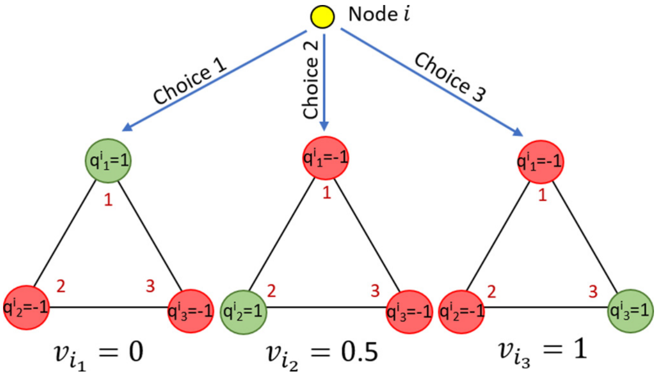
$$a_i = \sum_{j=1}^{3} v_{i_j}\frac{q_j^i + 1}{2}. \tag{11}$$

TABLE I. The mapping from qubits to state a_i at node.

(q_1^i, q_2^i, q_3^i)	a_i
$(1, 1, 1)$	$v_{i_1} + v_{i_2} + v_{i_3}$
$(1, 1, -1)$	$v_{i_1} + v_{i_2}$
$(1, -1, 1)$	$v_{i_1} + v_{i_3}$
$(1, -1, -1)$	v_{i_1}
$(-1, 1, 1)$	$v_{i_2} + v_{i_3}$
$(-1, 1, -1)$	v_{i_2}
$(-1, -1, 1)$	v_{i_3}
$(-1, -1, -1)$	0

It is observed that the mapping results in $a_i \in \{v_{i_1}, v_{i_2}, v_{i_3}\}$ when two qubits are labeled -1 and one qubit is labeled $+1$. Next it is shown that the Ising parameters can be manipulated to make these labelings energetically favorable, thereby eliminating the occurrence of undesirable labels.

Example 2. In general, the set $\{v_{i_1}, v_{i_2}, v_{i_3}\}$ is different for each node. However, for simplicity, consider the same set of admissible states for all three nodes given by $\{v_{i_1}, v_{i_2}, v_{i_3}\} \equiv \{0, 0.5, 1\}$. Each node is defined by three qubits as shown in the following diagram:



We know that the three qubits each defining the solution at the first and last nodes should take up choices 1 and 3, respectively, due to boundary conditions. The choice for the second node is to be solved. ■

B. Parameter estimation

To promote modularity, the graph representation is decomposed into two component subgraphs, namely, a nodal graph and an element graph (see Fig. 2). Each node and element of the FE discretization is endowed with a node graph and an element graph, respectively. This allows one to refine the mesh by extending the graph.

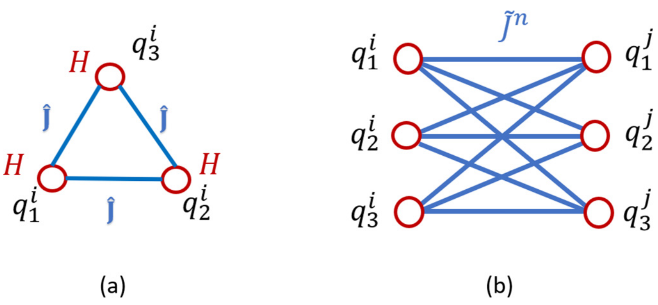


FIG. 2. Connectivity of (a) nodal graph and (b) element graph.

1. Nodal graph

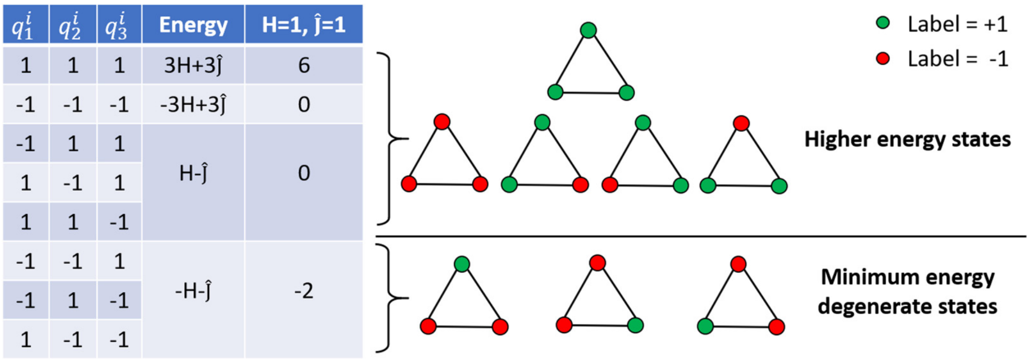
The nodal graph is given by a fully connected graph with three vertices representing the three qubits of the FE node. The nodal graph ensures that the energy-minimizing states of the Ising Hamiltonian correspond to state **a** with favorable choice of $a_i \in \{v_{i_1}, v_{i_2}, v_{i_3}\}$ with equal probability. As mentioned earlier, the set of favorable labeling of qubits at a node is given by $\{Q_1, Q_2, Q_3\} \equiv \{(1, -1, -1), (-1, 1, -1), (-1, -1, 1)\}$. Since each of the three labelings is equally likely in the absence of any functional minimization, it is expected that the same value of the coupling strength $(\hat{J})$ for each connection and the field strength (H) for each node is used. A choice of $\hat{J}$ and H that fulfills these conditions is presented in Fig. 3. Here, all the field and interaction terms for the nodal graph are given a value of 1. The application of the Dirichlet boundary condition is also done by augmenting the field strength of the nodal graph. For example, by switching the field term H corresponding to the second qubit q_2 of a boundary node b to -1 forces a lower value of the functional for the boundary node state of $(-1, +1, -1)$, which corresponds to the solution v_{b_2}. This allows us to encode the value at the boundary to be v_{b_2}.

2. Element graph

The element graph is used to make the energy of minimizing states of the graph correspond to the value of the functional Π_N of the continuous problem. Each element graph encodes the contribution of the respective element to the functional. Since the contribution of each element is dependent on the values at the nodes of the element, the element graph is constructed by connecting the vertices of neighboring nodes. In particular, the site-site interaction in the nth element graph can be estimated as a matrix, $\tilde{J}^n$, where $(\tilde{J}^n)_{kl}$ represents the coupling energy between qubits q_k^i (kth qubit of ith node) and q_l^j (lth qubit of jth node) with i, j being the nodes of the nth element. As shown in the previous section, the contribution of the nth element towards the functional, based on the choice of a compact basis function, is evaluated as $\mathbf{A_n}.\mathbf{S_n}$. The elements of the vector, $\mathbf{A_n} \equiv \mathbf{A_n}(a_i, a_j)$, can therefore take nine (3×3) possible values based on the values of (a_i, a_j). For a particular choice of labeling of qubit the Ising energy of the element graph is estimated as $E = \sum_{k=1}^{3} \sum_{l=1}^{3} (\tilde{J}^n)_{kl} q_k^i q_l^j$. When the labeling is chosen appropriately (each node has two "-1" labels and one "$+1$" label), this energy is equal to the value of the functional for the corresponding state, **a**, as shown in Eq. (12):

$$\sum_{k=1}^{3} \sum_{l=1}^{3} (\tilde{J}^n)_{kl} q_k^i q_l^j = \mathbf{A_n}(a_i, a_j).\mathbf{S_n}. \tag{12}$$

This relation can be used to estimate $\tilde{J}^n$ by solving the set of nine independent linear equations presented. It is an important observation that the independence of these sets of equations relies on the fact that for any node, $v_{i_k} \neq v_{i_l}$ for $k \neq l$. Additionally, the energy of the element graph breaks the symmetry between the states that minimize the energy of the

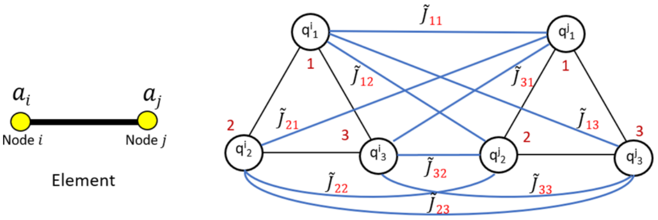


q_1^i	q_2^i	q_3^i	Energy	H=1, ĵ=1
1	1	1	3H+3ĵ	6
-1	-1	-1	-3H+3ĵ	0
-1	1	1	H-ĵ	0
1	-1	1		
1	1	-1		
-1	-1	1	-H-ĵ	-2
-1	1	-1		
1	-1	-1		

FIG. 3. Self-interaction and site-site interaction parameters for nodal subgraph.

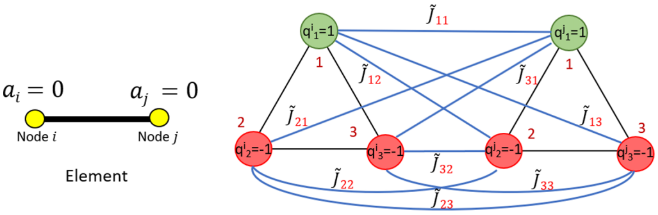
nodal graph; however, the values of $\widetilde{J}^n$ should be judiciously scaled (uniformly along all elements) such that the energy of unfavorable states remain high.

Example 3. A single element with two nodes admits to the following connectivity:



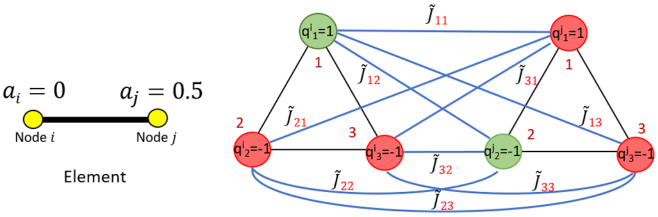
The estimated parameters reflect the contribution of the element to the functional for a given choice of labeling.

Sample 1 is



In the above diagram, both nodes take up choice 1 ($a_i = a_j = 0$). The interaction energy for qubits is $E = \widetilde{J}_{11} - \widetilde{J}_{12} - \widetilde{J}_{13} - \widetilde{J}_{21} + \widetilde{J}_{22} + \widetilde{J}_{23} - \widetilde{J}_{31} + \widetilde{J}_{32} + \widetilde{J}_{33} = (a_i - a_j)^2 = 0$.

Sample 2 is



In the above diagram, node i takes up choice 1 ($a_i = 0$), while node j takes up choice 2 ($a_j = 0.5$). The interaction energy for qubits is $E = -\widetilde{J}_{11} + \widetilde{J}_{12} + \widetilde{J}_{13} + \widetilde{J}_{21} - \widetilde{J}_{22} - \widetilde{J}_{23} - \widetilde{J}_{31} + \widetilde{J}_{32} + \widetilde{J}_{33} = (a_i - a_j)^2 = 0.25$.

Collectively solving the equation for all nine such possibilities [as shown in Eq. (12)] is as follows:

$$\begin{bmatrix} +1 & -1 & -1 & -1 & +1 & +1 & -1 & +1 & +1 \\ -1 & +1 & +1 & +1 & -1 & -1 & -1 & +1 & +1 \\ -1 & +1 & +1 & -1 & +1 & +1 & +1 & -1 & -1 \\ -1 & +1 & -1 & +1 & -1 & +1 & +1 & -1 & +1 \\ +1 & -1 & +1 & -1 & +1 & -1 & +1 & -1 & +1 \\ +1 & -1 & +1 & +1 & -1 & +1 & -1 & +1 & -1 \\ -1 & -1 & +1 & +1 & +1 & -1 & +1 & +1 & -1 \\ +1 & +1 & -1 & -1 & -1 & +1 & +1 & +1 & -1 \\ +1 & +1 & -1 & +1 & +1 & -1 & -1 & -1 & +1 \end{bmatrix} \begin{bmatrix} \widetilde{J}^n_{11} \\ \widetilde{J}^n_{12} \\ \widetilde{J}^n_{13} \\ \widetilde{J}^n_{21} \\ \widetilde{J}^n_{22} \\ \widetilde{J}^n_{23} \\ \widetilde{J}^n_{31} \\ \widetilde{J}^n_{32} \\ \widetilde{J}^n_{33} \end{bmatrix}$$

$$= \begin{bmatrix} (v_{i_1} - v_{j_1})^2 \\ (v_{i_2} - v_{j_1})^2 \\ (v_{i_3} - v_{j_1})^2 \\ (v_{i_1} - v_{j_2})^2 \\ (v_{i_2} - v_{j_2})^2 \\ (v_{i_3} - v_{j_2})^2 \\ (v_{i_1} - v_{j_3})^2 \\ (v_{i_2} - v_{j_3})^2 \\ (v_{i_3} - v_{j_3})^2 \end{bmatrix},$$

$$\widetilde{J}^1 = \widetilde{J}^2 = \begin{bmatrix} 0.1250 & 0.3750 & 0.3750 \\ 0.3750 & 0.5000 & 0.3750 \\ 0.3750 & 0.3750 & 0.1250 \end{bmatrix}.$$

The above parameters will exactly reproduce the functional in the interaction term. The boundary conditions are enforced by setting the self-interaction term for qubits q^0_1, q^2_3 to $H = -1$. This locks the state at the first boundary node as $a_0 = v_{0_1} = 0$ and at the second boundary node as $a_2 = v_{2_3} = 1$. Energy minimization of the resulting Ising Hamiltonian gives $a_1 = v_{1_2} = 0.5$, which is the exact solution for the discretized problem. ■

The process of the graphical representation of the discretized functional using the nodal and element graphs is referred to as ‘assembly.” Each node and element is endowed with a nodal and element graph, respectively. The effective site-site interaction energy is estimated by summing the nodal

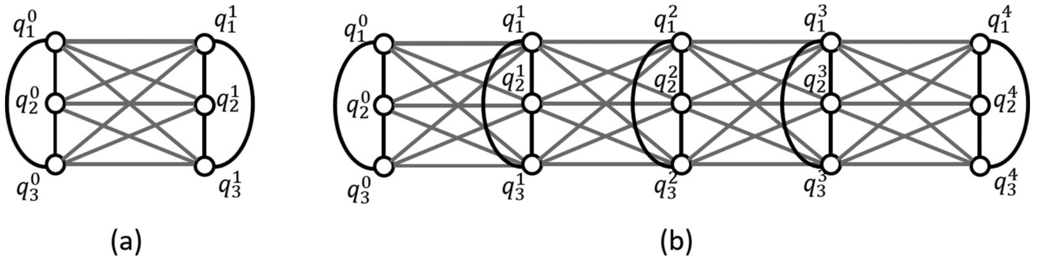


FIG. 4. Assembled graph for a domain discretized with (a) one element and (b) four elements.

coupling strength, $\hat{J}$, over all nodes, and element coupling strength, $\widetilde{J}$, over all elements. Due to the nature of discretization, N element graphs and $N+1$ nodal graphs are required for representing an N-element discretization of the domain. The assembled graph, from here on, is referred to as the logical graph. Connectivity of logical graphs for one-element and four-element discretization is presented in Fig. 4.

Two fundamental issues in this approach are addressed next using the box algorithm. First, the choices at a node $\{v_{i_1}, v_{i_2}, v_{i_3}\}$ were set in stone during initialization. The box algorithm makes this choice flexible. Second, as the number of nodes increases, three choices are insufficient. The number of qubits needed at a node must increase to make more choices available. The box algorithm, however, only requires three qubits per node for any level of discretization.

IV. BOX ALGORITHM

In this section, an iterative procedure is developed to minimize the functional, Π_N, using the graph coloring representation discussed in the previous section. For a particular choice of $\{v_{i_1}, v_{i_2}, v_{i_3}\}$, defined as Eq. (13), the possible values of the state a_i at the ith node are specialized to the set $\{u_i^c - r, u_i^c, u_i^c + r\}$, i.e.,

$$v_{i_j} = u_i^c + r(j-2). \tag{13}$$

The quantities $\mathbf{u^c} = (u_0^c, u_2^c, \dots, u_N^c)$ and r are hereafter referred to as the box center and the slack variable, respectively. The intention is to approximate functions using the box center while the slack variable provides a bound on this approximation. The precise meaning of this bound is presented later in this section. A linear approximation of $f(x) = \sqrt{x}$ using ten nodes is presented in Fig. 5 for different box centers and the slack variable (which can be interpreted as the box size). The function $f(x)$ is approximated as $\mathbf{u^c}$ at the nodes with linear interpolation in between the nodes. The blue region describes the possible value of the interpolation if the value at any node is perturbed within the range of $\pm r$. In Fig. 5(a), an exact approximation of the function at the nodes is presented with a slack variable of 0.2. In Fig. 5(b) the same approximation with a slack variable of 0.02 is presented. The same approximation is given in the two cases, but the bound on nodal values of Fig. 5(b) is tighter than in Fig. 5(a). In Fig. 5(c), the approximation is not exact; however, it lies within the bounds. In Fig. 5(d), the approximation is neither exact nor within the bound. In the context of the vectorial representation of the coefficients, $\mathbf{a}$, these bounds are represented as $3^N - 1$ points on the surface of a box, defined as $||\mathbf{a} - \mathbf{u^c}||_\infty = r$. An illustration for the vectorial representation of a two-node element is presented in Fig 6. The solution is sampled from a 3×3 grid in the $a_1 - a_2$ vector space.

In the discrete setting, the solution to the differential equation can be equivalently reduced to minimization of a function of the form $\mathbf{a^T M a}$, where $\mathbf{M}$ is some positive-definite matrix. The vector $\mathbf{a}$ takes value in one of the 3^N possibilities. The minimizer (not necessarily unique) is given by Eq. (14):

$$\mathbf{a}^{\min} = \underset{a_i \in \{u_i^c - r,\, u_i^c,\, u_i^c + r\}}{\arg\min} \mathbf{a^T M a}. \tag{14}$$

The solution, $\mathbf{a}^{\min}$, need not coincide with the best approximation solution, $\mathbf{a}^{b.a.}$, of the continuous problem. In the illustration presented in Fig. 6, the center is depicted as the solution ($\mathbf{a}^{\min} = \mathbf{u}^c$); the minimum is then contained within the elliptic region of the contour with $\mathbf{a}^{\min}$ on the edge. Geometrically, this gives $||\mathbf{a}^{\min} - \mathbf{a}^{b.a}|| \leqslant d \leqslant \sqrt{2} r(1 + \lambda_{\max}/\lambda_{\min})$ where $\lambda_{\max}$ and $\lambda_{\min}$ are the maximum and the minimum nonzero eigenvalues of M, respectively. This suggests that, as the box size decreases, the corresponding $\mathbf{u}^c$ approaches the best approximation solution of u. This argument is extended to n dimensions with the bound given in Eq. (15):

$$||\mathbf{a}^{\min} - \mathbf{a}^{b.a}|| \leqslant 2\left(1 + (n-1)\frac{\lambda_{\max}}{\lambda_{\min}}\right)\frac{r}{\sqrt{n}}. \tag{15}$$

A. Iterative procedure

In this section we present the details of the iterative procedure which locates the solution of the discretized problem, $\mathbf{a}^{\min}$, and updates the box center and slack variable such that $\mathbf{u}^C$ approaches the solution of the continuous problem (in the sense of best approximation).

The necessary information required for defining the functional is stored in the vector $\mathbf{S_i}$. It is computed once at the beginning of the procedure as the problem definition stage. The procedure is initiated with a guessed solution of the vector, $\mathbf{a}$, provided as a box centered at $\mathbf{u^c}$. The boundary nodes with Dirichlet boundary conditions are assigned the boundary value as the initial guess. The slack variable is initialized with an arbitrary scalar value. A better initial guess for r is the one which bounds the solution in the box defined by $\mathbf{u^c}$. Such initial guesses require fewer iterations in comparison to arbitrary ones; however, starting with a good guess is not a necessary condition for convergence. The Ising parameters, H, $\hat{J}$, and $\widetilde{J}$, are estimated as discussed in Sec. III.

In this work, D-Wave's 2000Q processor is used. This processor has a chimera-type structure with 2048 qubits and 6016 couplers [21]. A direct solution of the optimization

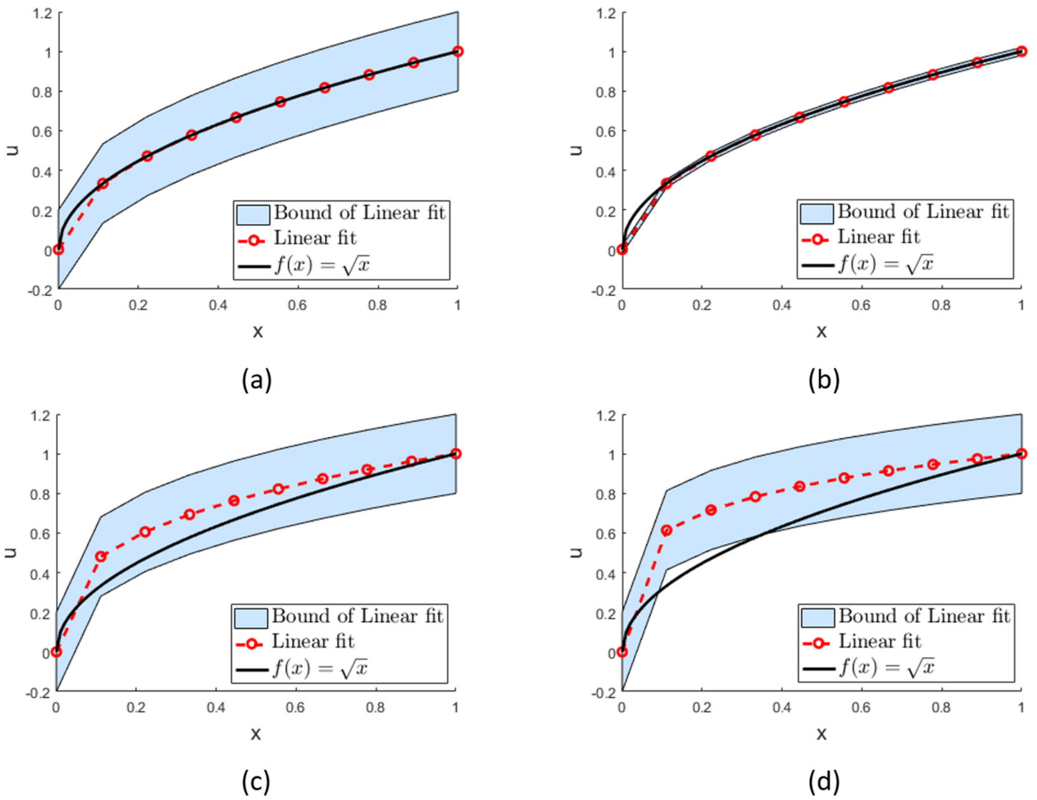


FIG. 5. Approximation of $\sqrt{x}$ function using boxed domain: (a) exact fit with a slack of 0.2 (loose fit), (b) exact fit with a slack of 0.02 (tight fit), (c) inexact fit but bounded in a box size of 0.2, and (d) inexact fit and unbounded by a slack of 0.2.

problem by renumeration of qubits is not possible as the assembled graph cannot be found in any subgraph of the physical graph, i.e., the processor. Therefore, it is required that the logical graph is mapped onto the physical graph via the process of embedding. This problem in itself is NP-hard and is not discussed here for brevity. The reader is referred to Ref. [22] for a discussion on this topic. The topology of the logical graph remains unchanged over the iterations. The search for embedding of a map is only conducted once, and in subsequent iterations the self-interaction and the site-site interaction parameters are updated for the same embedding.

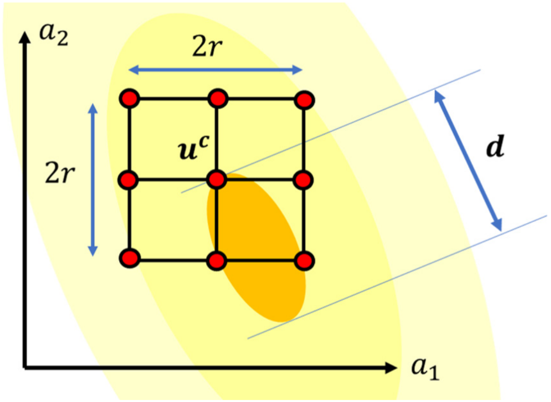


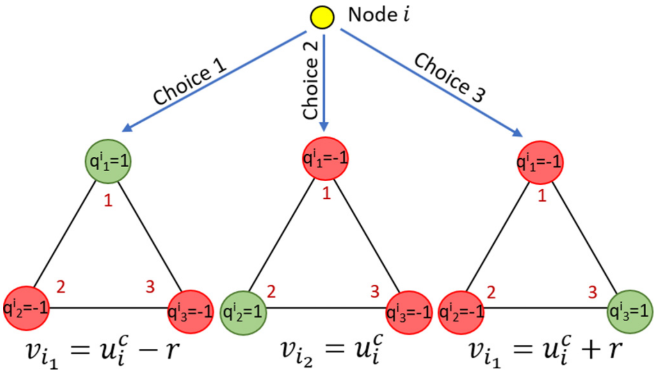
FIG. 6. An illustration of a two-node approximation in $a_1 - a_2$ vector space with contour plot of the functional, $\Pi_2(a_1, a_2)$, and a representative box with center at $\mathbf{u^c}$ and a box size of r.

The use of three qubits per node in this paper allows the D-Wave system to search for a minimum over a space of 3^N solution vectors in a single run. In each iteration, the box center is translated to the energy minimizer, $\mathbf{a}^{\min}$. This move is referred to as the translation step. In the case where the minimizing state is found at the center, the box size is reduced, and the search is continued with a smaller bound on error. This move is referred to as the contraction step. The complete procedure is presented in Algorithm 1.

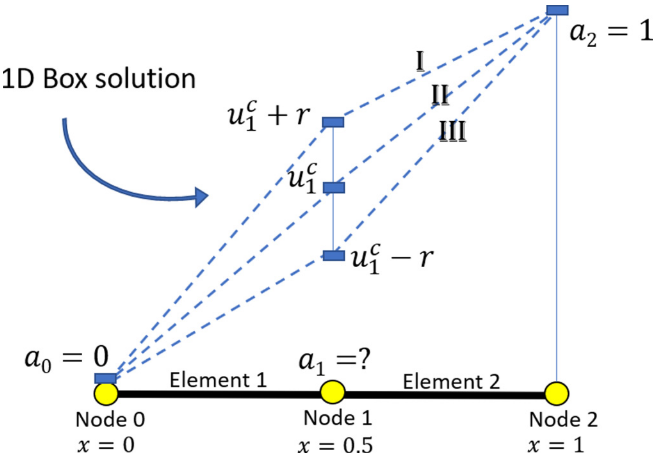
Algorithm 1 Box Algorithm

```
Problem definition: Calculate $\mathbf{S_i}$
Initialize $\{u_i^c\}, r$
Estimate $H$, $\hat{J}$ and $\widetilde{J}$
Find embedding: Logical graph $\xrightarrow{\text{embed}}$ Physical graph
while $r > r_{\min}$ do
    Update $\widetilde{J}$ for current $(u_i^c, r)$
    Anneal for $\{q_j^i\}$
    Map to $\mathbf{a}^{\min}$, $(\Pi_N)_{\min}$
    if $(\Pi_N)_{\min} < \Pi_N[\mathbf{u}^c]$ then
        $u_i^c = \mathbf{a}^{\min}$ (Translation step)
    else
        $r = \frac{r}{2}$ (Contraction step)
end
```

Example 4. In the box algorithm, the set $\{v_{i_1}, v_{i_2}, v_{i_3}\}$ is constructed using the box center and the slack variable:



With the application of the boundary condition, the favorable labeling of qubits gives the following three choices in the solution: **I**, **II**, or **III**.



The values of u^c_1 and r are initialized arbitrarily. One of the solutions among I, II, or III is selected by the annealer. If the minimizer is found to be solution II then the algorithm proceeds with the contraction step by halving the value of r. If solution I or III is selected, then the algorithm proceeds with the translation step by setting the new box center to $u^c_1 + r$ or $u^c_1 - r$, respectively. ■

V. RESULTS

The deformation of a bar under axial loading is modeled using an equation of form (2). In particular, the deflection (u) of a bar is related to the elastic stiffness (E), cross-section area (A), and the applied body force (f) using Eq. (16):

$$(EAu')' + f = 0. \tag{16}$$

The functional [Eq. (3)] is referred to as the potential energy of the system. The corresponding discretized form of the potential energy for piecewise linear E, A, and f is calculated using Eq. (17), where E_i, A_i, and f_i represent the elastic stiffness, area, and applied body force, respectively, at the center of the ith element:

$$\Pi_N[u] = \sum_{i=1}^{N} \frac{N}{2} E_i A_i (a_i - a_{i-1})^2 - \frac{1}{2N} f_i (a_i + a_{i-1}). \tag{17}$$

Two test cases are presented in Fig. 7. In the test case in Fig. 7(a), a bar with a discontinuity in EA is simulated. The body force is not applied in this case. A four-element discretization is used. The initial guesses are taken as $\mathbf{u^c} = \{0, 0.25, 0.5, 0.75, 1\}$ and $r = 0.2$. The numerical solution is observed to approach the exact solution in this case. The convergence in the functional is also evident.

In the test case in Fig. 7(b), a bar with continuously varying EA is simulated. A linearly varying body force is supplied. A six-element discretization of the bar is used with $\mathbf{u^c} = \{0, \frac{1}{6}, \frac{2}{6}, \frac{3}{6}, \frac{4}{6}, \frac{5}{6}, 1\}$ and $r = 0.2$. Based on the theory of finite element methods, the exact minimization of the energy in discretized space leads to a stiffer solution in comparison to the exact solution. It is observed that the numerical solution approaches the exact solution at nodes which is characteristic of finite element methods. Energy is also observed to be converging towards the finite element solution u^{fem} in this case. The mismatch of u within the element is expected to decrease with refinement in discretization.

Some implementation details on the D-Wave architecture are relevant here. Although the mapping only requires three qubits per node, embedding of this graph into the chimera graph produced an overhead of nine qubits per node—constant over a range of discretizations. It is understandable since a complete graph of three qubits used to represent a node is not directly represented on the chimera graph. In the future, the use of two qubits to represent a node can also be explored. While this ensures we still sample a large enough (2^N vectors) solution space in a single run, we lose out on information on the box center energy that is important for reducing the range of the slack variable. However, it is possible to compute the solution at the box center classically. The overhead of performing classical solutions can be offset by the fact that a two-qubit representation has smaller complete subgraphs and is easier to embed in the physical graph. Another important task in quantum computing is error suppression. Quantum processors, unlike classical computers, do not have parity correction algorithms due to the no-cloning theorem. A compilation of popular methods for quantum error correction is presented in Ref. [23]. Energy rescaling is one of the simpler approaches and is employed in this work. Here, in the estimation of $\widetilde{J}^n$, the energy was rescaled to ensure that the energy gap between feasible and unfeasible states is increased while maintaining a similar energy landscape. This step is a heuristic remedy for minimizing noise in quantum computation and has no bearing on the theoretical convergence of the algorithm.

VI. CONCLUSIONS AND FUTURE WORK

Recent rapid developments in quantum annealers warrant further investigation into re-formulation of scientific computation problems as graph coloring problems. The use of quantum computing for solving differential equations has, to date, focused on the use of a gate-computing-based QLSA. This algorithm attempts to sample from the solution space of the linear system of equations $Ax = b$. In the quantum-annealer-based algorithm described here, we do not solve the system of equations. We instead map the discretized version of the energy function of the differential equation

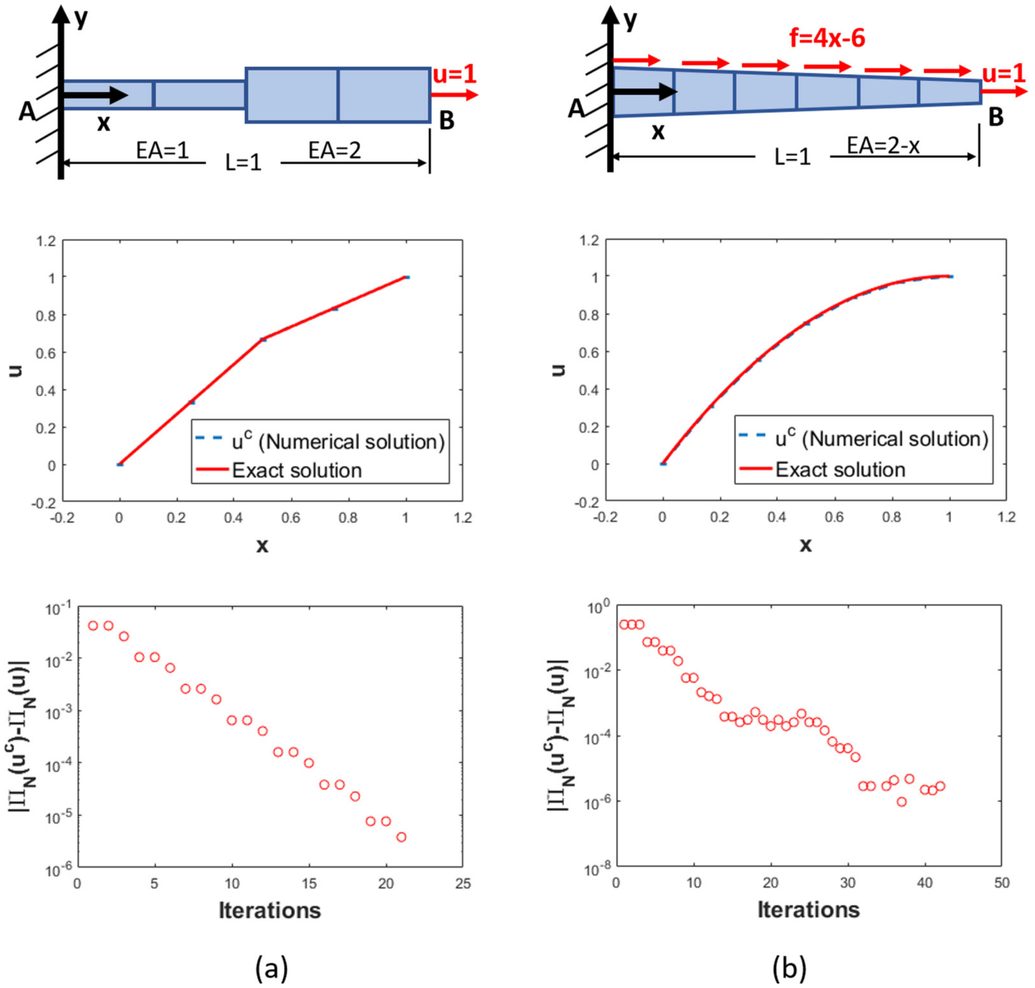


FIG. 7. Axial deformation of a bar with (a) a discontinuous cross section with a tip displacement and (b) a continuously varying cross section with a body force and a tip displacement.

to an Ising Hamiltonian. The solution vector, x, is directly obtained as the ground state of the qubits. The algorithm has low memory requirements since the global matrix is not stored and the local matrices are encoded in the interaction weights of the Ising model. Furthermore, the box algorithm allows mapping of up and down spin states of qubits in the ground state to rational numbers involved in the solution vector. Since we primarily solve the Ising model, the cost of computation is tied to the performance of the quantum annealer [24]. Within each iteration, however, Eq. (12) is solved for each element, leading to at least $O(n)$ operations.

We have shown that the box algorithm indeed guarantees convergence to the best approximation of the solution in the discretized space as the box size goes to zero. However, some improvements could be made to reduce the number of minimization runs. We could use the statistics of solutions that the D-Wave system returns from a single minimization run to drive the iteration process in an arbitrary direction. These data can also be used to heuristically develop "local" potential energy maps that can be used to identify larger step sizes for faster convergence. With future scaling up of quantum annealers up to millions of qubits, it will be possible to solve challenging engineering solid and fluid mechanics problems using quantum annealers.

ACKNOWLEDGMENTS

The authors would like to acknowledge Universities Space Research Association (USRA) for providing access to the use of the D-Wave quantum computer in the USRA-NASA-Google quantum Artificial Intelligence Laboratory at NASA's Ames Research Center. VS would like to acknowledge the Air Force Office of Scientific Research, contract FA9550-18-1-0091, for financial support.

[1] R. P. Feynman, Int. J. Theor. Phys. **21**, 467 (1982).

[2] A. W. Harrow, A. Hassidim, and S. Lloyd, Phys. Rev. Lett. **103**, 150502 (2009).

[3] A. Ambainis, Variable time amplitude amplification and quantum algorithms for linear algebra problems, in *STACS'12 29th Symposium on Theoretical Aspects of Computer Science*, (2012), Vol. 14, pp. 636–647.

[4] A. M. Childs, R. Kothari, and R. D. Somma, SIAM J. Comput. **46**, 1920 (2017).

[5] B. D. Clader, B. C. Jacobs, and C. R. Sprouse, Phys. Rev. Lett. **110**, 250504 (2013).

[6] L. Wossnig, Z. Zhao, and A. Prakash, Phys. Rev. Lett. **120**, 050502 (2018).

[7] D. W. Berry, J. Phys. A: Math. Theor. **47**, 105301 (2014).

[8] S. K. Leyton and T. J. Osborne, arXiv:0812.4423.

[9] A. Montanaro and S. Pallister, Phys. Rev. A **93**, 032324 (2016).

[10] Y. Cao, A. Papageorgiou, I. Petras, J. Traub, and S. Kais, New J. Phys. **15**, 013021 (2013).

[11] J. Pan, Y. Cao, X. Yao, Z. Li, C. Ju, H. Chen, X. Peng, S. Kais, and J. Du, Phys. Rev. A **89**, 022313 (2014).

[12] R. Steijl and G. N. Barakos, Comput. Fluids **173**, 22 (2018).

[13] P. Costa, S. Jordan, and A. Ostrander, Phys. Rev. A **99**, 012323 (2019).

[14] F. Fillion-Gourdeau and E. Lorin, Simple digital quantum algorithm for symmetric first-order linear hyperbolic systems. *Numerical Algorithms*, (2018), pp. 1–37.

[15] H. Sun and J. Zhang, Control Theory Technol. **15**, 267 (2017).

[16] S. R. Lee, F. J. Alexander, K. M. Barros, M. G. Daniels, J. R. Gattiker, M. S. Hamada, J. W. Howse, J. Loncaric, S. D. Pakin, R. D. Somma *et al.*, Los Alamos National Laboratory Technical Report No. LA-UR-16-22253, 2016 (unpublished) https://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-16-22253.

[17] Z. Bazant and L. Cedolin, *Stability of Structures: Elastic, Inelastic, Fracture and Damage Theories* (World Scientific, Singapore, 2010), https://books.google.com/books?id=CmnBPcLy2A4C.

[18] A. Borle and S. J. Lomonaco, Analyzing the quantum annealing approach for solving linear least squares problems, in *International Workshop on Algorithms and Computation* (2019) pp. 289-301.

[19] D. O'Malley, V. V. Vesselinov, B. S. Alexandrov, and L. B. Alexandrov, PLoS One **13**, e0206653 (2018).

[20] B. M. Levitan, I. S. Sargsian, and I. S. Sargsjan, *Introduction to Spectral Theory: Self-adjoint Ordinary Differential Operators*, Vol. 39 (American Mathematical Society, Providence, 1975).

[21] D-Wave Systems Inc., Technical description of the d-wave quantum processing unit (09-1109a-e, 2018) https://docs.dwavesys.com/docs/latest/_downloads/09-1109A-O_Technical_Description_of_DW_QPU.pdf.

[22] J. Cai, W. G. Macready, and A. Roy, arXiv:1406.2741.

[23] S. J. Devitt, W. J. Munro, and K. Nemoto, Rep. Prog. Phys. **76**, 076001 (2013).

[24] C. C. McGeoch and C. Wang, in *Proceedings of the ACM International Conference on Computing Frontiers* (ACM, New York, 2013), p. 23.