

# A quantum annealing-sequential quadratic programming assisted finite element simulation for non-linear and history-dependent mechanical problems

Van-Dung Nguyen<sup>a,1</sup>, Ling Wu<sup>a</sup>, Françoise Remacle<sup>b</sup>, Ludovic Noels<sup>a,\*</sup>

<sup>a</sup>*Computational & Multiscale Mechanical of Materials (CM3),  
Department of Aerospace and Mechanical Engineering,  
University of Liège,  
Quartier Polytech 1, Allée de la Découverte 9, B-4000 Liège, Belgium*  
<sup>b</sup>*MolSys Research Unit,  
University of Liège,  
Quartier Agora, Allée du six Août 11, B-4000 Liège, Belgium*

---

## Abstract

We propose a framework to solve non-linear and history-dependent mechanical problems based on a hybrid classical computer – quantum annealer approach. Quantum Computers are anticipated to solve particular operations exponentially faster. The available possible operations are however not as versatile as with a classical computer. However, quantum annealers (QAs) are well suited to evaluate the minimum state of a Hamiltonian quadratic potential. Therefore, we reformulate the elasto-plastic finite element problem as a double-minimisation process framed at the structural scale using the variational updates formulation. In order to comply with the expected quadratic nature of the Hamiltonian, the resulting non-linear minimisation problems are iteratively solved with the suggested Quantum Annealing-assisted Sequential Quadratic Programming (QA-SQP): a sequence of minimising quadratic problems is performed by approximating the objective function by a quadratic Taylor's series. Each quadratic minimisation problem of continuous variables is then transformed into a binary quadratic problem. This binary quadratic minimisation problem can be solved on quantum annealing hardware such as the D-Wave<sup>1</sup> system. The applicability of the proposed framework is demonstrated with one- and two-dimensional elasto-plastic numerical benchmarks. The current work provides a pathway of performing general non-linear finite element simulations assisted by quantum computing.

Preprint submitted to European Journal of Mechanics / A Solids. (C) 2024; Licensed under the Creative Commons (CC-BY-NC-ND); formal publication on: [1016/j.euromechsol.2024.105254](https://doi.org/10.1016/j.euromechsol.2024.105254)

*Keywords:* Quantum computing, Quantum annealing, Variational formulation, Finite

---

\*Corresponding author, Phone: +32 4 366 48 26, Fax: +32 4 366 95 05

*Email addresses:* [vandung.nguyen@uliege.be](mailto:vandung.nguyen@uliege.be) (Van-Dung Nguyen), [L.Wu@ulg.ac.be](mailto:L.Wu@ulg.ac.be) (Ling Wu), [fremacle@ulg.ac.be](mailto:fremacle@ulg.ac.be) (Françoise Remacle), [L.Noels@ulg.ac.be](mailto:L.Noels@ulg.ac.be) (Ludovic Noels)

<sup>1</sup>Postdoctoral Researcher at the Belgian National Fund for Scientific Research (FNRS)

<sup>1</sup>D-Wave trademarks and registered trademarks used herein include D-Wave and Ocean.

## 1. Introduction

Continuum mechanics represents the behaviour of solids and fluids through a system of complex non-linear Partial Differential Equations (PDEs). Whilst analytical solutions can be found for some simple cases, finite element (FE) method is a standard tool widely used to find approximate solutions, which is based on the discretisation of the continuum domain into a finite element mesh consisting of a finite number of elements and nodes. The physical field is thus approximated inside each element using a limited number of unknowns, which are often the nodal values, and which interpolate the fields through their associated shape functions. Such a procedure allows turning the continuous PDEs into a discrete form with a finite number of continuous degrees of freedom, and can be solved by different numerical techniques [1]. Only in a very limited situation, in which the underlying material behaviour is linear elastic, yields linear governing PDEs. Therefore, high computational resources are needed when the computation domain is very large, because of the very large resulting number of degrees of freedom, and/or when complex physical behaviours such as plasticity, damage and fracture are involved, because of the highly non-linear nature of the resulting PDEs.

Quantum computing is a rapidly developing field of computer science that uses the principles of quantum mechanics. Unlike classical computers, which store and manipulate information using bits represented by binary values (0s and 1s), quantum computers use quantum bits or so-called qubits, which can take values of 0 or 1, but which can also exist in a superposition of both 0 and 1 at the same time. Taking advantage of quantum effects allows quantum computers to potentially solve particular kinds of problems that are intractable for classical computers [2], see also the discussions in [3, 4].

There exist two main quantum implementations. On the one hand, gate-based quantum computers rely on so-called quantum gates to manipulate qubits, taking advantage of quantum superposition and entanglement to perform computations [5]. Quantum gates are analogous to classical logic gates, but operate on qubits instead of bits as in classical computers. Gate-based quantum computers are currently the most advanced type of quantum computer for universal computations. However, they require extremely precise control over the qubits and their environment, and are currently limited in size by the difficulty of maintaining this control. Waiting for efficient Quantum Error Correction (QECs), gate-based quantum computers are practically of the noisy intermediate-scale quantum class computers [6]. On the other hand, quantum annealers embracing quantum annealing technologies [7–9] are designed to solve optimisation problems by starting with a system of qubits in a simple state and then slowly modifying their interactions or entanglements until they settle into a state that represents the optimal solution of the problem. Quantum annealers are typically used to solve combinatorial optimisation problems and are not as versatile as gate-based quantum computers, which can run a wider range of algorithms.

Application of quantum computing in finite element simulations is a tempting target, see the review by [10]. A typical numerical technique, such as the finite element method, follows an iterative procedure in which a linear system needs to be solved at every iteration, and its solution is used to update the current solution until a terminating criterion is met [1]. In

this context, the exciting recent advance in the development of quantum algorithms in gate-based quantum computers for solving a large system of linear equations allows potentially solving large FE problems on quantum computers. Indeed, quantum linear solvers [11, 12] could provide an exponential quantum speed-up over classical algorithms when solving linear systems. The system of linear equations arising from the FE method is sparse, thus satisfying an important requirement of the exponential quantum speed-up [13]. However, there are still many challenges that need to be overcome before it can be effectively used for FE simulations. Firstly, quantum computers are still in their early stages as the current quantum hardware is sensitive to noise and has high error rates, in particular for gate-based quantum computers, which makes it difficult to implement and execute quantum algorithms accurately and efficiently. Secondly, the existing FE algorithms are designed to run on classical computers: the resolution of the system of linear equations is only one simple step in the workflows, and the remaining operations are not readily adaptable to quantum computing architectures. However, many of the steps in a typical finite element simulation can rely on classical computations, so integrating quantum algorithms into classical computing could be considered [14], requiring the development of an efficient hybrid classical computer – quantum annealer approach that can effectively use both quantum and classical resources.

Quantum annealers can be used to solve the system of linear equations, but this requires reformulating the problem to account for continuous variables. A general rule to apply quantum annealers is first to define the solution of this linear system as minimisation of a quadratic objective function, then to approximate each continuous variable by a discrete representation using the radix-2 approximation, leading to a binary quadratic form, which is an objective function compatible with quantum annealers [15–18]. Each binary variable whose value can be either 0 or 1 is represented by a qubit in the quantum annealers. At the end of the annealing process being programmed to minimise the binarised objective function, the measurement of the qubits provides the optimised solution, offering a speed advantage over classical methods. However, the solution precision in this so-called direct annealing approach is limited, on the one hand by the number of qubits used because of the radix-2 encoding, and on the other hand by the fact that the ground state solution of the quantum system is not guaranteed due to noise and errors in the quantum annealers. These issues can be improved using the so-called iterative annealing approach in which a series of minimisation problems are performed by quantum annealers to get better radix-2 approximation [19] with a lower number of qubits per degree of freedom until a terminating criterion is met. Such an iterative procedure generally does not require the correct ground state of the quantum system to be reached, but only to reach a better solution after each iteration.

Quantum annealers were directly considered in the context of the FE method. The box algorithm proposed in [20] consists in iteratively mapping each degree of freedom with 8 discrete values using 3 qubits and performing the quantum annealing to find the optimised solution until the desired precision is met. An iterative FEqa formulation was proposed in [21], in which each unknown is mapped into a single qubit and it was shown that the FEqa has clear advantages in computational time when considering the quantum annealing instead of considering simulated annealing. An iterative algorithm to solve the elliptic problems on the D-Wave quantum annealing device was proposed [19], in which a fixed number of qubits is used for each degree of freedom parameterised by a radix-2 approximation, the range of each degree of freedom being adjusted after each iteration. Nevertheless, these methods

are considered only in the context of the linear elasticity. Quantum annealing was also considered in the context of predicting phase separation with the phase-field method [22], in which the potential arising from the physical energies (the interaction –internal– energy term, the gradient –adjacent– energy term, and the long-distance energy term) at play has to be minimised. In this approach the unknowns are discretised by 4 binary variables, without continuous representation.

The present work focuses on developing a hybrid classical computer – quantum annealer framework for general non-linear and history-dependent finite element simulations. First, we formulate the elasto-plastic solid mechanics boundary value problem as non-linear minimisation problems using the variational updates formulation [23–25], leading to a so-called double-minimisation problem. Not only the unknowns field, *e.g.* displacements, but also the internal states, *e.g.* plastic deformation gradients, are obtained through minimisation processes within each load increment. Contrarily to what is classically done in finite element codes, in which the material model equations are solved locally at the Gauss point level whilst only the nodal degrees of freedom are solved at the structural scale in a nested way, in order to take advantage of the potential ability of quantum computing to solve large scale problem, this double-minimisation problem is expressed in terms of both the unknown field and internal states at the structural scale, so that only assembly operations have to be performed on the classical hardware. Then, in order to comply with the requested quadratic nature of the minimisation problems that can be solved by quantum annealers, a Sequential Quadratic Programming (SQP) with quantum annealing –or so-called Quantum Annealing-assisted Sequential Quadratic Programming (QA-SQP)– is suggested to solve a general non-linear minimisation problem using quantum annealers. Although, in general, a minimisation can be carried out by iteratively carrying out many minimisation problems of a single variable, *e.g.* line search, steepest descent, conjugate gradient methods *etc.* [26, 27], in this work, a Sequential Quadratic Programming (SQP) iteratively approximates the objective function using a quadratic Taylor’s series and solves a sequence of minimisation quadratic problems [28]. SQP is typically used to find the minimum of a non-linear function subject to constraints, such as inequality constraints or equality constraints. In the developed QA-SQP, the quadratic minimisation problems are solved by quantum annealing using an adaptive binary approximation to transform the continuous quadratic minimisation problems into quadratic binary problems.

The paper is organised as follows. Section 2 briefly introduces the quantum annealing working principles. Section 3 provides the mathematical foundation of the variational updates formulation in which we detail the minimisation principle of the mechanical boundary value problem in the context of a spatial finite element discretisation. Section 4 introduces the Quantum Annealing-assisted Sequential Quadratic Programming (QA-SQP) and its particularisation to the resolution of the double-minimisation problem formulated in Section 3. Section 5 presents numerical benchmarks with the proposed framework. Finally, conclusions are drawn in Section 6.

## 2. Quantum annealing

Quantum annealing is based on the principle of adiabatic quantum computation, which involves gradually transforming a quantum system from an initial Hamiltonian, in which its

ground state, *i.e.* the stationary state of the lowest energy, can be easily prepared, to a final Hamiltonian, in which the problem is encoded and whose ground state provides the solution.

### 2.1. Quantum state representation

The quantum states refer to the possible configurations in which the quantum system can exist. The key concepts related to the quantum states which make quantum computing more powerful than the classical one consist of the principle of superposition, in which a quantum system can exist in multiple states simultaneously, and of entanglement, in which the states of two or more particles become correlated in such a way that the state of one particle influences the one(s) of the other(s) regardless of the distance between them. Understanding and manipulating the quantum states are crucial in quantum computing. More comprehensive details about quantum computing can be found in [5, 29]. In this section, we briefly summarise the mathematical representation of quantum states.

In quantum computing, a quantum bit (so-called qubit) is the fundamental information unit that can be seen as the generalisation of a binary bit in classical computers. A qubit is a two-state quantum-mechanical system whose state can be represented by any normalised linear combination of the two basis states, which are usually denoted as  $|0\rangle = [1\ 0]^T$  and  $|1\rangle = [0\ 1]^T$ . The Dirac notation  $|\bullet\rangle$  is used to represent a state of a quantum system, as

$$|\phi\rangle = \phi_0 |0\rangle + \phi_1 |1\rangle , \quad (1)$$

where  $\phi_0$  and  $\phi_1$  are complex numbers satisfying  $|\phi_0|^2 + |\phi_1|^2 = 1$  to ensure that the total probability of being in any of the basis states is equal to 1. Upon measurement, a qubit loses its quantumness and its superposition state (1) is transformed into classical information, either  $|0\rangle$  with the probability  $|\phi_0|^2$  or  $|1\rangle$  with the probability  $|\phi_1|^2$ .

A system consisting of  $K$  qubits is a  $2^K$ -state quantum-mechanical system whose state can be represented by any normalised linear combination of  $2^K$  basis states following

$$|\phi\rangle = \phi_0 |0\rangle \otimes \dots \otimes |0\rangle \otimes |0\rangle + \phi_1 |0\rangle \otimes \dots \otimes |0\rangle \otimes |1\rangle + \phi_{2^{K-1}} |1\rangle \otimes \dots \otimes |1\rangle \otimes |1\rangle , \quad (2)$$

where  $\otimes$  denotes the Kronecker product<sup>2</sup>, and  $\phi_0, \dots, \phi_{2^{K-1}}$  are complex numbers satisfying the normalisation condition  $\sum_{i=0}^{2^K-1} |\phi_i|^2 = 1$ .

### 2.2. Principle of quantum annealing

The objective of the quantum annealing is to find the ground state of the Hamiltonian  $\mathbf{H}$ , *i.e.* solving

$$|\phi_0\rangle = \arg \min_{|\phi\rangle} \langle \phi | \mathbf{H} | \phi \rangle , \quad (3)$$

where the bra notation  $\langle \bullet |$  corresponds to the conjugate transpose of the ket notation  $|\bullet\rangle$ . The state  $|\phi_0\rangle$  is called the ground state of  $\mathbf{H}$  and  $E_0 = \langle \phi_0 | \mathbf{H} | \phi_0 \rangle$  is the ground-state energy

---

<sup>2</sup>The Kronecker product  $\otimes$  is commonly used in quantum computing. The Kronecker product of two matrices  $\mathbf{A} \in \mathbb{C}^{m \times n}$  and  $\mathbf{B} \in \mathbb{C}^{p \times q}$  results in a matrix  $\mathbf{C} \in \mathbb{C}^{mp \times nq}$  as  $C_{ab} = A_{ij}B_{kl}$ , where  $a = pi + k$  and  $b = qj + l$  with  $i = 0, \dots, m - 1$ ,  $j = 0, \dots, n - 1$ ,  $k = 0, \dots, p - 1$ , and  $l = 0, \dots, q - 1$ .

which corresponds to the minimum eigenvalue of  $\mathbf{H}$  with  $|\phi_0\rangle$  being its associated eigenvector [30].

Quantum annealing is based on the principle of adiabatic quantum computation. The basic idea is to start with a simple Hamiltonian  $\mathbf{H}_0$  whose ground state is known and which can be prepared easily, and then gradually transform this Hamiltonian into the problem Hamiltonian  $\mathbf{H}$  whose ground state encodes the solution to the optimisation problem at hand. The time-dependent Hamiltonian of the system reads

$$\mathbf{H}_{\text{QA}}(t) = A(t)\mathbf{H}_0 + B(t)\mathbf{H}, \quad (4)$$

where  $A(t)$  and  $B(t)$  are annealing functions satisfying  $A(0) = 1 \gg B(0)$  and  $B(t_a) = 1 \gg A(t_a)$  where  $t_a$  is the annealing time. By slowly varying the Hamiltonian from 0 to  $t_a$ , the system remains in its ground state throughout the process as the result of the adiabatic theorem [31], and, eventually, the final state of the system corresponds to the sought solution of the considered optimisation problem. The details of such a annealing process can be found in the review [32].

### 2.3. Ground state of an Ising Hamiltonian

We consider the problem of finding the ground state of the Ising Hamiltonian [7–9], which is represented by an undirected graph  $(V, E)$  where  $V = \{0, \dots, K-1\}$  is a set of  $K$  qubits and  $E$  specifies the set of interactions between two qubits, *i.e.*  $E \subset \{(i, j) | i \in V, j \in V, \text{ and } i < j\}$  as

$$\mathbf{H} = \sum_{i \in V} h_i \mathbf{H}_i + \sum_{(i, j) \in E} J_{ij} \mathbf{H}_{ij}, \quad (5)$$

where  $h_i \neq 0$  and  $J_{ij} \neq 0$ , with  $i \in V$  and  $(i, j) \in E$ , are constants, and  $\mathbf{H}_i$ , and  $\mathbf{H}_{ij}$  are the Pauli-Z operators respectively applied on the qubit  $i$  and on the qubits  $i$  and  $j$  as

$$\mathbf{H}_i = \underbrace{\mathbf{I}}_0 \otimes \dots \otimes \mathbf{I} \otimes \underbrace{\mathbf{Z}}_i \otimes \mathbf{I} \otimes \dots \otimes \underbrace{\mathbf{I}}_{K-1}, \text{ and} \quad (6)$$

$$\mathbf{H}_{ij} = \underbrace{\mathbf{I}}_0 \otimes \dots \otimes \mathbf{I} \otimes \underbrace{\mathbf{Z}}_i \otimes \mathbf{I} \otimes \dots \otimes \mathbf{I} \otimes \underbrace{\mathbf{Z}}_j \otimes \mathbf{I} \otimes \dots \otimes \underbrace{\mathbf{I}}_{K-1}, \quad (7)$$

$$\text{with } \mathbf{Z} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \text{ and } \mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (8)$$

It can be seen that  $\mathbf{H}$  in Eq. (5) is a  $2^K \times 2^K$  diagonal operator in the computational basis. Consequently, its ground state corresponds to the one of its computational basis. Assuming  $|\phi\rangle = |b_0 \dots b_{K-1}\rangle$ , with  $b_i \in \{0, 1\} \forall i = 0, \dots, K-1$ , is the computational basis of  $\mathbf{H}$ , its associated eigenvalue reads<sup>3</sup>

$$\mathcal{F}_{\text{Ising}} = \sum_{i \in V} h_i q_i + \sum_{(i, j) \in E} J_{ij} q_i q_j = \mathbf{q}^T \mathbf{h} + \mathbf{q}^T \mathbf{J} \mathbf{q}, \quad (9)$$

---

<sup>3</sup>This relation results from the definition of Eq. (5) and the use of the following properties of  $\mathbf{Z}$ ,  $\mathbf{H}_i$ , and

where  $q_i = (-1)^{b_i} \in \{-1, 1\}$ , which is known as the spin variable,  $\mathbf{q} = [q_i \forall i \in V]$  is the spin vector,  $\mathbf{h} = [h_i \forall i \in V]$  is the biases vector, and  $\mathbf{J} = [J_{ij} \text{ if } (i, j) \in E; 0 \text{ otherwise}]$  is a strictly upper diagonal matrix.

As a result, finding the ground state of the Ising Hamiltonian (5) is equivalent to minimising the quadratic function of the spin variables (9), yielding the optimisation problem

$$\mathbf{q} = \arg \min_{\mathbf{q}'} \mathcal{F}_{\text{Ising}}(\mathbf{q}'; \mathbf{h}, \mathbf{J}), \quad (10)$$

where  $\mathbf{h}$  and  $\mathbf{J}$  are the user programmable parameters.

The minimisation of  $\mathcal{F}_{\text{Ising}}$  stated by Eq. (10) belongs to the class of problems known as quadratic unconstrained binary optimisation (QUBO). Indeed, using the spin-binary variable transformation  $q_i = 2b_i - 1 : \{0, 1\} \rightarrow \{-1, 1\}$ , and the property  $b_i^2 = b_i \forall i$ , the minimisation of  $\mathcal{F}_{\text{Ising}}$  stated by Eq. (10) is equivalent to the minimisation of  $\mathcal{F}_{\text{QUBO}}$  which takes the form

$$\mathbf{b} = \arg \min_{\mathbf{b}'} \mathcal{F}_{\text{QUBO}}(\mathbf{b}'; \mathbf{A}) \text{ with } \mathcal{F}_{\text{QUBO}}(\mathbf{b}; \mathbf{A}) = \sum_{(i,j) \in E \cup \{(i,i) \forall i \in V\}} A_{ij} b_i b_j = \mathbf{b}^T \mathbf{A} \mathbf{b}, \quad (11)$$

where  $\mathbf{A} = [A_{ij} \text{ for } (i, j) \in E \cup \{(i, i) \forall i \in V\}; 0 \text{ otherwise}]$  is the QUBO matrix, a user programmable parameter, and  $\mathbf{b} = [b_0 \dots b_{K-1}]^T$  is the vector of the unknown binary variables.

#### 2.4. Ising Hamiltonian-based quantum annealing

Quantum annealing considering the Ising Hamiltonian in Eq. (4) can be implemented on a hardware called quantum annealer [33], which, from the computational point-of view, provides a black-box optimiser for solving the minimisation problems (10) or (11) defined by their user programmable parameters, respectively  $\mathbf{h}$  and  $\mathbf{J}$ , or  $\mathbf{A}$ . Different problems can be mapped to such a form through these programmable parameters  $\mathbf{h}$  and  $\mathbf{J}$ , or  $\mathbf{A}$ .

Quantum annealers are devices designed to solve optimisation problems using quantum mechanics principles. Their outputs are generally probabilistic rather than deterministic since quantum mechanics inherently introduces probabilities into the outcomes. Quantum annealers may not converge to the global minimum on a single run due to various factors, such as environmental noises, hardware imperfections, and pre- and post-processing errors. In practice, the annealing cycle needs to be repeated multiple times per input to obtain multiple candidate solutions and the cycle resulting into the smallest energy is chosen to provide the solution to the optimisation problem [34]. The probability of finding the optimal or near-optimal solution is generally higher when increasing the number of annealing cycles per input. In general, the following steps are followed to solve a problem with quantum annealers (see *e.g.* [35]):

---

$\mathbf{H}_{ij}$ :

$$\begin{aligned} \mathbf{Z} |b_i\rangle &= (-1)^{b_i} |b_i\rangle, \\ \mathbf{H}_i |\phi\rangle &= |b_0 \dots b_{i-1}\rangle \otimes \mathbf{Z} |b_i\rangle \otimes |b_{i+1} \dots b_{K-1}\rangle = (-1)^{b_i} |\phi\rangle, \text{ and} \\ \mathbf{H}_{ij} |\phi\rangle &= |b_0 \dots b_{i-1}\rangle \otimes \mathbf{Z} |b_i\rangle \otimes \dots \otimes b_{j-1} \otimes \mathbf{Z} |b_j\rangle \otimes |b_{j+1} \dots b_{K-1}\rangle = (-1)^{b_i} (-1)^{b_j} |\phi\rangle. \end{aligned}$$

1. The problem under consideration ( $P_1$ ) has to be transformed into an Ising or a QUBO form ( $P_2$ ) following, respectively Eq. (10) or Eq. (11);
2. An embedding technique is applied to transform  $P_2$  into  $P_3$ , which matches the quantum processing units (QPU) topology representing the physical connectivity between the qubits;
3.  $P_3$  is sent to the QPU together with other user parameters. The graph of  $P_3$  is limited by the number of qubits and their available interactions in the QPU. The solution of  $P_3$  from the QPU is returned as a set of samples, in which the best solution is selected as the case corresponding to the smallest energy.

In recent years, the progress in developing quantum annealing hardware has gained remarkable acceleration, in particular available D-Wave QPUs are now being used in various applications [34, 36]. This work considers the most recent D-Wave QPU (Advantage QPU) possessing more than 5000 qubits with more than 35000 couplers. The maximum size of a fully connected Ising graph that can be embedded in the QPU is equal to 177 since each qubit is only connected to a reduced number of other qubits [37]. A sparse graph with 1 to the maximum number of available qubits can be solved as long as it fits the QPU topology.

The solution time offered by the D-Wave quantum annealers depends on the complexity of the optimisation problem, the number of qubits involved, and the other settings. It can be measured by the QPU access time, which mainly consists of the programming time (with a magnitude of the order of the millisecond) and the sampling time per annealing cycle (including annealing time, readout time, and delay time with a magnitude of the order of the microsecond), and by the number of cycles (from 1 to 10000 cycles in Advantage QPU). With the current Advantage QPU, the maximum time that a submitted problem is allowed to run is 1 second. More details on the solution time on D-Wave QPU can be found in [38].

### 3. Variational formulation of the history-dependent mechanical boundary value problem

In this section, the mechanical boundary value problem is formulated as a minimisation problem using the variational updates formulation [23–25]. However, contrarily to classical finite element approaches for which the material model is solved locally whilst the finite element degrees of freedom are solved at the structural scale in a nested way, we develop a variational formulation at the structural scale. This approach is motivated by the fact that these minimisation problems can be performed in the quantum annealers after a suitable continuous-discrete variables transformation. The resulting hybrid framework combining classical computers and quantum annealers will be detailed in Section 4. It is thus advantageous to solve both the material model and the structural governing equations using annealing whilst minimising the interface steps between the different hardware.

#### 3.1. Governing equations

Considering a body  $V \in \mathbb{R}^d$  where  $d$  is the problem dimension. Its boundary  $\partial V$  is divided into two non-overlapping parts:  $\partial_D V$ , the Dirichlet boundary where a prescribed displacement  $\mathbf{u}_0$  is enforced, and  $\partial_N V$ , the Neumann boundary where the traction per unit

reference surface  $\mathbf{t}_0$  is prescribed. The quasi-static equilibrium equations over the body  $V$  are given as

$$\nabla \cdot \boldsymbol{\sigma}(\mathbf{x}) + \mathbf{b}_0(\mathbf{x}) = \mathbf{0} \quad \forall \mathbf{x} \in V, \quad (12)$$

$$\mathbf{u}(\mathbf{x}) = \mathbf{u}_0(\mathbf{x}) \quad \forall \mathbf{x} \in \partial_D V, \quad \text{and} \quad (13)$$

$$\boldsymbol{\sigma}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) = \mathbf{t}_0(\mathbf{x}) \quad \forall \mathbf{x} \in \partial_N V, \quad (14)$$

where  $\nabla$  is the gradient operator,  $\boldsymbol{\sigma}$  is the Cauchy stress tensor,  $\mathbf{b}_0$  is the volumetric force, and  $\mathbf{n}$  is the outward unit normal to  $\partial_N V$ . By assuming a small strain formulation, a constitutive relation must be introduced in order to solve the displacement field  $\mathbf{u}$  as

$$\left\{ \begin{array}{l} \boldsymbol{\sigma}(\mathbf{x}) = \boldsymbol{\sigma}(\boldsymbol{\varepsilon}(\mathbf{x}); \mathbf{q}(\mathbf{x})), \quad \text{and} \\ \text{evolution laws for } \mathbf{q}, \end{array} \right. \quad (15)$$

where  $\boldsymbol{\varepsilon} = \nabla \otimes^s \mathbf{u}$  is the deformation tensor with  $\varepsilon_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$ , and  $\mathbf{q}$  is the vector of all the internal variables used to describe a history-dependent stress-strain relationship. In this work, we follow the variational principle in which the mechanical boundary problem can be described as a double-minimisation problem: one with respect to the displacement field  $\mathbf{u}(\mathbf{x})$ ,  $\forall \mathbf{x} \in V$ , and one with respect to the internal variables field  $\mathbf{q}(\mathbf{x})$ ,  $\forall \mathbf{x} \in V$ .

### 3.2. Double-minimisation formulation

Quantum annealing is particularly efficient to solve a minimisation problem, which is fully compatible with the underlying physics of the numerical methods developed in computational mechanics. Indeed, not only the mechanical balance equation (12-14), but also the constitutive model (15) can be cast under the form of an functional minimisation. For this purpose, the minimisation of energy functions is first formulated in an incremental way at each loading increment and then solved, leading to the solution in terms of not only the displacement field but also of the internal variables field. To this end, the variational updates formulation pioneered for the visco-plastic constitutive models [23, 24], and later extended to porous metal plasticity [25], elasto-plastic solids [39], thermo-mechanical coupling [40, 41], visco-elastic solids [42, 43], elasto-plastic-damage solids [44, 45], and multi-scale simulations [46, 47], is reframed in order to express the finite element and constitutive model resolutions as a double-minimisation problem expressed at the global or structural scale.

The existence of a Helmholtz free energy  $\Psi$  is postulated under the following form

$$\Psi = \Psi(\boldsymbol{\varepsilon}, \mathbf{q}), \quad (16)$$

explicitly depending on the strain tensor  $\boldsymbol{\varepsilon}$  and on a collection of history-dependent variables  $\mathbf{q}$  introduced to model history dependency. The Clausius-Duhem inequality imposes the following restriction

$$\mathcal{D} = \boldsymbol{\sigma} : \dot{\boldsymbol{\varepsilon}} - \dot{\Psi} \geq 0, \quad (17)$$

where  $\mathcal{D}$  is the dissipation. This inequality becoming an equality for any reversible process,

Eq. (16) results in the following stress-strain relationship

$$\boldsymbol{\sigma} = \frac{\partial \Psi}{\partial \boldsymbol{\varepsilon}}, \quad (18)$$

whilst the non-negativeness constraint on the dissipation for any reversible or irreversible process implies

$$\mathcal{D}(\mathbf{Y}, \dot{\mathbf{q}}) = \mathbf{Y} \cdot \dot{\mathbf{q}} \geq 0 \text{ with } \mathbf{Y} = -\frac{\partial \Psi}{\partial \dot{\mathbf{q}}}, \quad (19)$$

where  $\mathbf{Y}$  is the vector of internal forces conjugate to  $\mathbf{q}$ , and the operator “ $\cdot$ ” denotes the inner product between two vectors.

For problems involving history dependency, *i.e.*  $\mathbf{q} \neq \emptyset$ , the evolution laws specifying the evolution of the internal variables  $\mathbf{q}$  must be provided. In the case of standard materials [48], it is assumed that there exists a dissipation pseudo-potential  $\Theta(\dot{\mathbf{q}})$  and its convex dual  $\Theta^*(\mathbf{Y})$  derived by considering the Legendre transformation and by postulating the maximum dissipation, yielding

$$\Theta(\dot{\mathbf{q}}) = \max_{\mathbf{Y}} [\mathbf{Y} \cdot \dot{\mathbf{q}} - \Theta^*(\mathbf{Y})]. \quad (20)$$

The evolution law for  $\dot{\mathbf{q}}$  thus reads

$$\dot{\mathbf{q}} = \frac{\partial \Theta^*(\mathbf{Y})}{\partial \mathbf{Y}}, \text{ or equivalently } \mathbf{Y} = \frac{\partial \Theta(\dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}}. \quad (21)$$

Using Eq. (19), the local power functional  $\mathcal{E}(\dot{\boldsymbol{\varepsilon}}, \dot{\mathbf{q}})$  is defined from new independent variables  $(\dot{\boldsymbol{\varepsilon}}, \dot{\mathbf{q}})$  as

$$\mathcal{E}(\dot{\boldsymbol{\varepsilon}}, \dot{\mathbf{q}}) = \dot{\Psi} + \Theta = \frac{\partial \Psi}{\partial \boldsymbol{\varepsilon}} : \dot{\boldsymbol{\varepsilon}} - \mathbf{Y} \cdot \dot{\mathbf{q}} + \Theta. \quad (22)$$

It is clear that this local power functional  $\mathcal{E}(\dot{\boldsymbol{\varepsilon}}, \dot{\mathbf{q}})$  has to be minimised with respect to the internal variable rate since, using Eq. (21), one has

$$\frac{\partial \mathcal{E}(\dot{\boldsymbol{\varepsilon}}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}} = -\mathbf{Y} + \frac{\partial \Theta(\dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}} = \mathbf{0}. \quad (23)$$

Following [23], the effective power functional  $\mathcal{E}^{\text{eff}}(\dot{\boldsymbol{\varepsilon}})$  is thus defined by

$$\mathcal{E}^{\text{eff}}(\dot{\boldsymbol{\varepsilon}}) = \min_{\dot{\mathbf{q}}} \mathcal{E}(\dot{\boldsymbol{\varepsilon}}, \dot{\mathbf{q}}) \text{ with } \frac{\partial \mathcal{E}^{\text{eff}}(\dot{\boldsymbol{\varepsilon}})}{\partial \dot{\boldsymbol{\varepsilon}}} = \frac{\partial \Psi}{\partial \boldsymbol{\varepsilon}} = \boldsymbol{\sigma}. \quad (24)$$

Considering now an arbitrary time step  $[t_n, t_{n+1}]$ , the incremental form [23] of Eq. (23) reads

$$\Delta \mathcal{E}_{n+1}(\boldsymbol{\varepsilon}_{n+1}, \mathbf{q}_{n+1}; \boldsymbol{\varepsilon}_n, \mathbf{q}_n) = \int_{t_n}^{t_{n+1}} \mathcal{E} dt, \quad (25)$$

where  $\Delta \bullet_{n+1} = \bullet_{n+1} - \bullet_n$  denotes the increment of an arbitrary quantity  $\bullet$ , and  $\bullet_{n+1}$  and  $\bullet_n$  are respectively its values at configurations  $t_{n+1}$  and  $t_n$ . For a strain-driven constitutive model,  $\boldsymbol{\varepsilon}_{n+1}$  and  $\mathbf{q}_n$  are known and  $\mathbf{q}_{n+1}$  is obtained from the incremental form of Eq. (24), yielding

$$\mathbf{q}_{n+1} = \arg \min_{\mathbf{q}'_{n+1}} \Delta \mathcal{E}_{n+1} (\boldsymbol{\varepsilon}_{n+1}, \mathbf{q}'_{n+1}) \quad \text{and} \quad \Delta \mathcal{E}_{n+1}^{\text{eff}} (\boldsymbol{\varepsilon}_{n+1}) = \min_{\mathbf{q}_{n+1}} \Delta \mathcal{E}_{n+1} (\boldsymbol{\varepsilon}_{n+1}, \mathbf{q}_{n+1}), \quad (26)$$

where from now on the known terms at configuration  $n$  are omitted in the functional argument for conciseness. Eventually, the stress tensor arises from

$$\boldsymbol{\sigma}_{n+1} = \frac{\partial \Delta \mathcal{E}_{n+1}^{\text{eff}} (\boldsymbol{\varepsilon}_{n+1})}{\partial \boldsymbol{\varepsilon}_{n+1}}. \quad (27)$$

One can thus define the power functional over the domain volume  $V$  as

$$\dot{\Phi} (\dot{\mathbf{u}}, \dot{\mathbf{q}}) = \int_V \mathcal{E} (\boldsymbol{\varepsilon} (\dot{\mathbf{u}}), \dot{\mathbf{q}}) dV - \dot{W}^{\text{ext}} (\dot{\mathbf{u}}), \quad (28)$$

where  $\dot{W}^{\text{ext}}$  is the external power which is given as

$$\dot{W}^{\text{ext}} = \int_V \mathbf{b}_0 \cdot \dot{\mathbf{u}} dV + \int_{\partial_N V} \mathbf{t}_0 \cdot \dot{\mathbf{u}} dA. \quad (29)$$

Considering the time step  $[t_n, t_{n+1}]$  and assuming  $\mathbf{u}_n (\mathbf{x})$  and  $\mathbf{q}_n (\mathbf{x})$ ,  $\forall \mathbf{x} \in V$ , are known, Eq. (28) can be integrated over  $[t_n, t_{n+1}]$ , leading to

$$\Delta \Phi_{n+1} (\mathbf{u}_{n+1}, \mathbf{q}_{n+1}) = \int_{t_n}^{t_{n+1}} \dot{\Phi} dt = \int_V \Delta \mathcal{E}_{n+1} (\boldsymbol{\varepsilon} (\mathbf{u}_{n+1}), \mathbf{q}_{n+1}) dV - \Delta W_{n+1}^{\text{ext}} (\mathbf{u}_{n+1}). \quad (30)$$

For a given value of  $\mathbf{u}_{n+1}$ , the minimisation problems (26) can be performed locally to find  $\mathbf{q}_{n+1} (\mathbf{x})$ ,  $\forall \mathbf{x} \in V$ , leading to the overall optimisation problem

$$\mathbf{q}_{n+1} = \arg \min_{\mathbf{q}'_{n+1}} \Delta \Phi_{n+1} (\mathbf{u}_{n+1}, \mathbf{q}'_{n+1}). \quad (31)$$

Combining Eqs. (30, 31) leads to

$$\Delta \Phi_{n+1}^{\text{eff}} (\mathbf{u}_{n+1}) = \int_V \Delta \mathcal{E}_{n+1}^{\text{eff}} (\boldsymbol{\varepsilon} (\mathbf{u}_{n+1})) dV - \Delta W_{n+1}^{\text{ext}} (\mathbf{u}_{n+1}), \quad (32)$$

where  $\mathcal{E}_{n+1}^{\text{eff}}$  is estimated from the solution of the optimisation problem (31). The updated displacement field  $\mathbf{u}_{n+1}$  finally follows from the minimisation problem

$$\begin{cases} \mathbf{u}_{n+1} = \arg \min_{\mathbf{u}'_{n+1}} \Delta \Phi_{n+1}^{\text{eff}} (\mathbf{u}'_{n+1}), \\ \text{with } \mathbf{u}'_{n+1} \text{ kinematically admissible.} \end{cases} \quad (33)$$

These expressions will be particularised to the considered material models in Section 3.3.

The double-minimisation algorithm to be solved, *i.e.* the minimisation problem (31) and the minimisation problem (33), is illustrated in Alg. 1, in which the solution at time  $t_n$  is known and the problem is solved at time  $t_{n+1}$ .

---

**Algorithm 1** Double-Minimisation Algorithm.

---

**Require:**  $\mathbf{u}_n, \mathbf{q}_n$ ;

**Output:**  $\mathbf{u}_{n+1}, \mathbf{q}_{n+1}$ ;

- 1: initialise  $\mathbf{u}_{n+1} \leftarrow \mathbf{u}_n$  and  $\mathbf{q}_{n+1} \leftarrow \mathbf{q}_n$ ;
  - 2: apply boundary conditions;
  - 3: **repeat**
  - 4:      $\mathbf{u}_{n+1} \leftarrow$  solve the problem (33);
  - 5:      $\mathbf{q}_{n+1} \leftarrow$  solve the problem (31);
  - 6: **until** convergence
- 

### 3.3. Constitutive models

In this section, the variational formulation is detailed for two usual material models: linear elasticity and  $J_2$ -von Mises elasto-plasticity.

#### 3.3.1. Linear elasticity

In the case of a linear elastic material, one has  $\mathbf{q} = \emptyset$  and the free energy reads

$$\Psi = \frac{K}{2} [\text{tr}(\boldsymbol{\varepsilon})]^2 + \mu \text{dev}(\boldsymbol{\varepsilon}) : \text{dev}(\boldsymbol{\varepsilon}), \quad (34)$$

where  $K$  and  $\mu$  are the bulk and shear moduli, respectively. The stress tensor  $\boldsymbol{\sigma} = \frac{\partial \Psi}{\partial \boldsymbol{\varepsilon}}$

and the tangent operator  $\mathbf{C} = \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}}$  follow directly from Eq. (34). Since there is no internal variable, the minimisation problem (31) is trivial and the minimisation problem (33) is solved by considering directly  $\Phi(\mathbf{u})$  from Eq. (30), which is rewritten as

$$\Phi(\mathbf{u}) = \int_V \Psi(\mathbf{u}) dV - W^{\text{ext}}. \quad (35)$$

#### 3.3.2. $J_2$ -elasto-plasticity

In this model, the total strain tensor  $\boldsymbol{\varepsilon}$  is decomposed into a reversible elastic part  $\boldsymbol{\varepsilon}^e$  and an irreversible plastic part  $\boldsymbol{\varepsilon}^p$ , such that  $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^e + \boldsymbol{\varepsilon}^p$ . The collection of internal variables reads

$$\mathbf{q} = \{\boldsymbol{\varepsilon}^p, \gamma\}, \quad (36)$$

where  $\gamma$  is a scalar characterising the isotropic hardening evolution and which is related to the plastic strain tensor  $\boldsymbol{\varepsilon}^p$  through

$$\dot{\boldsymbol{\varepsilon}}^p = \dot{\gamma} \mathbf{N}, \quad (37)$$

where  $\dot{\gamma} \geq 0$  and  $\mathbf{N}$  is the plastic normal subjected to the constraints

$$\text{tr}(\mathbf{N}) = 0 \text{ and } \mathbf{N} : \mathbf{N} = \frac{3}{2}. \quad (38)$$

The free energy and dissipation pseudo-potential are chosen as follows

$$\Psi = \frac{K}{2} [\text{tr}(\boldsymbol{\varepsilon})]^2 + \mu [\text{dev}(\boldsymbol{\varepsilon}) - \boldsymbol{\varepsilon}^{\text{P}}] : [\text{dev}(\boldsymbol{\varepsilon}) - \boldsymbol{\varepsilon}^{\text{P}}], \text{ and} \quad (39)$$

$$\Theta = \begin{cases} [\sigma_y^0 + R(\gamma)] \dot{\gamma} & \text{if } \dot{\gamma} \geq 0, \\ \infty & \text{otherwise,} \end{cases} \quad (40)$$

where  $K$  and  $\mu$  are respectively the bulk and shear moduli,  $\sigma_y^0$  is the initial yield stress characterising the onset of plasticity and  $R(\gamma)$  is the hardening law introduced to capture the isotropic hardening effect. The corresponding dual potential is

$$\Theta^* = \begin{cases} 0 & \text{if } \sigma_{\text{eq}} - \sigma_y^0 - R \leq 0, \\ \infty & \text{otherwise.} \end{cases}, \quad (41)$$

where  $\sigma_{\text{eq}}$  is the equivalent von Mises stress of the stress tensor  $\boldsymbol{\sigma}$ .<sup>4</sup>

Whilst the stress tensor  $\boldsymbol{\sigma}$  follows from Eq. (18) or Eq. (27), this requires to know the internal state of the system defined by  $\{\boldsymbol{\varepsilon}^{\text{P}}, \gamma\}$ . Considering a time step  $[t_n, t_{n+1}]$ , all quantities at time  $t_n$  are known. With the introduction of the plastic flow direction  $\mathbf{N}$  (37), one has

$$\Delta \boldsymbol{\varepsilon}_{n+1}^{\text{P}} = \boldsymbol{\varepsilon}_{n+1}^{\text{P}} - \boldsymbol{\varepsilon}_n^{\text{P}} = \Delta \gamma_{n+1} \mathbf{N}_{n+1}, \quad (44)$$

and the internal state of the model can thus be defined by

$$\mathbf{q}_{n+1} = \{\boldsymbol{\varepsilon}_{n+1}^{\text{P}}, \gamma_{n+1}\} \equiv \{\gamma_{n+1}, \mathbf{N}_{n+1}\}. \quad (45)$$

---

<sup>4</sup>There are other possibilities to define the free energy and dissipation pseudo-potential, see *e.g.* the works [23, 46, 47] in which the free energy consists of the reversible and irreversible parts. As an example, in [23], the free energy functional was defined by

$$\Psi = \frac{K}{2} [\text{tr}(\boldsymbol{\varepsilon})]^2 + \mu [\text{dev}(\boldsymbol{\varepsilon}) - \boldsymbol{\varepsilon}^{\text{P}}] : [\text{dev}(\boldsymbol{\varepsilon}) - \boldsymbol{\varepsilon}^{\text{P}}] + \int_0^\gamma [\sigma_y^0 + R(\gamma')] d\gamma', \quad (42)$$

and the dissipation pseudo-potential was defined as an indicator specifying the admissible domain as

$$\Theta = \begin{cases} 0 & \text{if } \dot{\gamma} \geq 0, \\ \infty & \text{otherwise.} \end{cases} \quad (43)$$

Although different definitions can be employed, the same functional to be minimised is eventually obtained.

The energy increment of the energy functional (25) thus reads

$$\begin{aligned} \Delta \mathcal{E}_{n+1}(\mathbf{u}_{n+1}, \Delta \gamma_{n+1}, \mathbf{N}_{n+1}) = & \\ \frac{K}{2} [\text{tr}(\boldsymbol{\varepsilon}_{n+1})]^2 + \mu [\text{dev}(\boldsymbol{\varepsilon}_{n+1}) - \boldsymbol{\varepsilon}_n^p - \Delta \gamma_{n+1} \mathbf{N}_{n+1}] : [\text{dev}(\boldsymbol{\varepsilon}_{n+1}) - \boldsymbol{\varepsilon}_n^p - \Delta \gamma_{n+1} \mathbf{N}_{n+1}] & \\ - \frac{K}{2} [\text{tr}(\boldsymbol{\varepsilon}_n)]^2 + \mu [\text{dev}(\boldsymbol{\varepsilon}_n) - \boldsymbol{\varepsilon}_n^p] : [\text{dev}(\boldsymbol{\varepsilon}_n) - \boldsymbol{\varepsilon}_n^p] + \int_{\gamma_n}^{\gamma_n + \Delta \gamma_{n+1}} [\sigma_y^0 + R(\gamma')] d\gamma'. & \end{aligned} \quad (46)$$

Therefore, at a material point  $\mathbf{x}$ , the internal variables arise from the minimisation of the convex energy function (46) under the constraints (38), yielding

$$\{\Delta \gamma_{n+1}(\mathbf{x}), \mathbf{N}_{n+1}(\mathbf{x})\} = \arg \min_{\{\Delta \gamma'_{n+1}, \mathbf{N}'_{n+1}\}} \Delta \mathcal{E}_{n+1}(\mathbf{u}_{n+1}, \Delta \gamma'_{n+1}, \mathbf{N}'_{n+1}), \quad (47)$$

$$\text{satisfying } \Delta \gamma_{n+1} \geq 0, \text{tr}(\mathbf{N}_{n+1}) = 0, \text{ and } \mathbf{N}_{n+1} : \mathbf{N}_{n+1} = \frac{3}{2}. \quad (48)$$

The constrained minimisation problem (47, 48) yields the well-known radial return scheme in the conventional  $J_2$ -plasticity model [47, 49]. Nevertheless, within the variational updates formulation, the values of  $\mathbf{N}_{n+1}$  and of  $\Delta \gamma_{n+1}$  are obtained from the minimisation process, and not following some model assumptions.

At the structural domain level, the functional defining the minimisation problem yielding  $\mathbf{u}_{n+1}$  as well as  $\Delta \gamma_{n+1}(\mathbf{x})$  and  $\mathbf{N}_{n+1}(\mathbf{x})$ ,  $\forall \mathbf{x} \in V$ , is obtained by combining Eqs. (30, 33, 47, 48), resulting into

$$\Delta \Phi_{n+1} = \Delta \Phi_{n+1}(\mathbf{u}_{n+1}, \Delta \gamma_{n+1}, \mathbf{N}_{n+1}), \quad (49)$$

with the constraints (48) having to be applied at every material point  $\mathbf{x} \in V$ .

### 3.4. Finite element approximation

In a classical finite element approach, the resolution of the material law, *e.g.* the minimisation of Eq. (46), is conducted at the local level, *i.e.* at the quadrature points, from an estimation of the displacement field. The latter is corrected considering the resolution of the balance equation at the finite element mesh level. Nested Newton-Raphson loops are thus required at both local and global levels. In this section we summarise the finite element approach in which the resolution of the constitutive law (46) is conducted at the finite element level.

The functional (30) is estimated using the finite element approximation. It is necessary to discretise not only the displacement field  $\mathbf{u}$  but also the internal variables field  $\mathbf{q}$ . The displacement field degrees of freedom are defined at the finite element nodes whilst the internal variable degrees of freedom are defined at the quadrature points in order to avoid their interpolation, which could be inconsistent, respectively defining the vectors of degrees of freedom  $\mathbf{U}$  and  $\mathbf{Q}$ . More details on the finite element discretisation are given in [Appendix A](#).

The energy functional (30) is therefore estimated from the fields evaluated at the quadrature points and from the variation of incremental functional  $\Delta \mathcal{E}_{n+1}$  evaluated following Sec-

tion 3.3, leading to

$$\Delta\Phi_{n+1} = \Delta W_{n+1}^{\text{int}} - \Delta W_{n+1}^{\text{ext}}, \quad (50)$$

where

$$\Delta W_{n+1}^{\text{int}} = \sum_{e=0}^{N_{\text{ele}}-1} \sum_{\eta=0}^{m_e-1} \omega^{e,\eta} \Delta\mathcal{E}_{n+1}(\mathbf{B}^{e,\eta}\mathbf{U}_{n+1}^e, \mathbf{q}_{n+1}^{e,\eta}), \quad \text{and} \quad (51)$$

$$\Delta W_{n+1}^{\text{ext}} = \Delta\mathbf{U}_{n+1}^T \mathbf{f}_{n+1}^{\text{ext}}, \quad (52)$$

where  $N_{\text{ele}}$  is the number of finite elements,  $m_e$  is the number of quadrature points in element  $e$ ,  $\mathbf{B}^{e,\eta}$  is the elementary matrix of the gradient of the shape functions associated with the elementary displacement vector field  $\mathbf{U}^e$  evaluated at the quadrature point  $\eta$  of element  $e$  –i.e.  $\Xi^{e,\eta}$ ,  $\mathbf{q}^{e,\eta}$  is the internal variables defined at the quadrature point  $\Xi^{e,\eta}$ ,  $\omega^{e,\eta}$  denotes the weights of the quadrature points  $\Xi^{e,\eta}$ , and where  $\mathbf{f}_{n+1}^{\text{ext}}$  is the external force vector.

The double-minimisation problem detailed in Alg. 1 is reformulated as the variational formulation of the finite element method that consists in evaluating the minimum state of the constrained energy functional (50), which is convex for the models presented in Section 3.3, following

$$\begin{cases} \mathbf{Q}_{n+1} & = \arg \min_{\mathbf{Q}'_{n+1}} \Delta\Phi_{n+1}(\mathbf{U}_{n+1}, \mathbf{Q}'_{n+1}) \quad \text{under constraints on } \mathbf{Q}'_{n+1}; \\ \Delta\Phi_{n+1}^{\text{eff}}(\mathbf{U}_{n+1}) & = \min_{\mathbf{Q}'_{n+1}} \Delta\Phi_{n+1}(\mathbf{U}_{n+1}, \mathbf{Q}'_{n+1}) \quad \text{under constraints on } \mathbf{Q}'_{n+1}; \\ \mathbf{U}_{n+1} & = \arg \min_{\mathbf{U}'_{n+1}} \Delta\Phi_{n+1}^{\text{eff}}(\mathbf{U}'_{n+1}) \quad \text{with } \mathbf{U}'_{n+1} \text{ kinematically admissible.} \end{cases} \quad (53)$$

The double-minimisation process (53) could be solved with a classical hardware using available optimisation methods. In particular, since the functional arising from the variational formulation is convex, conic programming [50] was shown to be well suited to conduct finite element analyses since both the free energy and dissipation pseudo-potential can be formulated to be conic representable [51]. However, conducting the optimisation on a classical computer involves creating algorithms that execute computer instructions sequentially since classical computers process information using bits, which can only be either 0 or 1. On the other hand, quantum annealing leverages principles of quantum mechanics to find solutions to optimisation problems at once by exploiting quantum superposition and entanglement to explore multiple possible solutions simultaneously. This could make the quantum annealing potentially more efficient than classical optimisers in classical computers. This is particularly true since the minimisation problem in terms of the internal variables is simply the combination of independent constrained minimisation problems that do not require all the qubits to be connected and can therefore exploit the parallelism advantage of quantum computers. Nevertheless not all functions can be encoded into the current quantum annealing hardware since only minimisation of quadratic potentials under the form of either Eq. (10) or Eq. (11) can be conducted, which motivates the development of a hybrid resolution of the double-minimisation problem.

#### 4. Hybrid resolution of the double-minimisation problem

This section presents a hybrid strategy, *i.e.* using classical computers and the quantum annealing approach described in Section 2, to solve the double-minimisation problem arising from the variational formulation described in Section 3. For this purpose, we suggest a quantum annealing-assisted sequential quadratic programming procedure to minimise a multivariate convex function with bound constraints. This procedure can then be used for the resolution of the double-minimisation described in Alg. 1, whenever such a minimisation is required, *i.e.* at lines 4 and 5.

##### 4.1. Quantum annealing-assisted sequential quadratic programming (QA-SQP)

Sequential Quadratic Programming (SQP) is a numerical optimisation method used to solve non-linear optimisation problems. It is an iterative algorithm that aims at finding the solution of a multivariate optimisation problem by approximating it as a series of quadratic problems.

Let us consider a multivariate minimisation problem of the  $N$ -variable objective function  $f$ :

$$\begin{cases} \min_{\mathbf{w}} f(\mathbf{w}) , \\ \text{subject to } \mathbf{w}^{\min} \leq \mathbf{w} \leq \mathbf{w}^{\max} . \end{cases} \quad (54)$$

For the cases involving the following equality and/or inequality constraints

$$\begin{cases} h_j(\mathbf{w}) = 0 \text{ with } j = 0, \dots, N_h - 1, \text{ and} \\ l_j(\mathbf{w}) \leq 0 \text{ with } j = 0, \dots, N_l - 1, \end{cases} \quad (55)$$

where  $N_h$  and  $N_l$  are the number of equality and inequality constraints, respectively, the corresponding constrained minimisation problem needs to be transformed into the quasi-unconstrained form (54). For this purpose, we define the augmented objective function using penalties as

$$f_{\text{aug}}(\mathbf{v}) = f_{\text{aug}}(\mathbf{w}, \boldsymbol{\lambda}) = f(\mathbf{w}) + \sum_{j=0}^{N_h-1} c_j^h [h_j(\mathbf{w})]^2 + \sum_{j=0}^{N_l-1} c_j^l [l_j(\mathbf{w}) + \lambda_j]^2, \quad (56)$$

where  $c_j^h$  and  $c_j^l$ ,  $\forall j$  are the penalty factors associated with the constraints  $h_j$  and  $l_j$ , respectively,  $\boldsymbol{\lambda} = [\lambda_j, \forall j = 0, \dots, N_l - 1]$  is the vector of the so-called auxiliary variables, and where  $\mathbf{v} = [\mathbf{w}^T \boldsymbol{\lambda}^T]^T$  gathers all the unknowns. The quasi-unconstrained form (54) is then rewritten as

$$\min_{\mathbf{w}, \boldsymbol{\lambda}} f_{\text{aug}}(\mathbf{w}, \boldsymbol{\lambda}) \text{ subject to } \mathbf{w}^{\min} \leq \mathbf{w} \leq \mathbf{w}^{\max} \text{ and } \lambda_j \geq 0 \forall j = 0, \dots, N_l - 1, \quad (57)$$

or in the condensed form

$$\min_{\mathbf{v}} f_{\text{aug}}(\mathbf{v}) \text{ subject to } \mathbf{v}^{\min} \leq \mathbf{v} \leq \mathbf{v}^{\max}, \quad (58)$$

in which the lower and upper bounds associated to the components of  $\mathbf{v}$  corresponding to  $\lambda_j$  are respectively 0 and  $\infty$ .

For a commonly encountered situation in computational mechanics in which the gradient  $\mathbf{g}$  and the non-negative definite hessian  $\mathbf{K}$  of  $f_{\text{aug}}$  exist everywhere, one has the following approximation

$$f_{\text{aug}}(\mathbf{v} + \mathbf{z}) \approx f_{\text{aug}}(\mathbf{v}) + \text{QF}(\mathbf{z}; \mathbf{g}, \mathbf{K}), \quad (59)$$

where the components of  $\mathbf{g}$  and of  $\mathbf{K}$  read

$$g_i = \left. \frac{\partial f_{\text{aug}}}{\partial v_i} \right|_{\mathbf{v}} \quad \text{and} \quad K_{ij} = \left. \frac{\partial^2 f_{\text{aug}}}{\partial v_i \partial v_j} \right|_{\mathbf{v}}, \quad (60)$$

and where  $\text{QF}(\mathbf{z}; \mathbf{g}, \mathbf{K})$  denotes a general quadratic form in the vector  $\mathbf{z}$  defining the problem unknowns. This quadratic form is defined by the vector  $\mathbf{g}$  and the matrix  $\mathbf{K}$  following

$$\text{QF}(\mathbf{z}; \mathbf{g}, \mathbf{K}) = \mathbf{z}^T \mathbf{g} + \frac{1}{2} \mathbf{z}^T \mathbf{K} \mathbf{z}. \quad (61)$$

The key steps in the SQP algorithm to solve the minimisation problem stated by Eq. (58) are:

- (i) Start with an initial admissible solution;
- (ii) Construct a quadratic model (61) of the objective function  $f$  and the constraints around the current solution;
- (iii) Solve the quadratic minimisation problem  $\mathbf{z}^* = \arg \min_{\mathbf{z}} \text{QF}(\mathbf{z}; \mathbf{g}, \mathbf{K})$  with respect to the constraints of Eq. (58);
- (iv) Check a convergence criterion: if the convergence is reached terminate the algorithm; otherwise, go back to step (ii).

In this work, we suggest to use the quantum annealing for step (iii), leading to the so-called Quantum Annealing-assisted Sequential Quadratic Programming (QA-SQP). For this purpose, the transformation from the continuous quadratic form (61) into the binary quadratic form (11) is first introduced. Then a hybrid nested scheme to solve the minimisation problem (58) is provided.

#### 4.1.1. Conversion from QF to QUBO

A  $L$ -bit string, *i.e.*  $b_{L-1} \dots b_0$  with  $b_i \in \{0, 1\}$ , can represent an arbitrary whole number ranging from 0 to  $2^L - 1$  through the binary-decimal conversion

$$b_{L-1} \dots b_0 \equiv \sum_{j=0}^{L-1} b_j 2^j. \quad (62)$$

As a result, one can discretise the range  $[z^{\min}, z^{\max}]$  into  $2^L$  discrete values following

$$z(b_0, \dots, b_{L-1}) = \bar{z} + \epsilon \left( \sum_{j=0}^{L-1} b_j 2^j - 2^{L-1} + 1 \right) \in [z^{\min}, z^{\max}], \quad (63)$$

where  $\epsilon$  is the so-called discretisation error, and where

$$\begin{cases} z^{\min} &= \bar{z} - (2^{L-1} - 1) \epsilon, \text{ and} \\ z^{\max} &= \bar{z} + 2^{L-1} \epsilon. \end{cases} \quad (64)$$

Therefore, Eq. (63) can be rewritten as

$$z(\mathbf{b}) = z^{\min} + \epsilon \boldsymbol{\beta}^T \mathbf{b}, \quad (65)$$

where  $\boldsymbol{\beta} = [2^0 \ 2^1 \ \dots \ 2^{L-1}]^T$  and  $\mathbf{b} = [b_0 \ \dots \ b_{L-1}]^T$ . For a given error  $\epsilon$ , a larger range  $[z^{\min}, z^{\max}]$  is obtained using more qubits whilst when the number of qubits is fixed, the range  $[z^{\min}, z^{\max}]$  becomes smaller with decreasing  $\epsilon$ .

Let us consider a vector  $\mathbf{z}$  of  $N$  components, Eq. (65) is thus applied to each component of  $\mathbf{z}$ , leading to

$$\mathbf{z} = \mathbf{z}^{\min} + [\epsilon_i \boldsymbol{\beta}^T \mathbf{b}_i \text{ for } i = 0, \dots, N-1] = \mathbf{a} + \mathbf{D} \mathbf{b} \in [\mathbf{z}^{\min}, \mathbf{z}^{\max}], \quad (66)$$

where  $\bullet_i$  denotes a quantity associated with the component  $z_i$ ,  $\mathbf{z}^{\min}$  and  $\mathbf{z}^{\max}$  denote respectively the lower-bound and upper-bound of  $\mathbf{z}$  and where

$$\mathbf{a} = \mathbf{z}^{\min} = [z_0^{\min} \ \dots \ z_{N-1}^{\min}]^T, \quad (67)$$

$$\mathbf{D} = \text{diag}(\epsilon_0 \boldsymbol{\beta}^T, \dots, \epsilon_{N-1} \boldsymbol{\beta}^T), \text{ and} \quad (68)$$

$$\mathbf{b} = [\mathbf{b}_0^T \ \dots \ \mathbf{b}_{N-1}^T]^T, \quad (69)$$

with  $\text{diag}(\bullet)$  denoting the  $N \times (N \times L)$  block diagonal matrix.

We now reconsider the multivariate optimisation problem in terms of the objective function (59). Using Eq. (66), Eq. (61) becomes

$$\text{QF}(\mathbf{z}; \mathbf{g}, \mathbf{K}) = \frac{1}{2} \mathbf{b}^T \mathbf{D}^T \mathbf{K} \mathbf{D} \mathbf{b} + \mathbf{b}^T \mathbf{D}^T (\mathbf{K} \mathbf{a} + \mathbf{g}) + \frac{1}{2} \mathbf{a}^T \mathbf{K} \mathbf{a} + \mathbf{a}^T \mathbf{g}. \quad (70)$$

Since  $b_i \in \{0, 1\}$ , one has  $b_i^2 = b_i$ ,  $\forall i$ . As a result, Eq. (70) can be rewritten in terms of a QUBO function (11) as

$$\text{QF}(\mathbf{z}; \mathbf{g}, \mathbf{K}) = \text{QUBO}(\mathbf{b}; \mathbf{A}) + \frac{1}{2} \mathbf{a}^T \mathbf{K} \mathbf{a} + \mathbf{a}^T \mathbf{g}, \quad (71)$$

where

$$\mathbf{A} = \frac{1}{2} \mathbf{D}^T \mathbf{K} \mathbf{D} + \text{diag}(\mathbf{D}^T (\mathbf{K} \mathbf{a} + \mathbf{g})), \quad (72)$$

with  $\text{diag}(\bullet)$  denoting the diagonal matrix whose diagonal is given by a vector  $\bullet$ .

Contrarily to the quadratic form (61), which is expressed in terms of continuous variables, the binary quadratic form (71) can be minimised by quantum annealing and the continuous variables are recovered using Eq. (66). The difficulty is now to define the parameters  $\mathbf{a}$  and  $\mathbf{D}$  required for the binary encoding (66). Toward this end, an iterative procedure is introduced in the hybrid nested scheme.

#### 4.1.2. Hybrid nested scheme

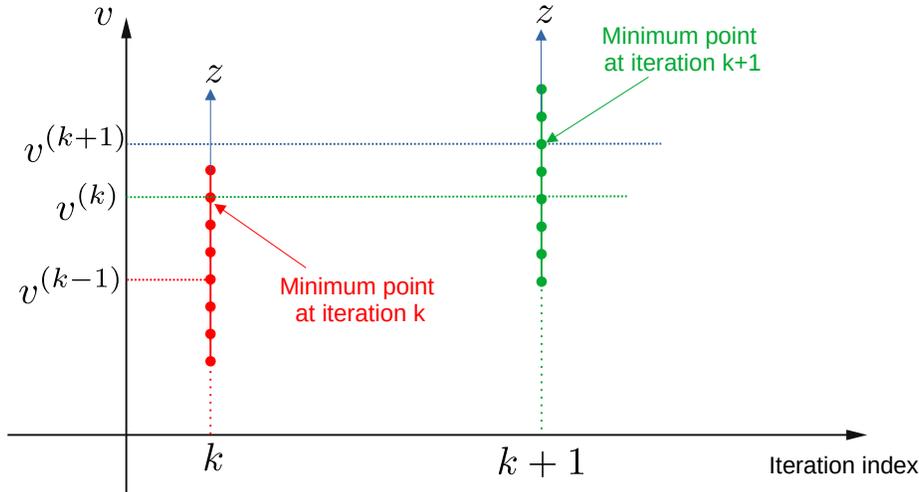


Figure 1: Illustration of the minimisation of a single variable  $v$  function considering  $L = 3$  qubits. At each iteration, the minimum between 8 discrete values is found and is used as the starting point for the next iteration.

The basic idea of the iterative procedure is illustrated in Fig. 1 when performing the minimisation of a function of a single variable  $v$  with  $L = 3$  qubits. At an iteration  $k$ , the solution  $v^{(k-1)}$  and the discretisation error  $\epsilon^{(k-1)}$  are known. Using  $L$  qubits, the discretisation error  $\epsilon^{(k)} = \epsilon^{(k-1)}$ , and  $\bar{z}^{(k)} = 0$ ,  $2^L$  discrete values are defined around  $v^{(k-1)}$ . As a result a QUBO problem is defined as shown by Eq. (71) and whose minimisation allows obtaining  $v^{(k)}$  for the next iteration. To this end, two situations need to be considered:

- At iteration  $k$ , the upper-bound and lower-bound of  $v$  have to be respected, *i.e.*

$$v^{\min} \leq v^{(k-1)} + z \leq v^{\max}. \quad (73)$$

For this purpose, the discretisation around the solution of the previous iteration needs to be adjusted as follows

$$z^{\min} = \max(v^{\min} - v^{(k-1)}, -(2^{L-1} - 1)\epsilon^{(k-1)}), \quad (74)$$

$$z^{\max} = \min(v^{\max} - v^{(k-1)}, 2^{L-1}\epsilon^{(k-1)}), \quad (75)$$

$$\epsilon^{(k)} = \frac{z^{\max} - z^{\min}}{2^L - 1}, \text{ and} \quad (76)$$

$$\bar{z}^{(k)} = \frac{z^{\max} + z^{\min} - \epsilon^{(k)}}{2}. \quad (77)$$

Finally, the new discretisation is built around  $\bar{z}^{(k)}$  following Eq. (77) and using the discretisation error  $\epsilon^{(k)}$  following Eq. (76).

- The objective function value does not decrease. In this situation, the discretisation error is decreased by a factor  $\xi < 1$ , *i.e.*  $\epsilon^{(k)} = \epsilon^{(k)}\xi$ . In this work, we use  $\xi = 0.5$ .

---

**Algorithm 2** QA-SQP for solving the minimisation problem stated by Eqs. (58).

---

**Require:**  $\mathbf{v}_0$  (initial solution),  $L$  (number of qubits),  $\mathbf{z}^{\min}$  (lower bound),  $\mathbf{z}^{\max}$  (upper bound),  $\epsilon_0$  (initial discretisation error),  $\xi$  (shrinking factor),  $N_{\text{steps}}$  (maximum number of successful iterations),  $N_{\text{failed}}$  (maximum number of failed iterations)

**Output:**  $\mathbf{v}$  (optimised solution)

- 1: initialise  $\mathbf{v} \leftarrow \mathbf{v}_0$ ,  $k \leftarrow 0$ ,  $l \leftarrow 0$ ;
  - 2: estimate  $f_{\text{aug}}$  following Eq. (56) and  $\mathbf{g}$ ,  $\mathbf{K}$  following Eq. (60) at  $\mathbf{v}$ ;
  - 3: initialise  $\epsilon \leftarrow \epsilon_0$ ;
  - 4: **repeat**
  - 5:   update  $\epsilon$  and  $\bar{\mathbf{z}}$  from  $L$ ,  $\mathbf{v}$ ,  $\mathbf{z}^{\min}$ , and  $\mathbf{z}^{\max}$  following Eq. (82);
  - 6:   estimate  $\mathbf{a}$  and  $\mathbf{D}$  from  $L$ ,  $\bar{\mathbf{z}}$  and  $\epsilon$  following Eqs. (67, 68);
  - 7:   estimate the QUBO matrix  $\mathbf{A}$  following Eq. (72);
  - 8:   call quantum annealer (11), leading to the solution  $\mathbf{b}$ ;
  - 9:   estimate  $\mathbf{z} = \mathbf{a} + \mathbf{D}\mathbf{b}$ ;
  - 10:   estimate  $f_{\text{aug}_{\text{new}}} = f_{\text{aug}}(\mathbf{v} + \mathbf{z})$  following Eq. (56);
  - 11:   **if**  $f_{\text{aug}_{\text{new}}} < f_{\text{aug}}$  **then**
  - 12:      $k \leftarrow k + 1$ ;
  - 13:     estimate  $\mathbf{g}_{\text{new}}$ ,  $\mathbf{K}_{\text{new}}$  at  $\mathbf{v} + \mathbf{z}$  following Eq. (60);
  - 14:      $\mathbf{v} \leftarrow \mathbf{v} + \mathbf{z}$ ,  $f_{\text{aug}} \leftarrow f_{\text{aug}_{\text{new}}}$ ,  $\mathbf{g} \leftarrow \mathbf{g}_{\text{new}}$ , and  $\mathbf{K} \leftarrow \mathbf{K}_{\text{new}}$ ;
  - 15:   **else**
  - 16:      $l \leftarrow l + 1$ ;
  - 17:      $\epsilon \leftarrow \max(\epsilon^{\min}, \xi\epsilon)$ ;
  - 18:   **end if**
  - 19:   **if**  $k \geq N_{\text{steps}}$  or  $l \geq N_{\text{failed}}$  **then**
  - 20:     break;
  - 21:   **end if**
  - 22: **until** convergence
- 

The minimisation problem (59) of the multi-variable function is directly obtained by generalisation of the single variable case above and is detailed in Alg. 2. The solution is initialised with a feasible initial guess  $\mathbf{v}_0$ , *i.e.*  $\mathbf{v}^{\min} \leq \mathbf{v}_0 \leq \mathbf{v}^{\max}$ . This allows initialising the values of  $\mathbf{g}$  and  $\mathbf{K}$  in line (2). In line (5), the pair  $(\epsilon, \bar{\mathbf{z}})$  defines a range for the solution of the quadratic form (61). The initial discretisation error  $\epsilon_0$  can be chosen through an initial solution interval  $\Delta\mathbf{v}_0$  such that

$$\epsilon_0 = \frac{\Delta\mathbf{v}_0}{2^L - 1}. \quad (78)$$

At each iteration with the current solution  $\mathbf{v}$ , the value of  $\mathbf{z}$  will be found such that

$$\mathbf{v}^{\min} - \mathbf{v} \leq \mathbf{z} \leq \mathbf{v}^{\max} - \mathbf{v}. \quad (79)$$

Besides, for the current  $\epsilon$  and  $\bar{\mathbf{z}} = \mathbf{0}$ , we have the range of the solution

$$\mathbf{v} - (2^{L-1} - 1) \epsilon \leq \mathbf{v} + \mathbf{z} \leq \mathbf{v} + 2^{L-1} \epsilon, \quad (80)$$

leading to new bounds on  $\mathbf{z}$ ,

$$\begin{cases} \mathbf{z}^{\min} &= \max(\mathbf{v}^{\min} - \mathbf{v}, -(2^{L-1} - 1) \epsilon), \text{ and} \\ \mathbf{z}^{\max} &= \min(\mathbf{v}^{\max} - \mathbf{v}, 2^{L-1} \epsilon). \end{cases} \quad (81)$$

As a result, the updated values of the pair  $(\epsilon, \bar{\mathbf{z}})$  read

$$\begin{cases} \epsilon &= \frac{\mathbf{z}^{\max} - \mathbf{z}^{\min}}{2^L - 1}, \text{ and} \\ \bar{\mathbf{z}} &= \frac{\mathbf{z}^{\min} + \mathbf{z}^{\max} - \epsilon}{2}. \end{cases} \quad (82)$$

#### 4.2. Application of the QA-SQP to the finite element double-minimisation

In this section, the QA-SQP algorithm proposed in the previous section is particularised to solve the double-minimisation problem detailed in Alg. 1 arising from the variational formulation of the finite element method.

---

**Algorithm 3** Finite element simulation with Double-Minimisation Algorithm.

---

**Require:**  $\mathbf{U}_n, \mathbf{Q}_n$ ;

**Output:**  $\mathbf{U}_{n+1}, \mathbf{Q}_{n+1}$ ;

1: initialise  $\mathbf{U}_{n+1} \leftarrow \mathbf{U}_n$  and  $\mathbf{Q}_{n+1} \leftarrow \mathbf{Q}_n$ ;

2: apply boundary conditions;

3: **repeat**

4:    $\mathbf{U}_{n+1} \leftarrow$  minimise the functional (50) with the last value of  $\mathbf{Q}_{n+1}$  using QA-SQP, Alg. 2;

5:    $\mathbf{Q}_{n+1} \leftarrow$  minimise the functional (50) with the last value of  $\mathbf{U}_{n+1}$  using QA-SQP, Alg. 2;

6: **until** convergence

---

Following Section 3.4 and the set of Eqs. (53), the finite element simulation version of Alg. 1 follows as Alg. 3. As a reminder, the vectors of degrees of freedom  $\mathbf{U}$  and  $\mathbf{Q}$  arise from the discretisation of the displacement field  $\mathbf{u}$  and of the internal variables field  $\mathbf{q}$ , respectively.

##### 4.2.1. Constitutive model quasi-constrained minimisation

In the case of  $J_2$ -plasticity, the optimisation problem stated in line (5) in Alg. 3 corresponds to the constrained minimisation problem stated by Eqs. (47, 48). To apply the framework developed in Section 4.1, this constrained minimisation problem needs to be

transformed into a quasi-constrained minimisation of the form (56-58). For this purpose, we minimise the augmented incremental functional defined from Eq. (46) as

$$\Delta\gamma_{n+1}, \boldsymbol{\alpha} = \arg \min_{\Delta\gamma_{n+1}, \boldsymbol{\alpha}} \left[ \Delta\mathcal{E}_{n+1}(\mathbf{u}_{n+1}, \Delta\gamma_{n+1}, \mathbf{N}_{n+1}) + c_0^h \left( \boldsymbol{\alpha}^T \mathbf{M} \boldsymbol{\alpha} - \frac{3}{2} \right)^2 \right], \quad (83)$$

subject to  $\Delta\gamma_{n+1} \geq 0$ , and  $-\sqrt{\frac{3}{2}} \leq \alpha_j \leq \sqrt{\frac{3}{2}} \quad \forall \alpha_j \in \boldsymbol{\alpha}$ ,

where  $\mathbf{N}_{n+1}$  is parameterised as a purely deviatoric tensor such that  $\mathbf{N}_{n+1} : \mathbf{N}_{n+1} = \boldsymbol{\alpha}^T \mathbf{M} \boldsymbol{\alpha}$  with<sup>5</sup>

$$\mathbf{N}_{n+1} = \begin{bmatrix} \alpha_0 & \alpha_2/\sqrt{2} & \alpha_3/\sqrt{2} \\ \alpha_2/\sqrt{2} & \alpha_1 & \alpha_4/\sqrt{2} \\ \alpha_3/\sqrt{2} & \alpha_4/\sqrt{2} & -\alpha_0 - \alpha_1 \end{bmatrix}, \boldsymbol{\alpha} = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix}, \text{ and } \mathbf{M} = \begin{bmatrix} 2 & 1 & 0 & 0 & 0 \\ 1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (84)$$

The penalty parameter  $c_0^h$  is choose to be equal to the shear modulus  $\mu$  in Eq. (83).

In practice, each minimisation problem of Alg. 3 is performed using the QA-SQP presented in Alg. 2. Actually, Eq. (83) is a fourth-order polynomial in terms of  $\boldsymbol{\alpha}$ . When using the binarisation (66), the higher third- and fourth-order binary terms arise, leading to a higher-order unconstrained binary optimisation (HUBO) problem in terms of  $\boldsymbol{\alpha}$ . Such a HUBO can be transformed into a QUBO by replacing the product of two binary variables by a new one and by including an appropriate penalty term to enforce this equality [52–55]. In that case the constraints are naturally quadratic and the quadratic approximation (59) is no longer required to handle the constraints part of the objective function (56). However, in this work, since we have to consider this quadratic approximation for the other terms of the objective function anyway, we apply the developed SQP strategy for all the terms of the augmented incremental functional defined in Eq. (83).

#### 4.2.2. Finite element assembly

In order to make the finite element optimisation formulation ready for QA-SQP, the quadratic approximation of the function (50) needs to be constructed.

For a given  $\Delta\mathbf{U}_{n+1}$  and a given  $\Delta\mathbf{Q}_{n+1}$ , the quadratic approximation of Eq. (50) with respect to the perturbation of  $\Delta\mathbf{Q}_{n+1}$ , *i.e.*  $\mathbf{z}_Q$ , reads

$$\Delta\Phi_{n+1}(\Delta\mathbf{U}_{n+1}, \Delta\mathbf{Q}_{n+1} + \mathbf{z}_Q) \approx \Delta\Phi_{n+1}(\Delta\mathbf{U}_{n+1}, \Delta\mathbf{Q}_{n+1}) + \mathbf{z}_Q^T \mathbf{r}_{n+1} + \frac{1}{2} \mathbf{z}_Q^T \mathbf{T}_{n+1} \mathbf{z}_Q, \quad (85)$$

---

<sup>5</sup>In a two-dimensional setting, one has  $\alpha_3 = \alpha_4 = 0$ , leading to

$$\mathbf{N}_{n+1} = \begin{bmatrix} \alpha_0 & \alpha_2/\sqrt{2} & 0 \\ \alpha_2/\sqrt{2} & \alpha_1 & 0 \\ 0 & 0 & -\alpha_0 - \alpha_1 \end{bmatrix}, \boldsymbol{\alpha} = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{bmatrix}, \text{ and } \mathbf{M} = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

where

$$\mathbf{r}_{n+1} = \frac{\partial \Delta \Phi_{n+1}}{\partial \Delta \mathbf{Q}_{n+1}} = - \bigwedge_{V^e \subset V} \sum_{\eta=0}^{m_e-1} \omega^{e,\eta} \mathbf{Y}_{n+1}, \text{ and} \quad (86)$$

$$\mathbf{T}_{n+1} = \frac{\partial \Delta \mathbf{r}_{n+1}}{\partial \Delta \mathbf{Q}_{n+1}} = - \bigwedge_{V^e \subset V} \sum_{\eta=0}^{m_e-1} \omega^{e,\eta} \mathbf{Z}_{n+1}, \quad (87)$$

with  $\bigwedge$  representing the assembly process of the elements  $V^e$  of the volume  $V$ , and with  $\mathbf{Y}_{n+1}$  and  $\mathbf{Z}_{n+1}$  being available from  $\Delta \mathcal{E}_{n+1}$  as

$$\mathbf{Y}_{n+1} = - \frac{\partial \Delta \mathcal{E}_{n+1}}{\partial \Delta \mathbf{q}_{n+1}} \text{ and } \mathbf{Z}_{n+1} = \frac{\partial \mathbf{Y}_{n+1}}{\partial \Delta \mathbf{q}_{n+1}}. \quad (88)$$

For a given  $\Delta \mathbf{U}_{n+1}$  and a given  $\Delta \mathbf{Q}_{n+1}$ , the quadratic approximation of Eq. (50) with respect to the perturbation of  $\Delta \mathbf{U}_{n+1}$ , *i.e.*  $\mathbf{z}_U$ , reads

$$\Delta \Phi_{n+1}(\Delta \mathbf{U}_{n+1} + \mathbf{z}_U, \Delta \mathbf{Q}_{n+1}) \approx \Delta \Phi_{n+1}(\Delta \mathbf{U}_{n+1}, \Delta \mathbf{Q}_{n+1}) + \mathbf{z}_U^T \mathbf{g}_{n+1} + \frac{1}{2} \mathbf{z}_U^T \mathbf{K}_{n+1} \mathbf{z}_U, \quad (89)$$

where

$$\mathbf{g}_{n+1} = \frac{\partial \Delta \Phi_{n+1}}{\partial \Delta \mathbf{U}_{n+1}} = \bigwedge_{V^e \subset V} \sum_{\eta=0}^{m_e-1} \omega^{e,\eta} (\mathbf{B}^{e,\eta})^T [\boldsymbol{\sigma}_{n+1}] - \mathbf{f}_{n+1}^{\text{ext}}, \text{ and} \quad (90)$$

$$\mathbf{K}_{n+1} = \frac{\partial \mathbf{g}_{n+1}}{\partial \Delta \mathbf{U}_{n+1}} = \bigwedge_{V^e \subset V} \sum_{\eta=0}^{m_e-1} \omega^{e,\eta} (\mathbf{B}^{e,\eta})^T [\mathbf{C}_{n+1}] \mathbf{B}^{e,\eta}, \quad (91)$$

with  $\boldsymbol{\sigma}_{n+1}$  being the current stress tensor and  $\mathbf{C}_{n+1}$  being the current elastic tangent operator<sup>6</sup>

$$\boldsymbol{\sigma}_{n+1} = \frac{\partial \Delta \mathcal{E}_{n+1}^{\text{eff}}}{\partial \Delta \boldsymbol{\varepsilon}_{n+1}}, \text{ and } \mathbf{C}_{n+1} = \frac{\partial^2 \Delta \mathcal{E}_{n+1}^{\text{eff}}}{\partial \boldsymbol{\varepsilon}^2} = \mathbf{C}_{n+1}^{\text{el}}. \quad (92)$$

## 5. Numerical examples

The minimisation algorithm sketched in Alg. 2 is carried out on the D-wave systems. The binary optimisation problem (11) is submitted using the D-wave Ocean Python library [56]. Whenever the matrix  $\mathbf{A}$  in Eq. (11) is available, Alg. 4 shows the Python code to submit the binary optimisation problem to the D-wave quantum annealer. *DWaveSampler* is used to submit the problem to an available quantum processing unit. Since the submitted problem does not automatically fit the hardware graph, it is necessary to find an embedding for this purpose: we consider the general class *EmbeddingComposite* for embedding. Unless stated otherwise, the function *sample\_qubo* is used with 100 reads to evaluate the minimum of the

---

<sup>6</sup>Since the internal state is fixed, the material behaviour is considered as an elastic medium during this step of the double-minimisation problem.

objective function and 20  $\mu$ s for the annealing time. The output is chosen as the variables making the objective function the smallest.

---

**Algorithm 4** Binary optimisation problem submitted to the D-wave system.

---

```

from dwave.system import DWaveSampler, EmbeddingComposite
sampler = EmbeddingComposite(DWaveSampler())
sampleset = sampler.sample_qubo(A, num_reads=100, annealing_time=20)
b = sampleset.first.sample

```

---

In what follows, we present the numerical results obtained for elasto-plastic one and two-dimensional problems to demonstrate the applicability of the proposed framework to solve non-linear history-dependent mechanical problems.

### 5.1. Elasto-plastic bar under tensile loading

In this section, we apply the proposed hybrid framework on a uniaxial-strain tensile test, in order to study the effect of the number of qubits used in the binary discretisation.

The displacement field over a bar of length  $l = 1$ mm reads

$$\mathbf{u} = [u_x \ 0 \ 0]^T \text{ over } 0 \leq x \leq l. \quad (93)$$

The bar is clamped at  $x = 0$ . The governing equations (12-14) simplify for this case as

$$\begin{cases} \frac{d\sigma_{xx}}{dx} + b_0 &= 0, \\ u_x(x=0) &= 0, \text{ and} \\ \sigma_{xx}(x=l) &= 0, \end{cases} \quad (94)$$

where  $b_0 > 0$  is constant over the bar. The strain tensor  $\boldsymbol{\varepsilon}$  and the stress tensor  $\boldsymbol{\sigma}$  over the bar read

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_{xx} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \text{ and } \boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} & 0 & 0 \\ 0 & \sigma_{yy} & 0 \\ 0 & 0 & \sigma_{zz} \end{bmatrix} \quad (95)$$

where  $\varepsilon_{xx} = \frac{du_x}{dx}$  and  $\sigma_{yy} = \sigma_{zz}$ . For a given uniform  $b_0$ , the analytical solution can be found, see [Appendix B](#) for details. The material in the bar obeys  $J_2$ -elasto-plasticity with linear isotropic hardening, and with the material parameters reported in [Tab. 1](#). In this case, the plastic deformation does not occur inside the bar if

$$b_0 \leq \left( K + \frac{4}{3}\mu \right) \frac{\sigma_y^0}{2\mu l} = 122.5 \text{ MPa} \cdot \text{mm}^{-1}. \quad (96)$$

The purely elastic case is first tested with  $b_0 = 100$  MPa/mm. Different numbers of qubits, namely  $L = 2, 3$ , and 4 qubits, per degree of freedom are successively considered for three successive discretisations of the bar into 5, 10, and 20 elements. Since the solution remains in

Table 1: Material properties considered in the uniaxial-strain tensile test.

Parameter	Notation/Expression	Value/Expression	Unit
Young’s modulus	$E$	2	GPa
Poisson’s ratio	$\nu$	0.3	-
Initial yield stress	$\sigma_y^0$	70	MPa
Linear isotropic hardening	$R$	$20\gamma$	MPa

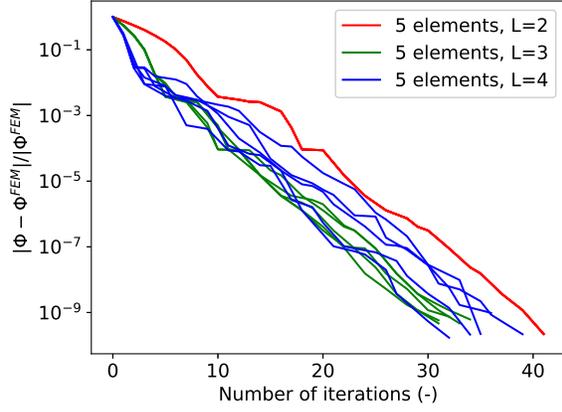
the linear elastic domain, only a single minimisation problem of the total potential energy is performed. Each case is solved with five different realisations to study the randomness in the resolution. Figure 2(a,c,e) shows the convergence history of all realisations and Fig. 2(b,d,f) illustrates the displacement field distributions predicted by the first realisation of each case, which are also compared to the reported analytical results. Figure 2(a,c,e) shows that, for a given discretisation, the convergence histories are different for each realisation, although in the case of 5 elements and  $L = 2$ , in which only a small number of qubits is required in the optimisation problem, the resolution exhibits no discrepancy. For all the studied cases, the accuracy keeps improving with the number of iterations, and relative errors as low as  $1 \times 10^{-9}$ —lower error can still be achieved by increasing the iterations number—can be reached.

From a practical point of view, the two most important user-defined parameters when using the DWave quantum annealers are the number of reads to evaluate the minimum of the objective function (`num_reads`) and the annealing time (`annealing_time`) as illustrated in Alg. 4. Considering the elastic case with a 20-element discretisation and  $L = 4$ , Fig. 3(a) studies the effect of the annealing time (`annealing_time`) successively taken as  $20 \mu\text{s}$  (the default value),  $60 \mu\text{s}$ , and  $200 \mu\text{s}$  whilst the number of reads is kept constant and equal to 100. It is shown that, in this range, the annealing time does not significantly affect the convergence. Fig. 3(b) studies the influence of the number of reads (`num_reads`) for a constant annealing time equal to  $20 \mu\text{s}$ . The higher the number of reads, the faster the convergence: the current quantum hardware is sensitive to various factors, such as environmental noises, hardware imperfections, and pre- and post-processing errors, *etc.*, so increasing the number of reads increases the probability of measuring the global minimum.

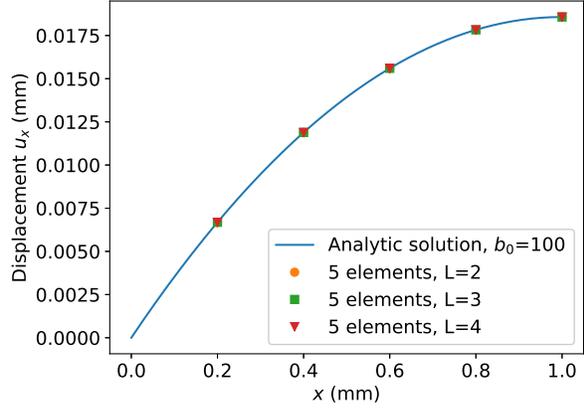
In order to assess the effect of the number of reads on the global computation time, the QPU access time for different realisations with different values of number of reads and different number of qubits  $L$  per degree of freedom is reported in Fig. 4(a). For a given number of qubits, a larger number of reads leads to a larger QPU access time, despite a lower total number of the iterations required to reach the same error, see Fig. 4(b). We also observe the saturation of the QPU access time with the number of qubits for a given value of the number of reads, with a similar trend for the total number of iterations. We note that the field of quantum computing is rapidly evolving, and research is ongoing to improve the stability and scalability of quantum systems, so the QPU access time could be improved in the future.

The elasto-plastic case is now considered with  $b_0 = 400 \text{ MPa/mm}$ . The bar is discretised into 20 elements. Because of the existence of the plastic flow, the double-minimisation process detailed in Alg. 3 is considered and each degree of freedom is discretised with three qubits.

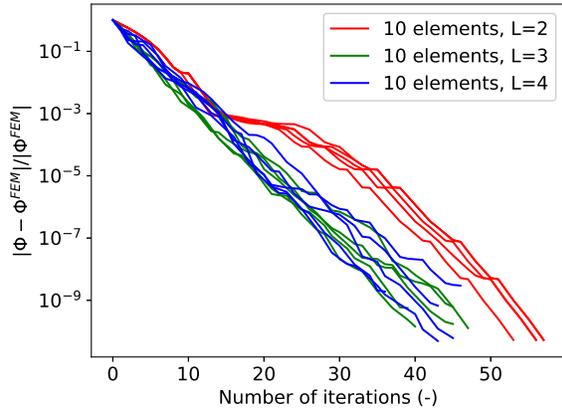
The obtained results in terms of the displacement distribution and of the equivalent plastic strain distribution are compared with the analytical results in Fig. 5, demonstrating that non-



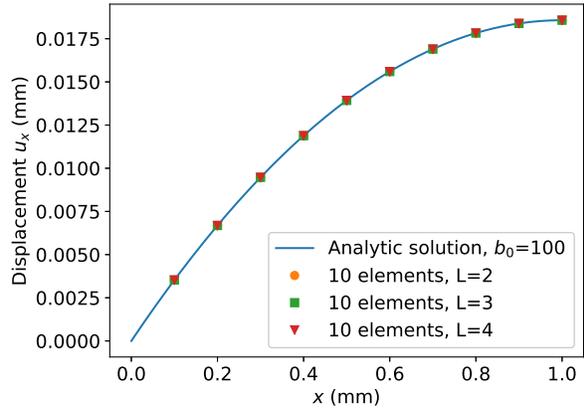
(a) 5 elements, convergence



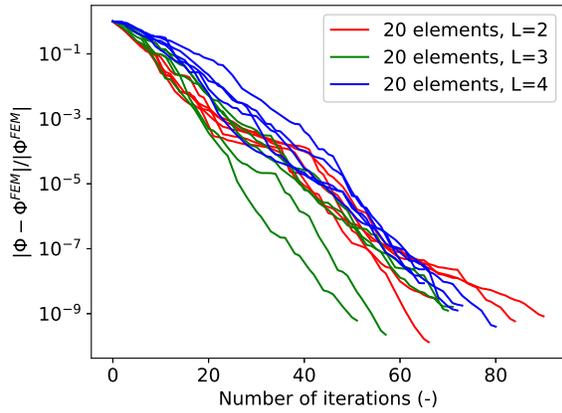
(b) 5 elements,  $u_x$



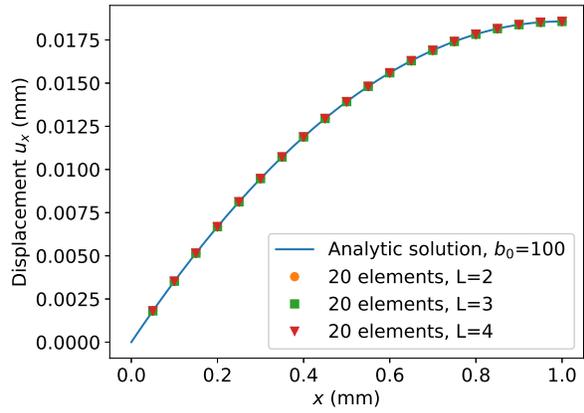
(c) 10 elements, convergence



(d) 10 elements,  $u_x$



(e) 20 elements, convergence



(f) 20 elements,  $u_x$

Figure 2: Uniaxial-strain tensile test with  $b_0 = 100 \text{ MPa} \cdot \text{mm}^{-1}$ : (a, c, e) convergence histories of 5 realisations for each value of  $L$ ; and (b, d, f) displacement distribution over the bar predicted by the first realisation for each value of  $L$ . The default annealing time equal to  $20 \mu\text{s}$  is considered.

linear elasto-plastic problems can be solved with the quantum annealing hybrid approach. Figure 6(a) shows the convergence history of the error in terms of the number of double-minimisation iterations, *i.e.* the number of iterations at line 3 of Alg. 3. For each double-minimisation iteration, the two local minimisation problems in terms of the displacement

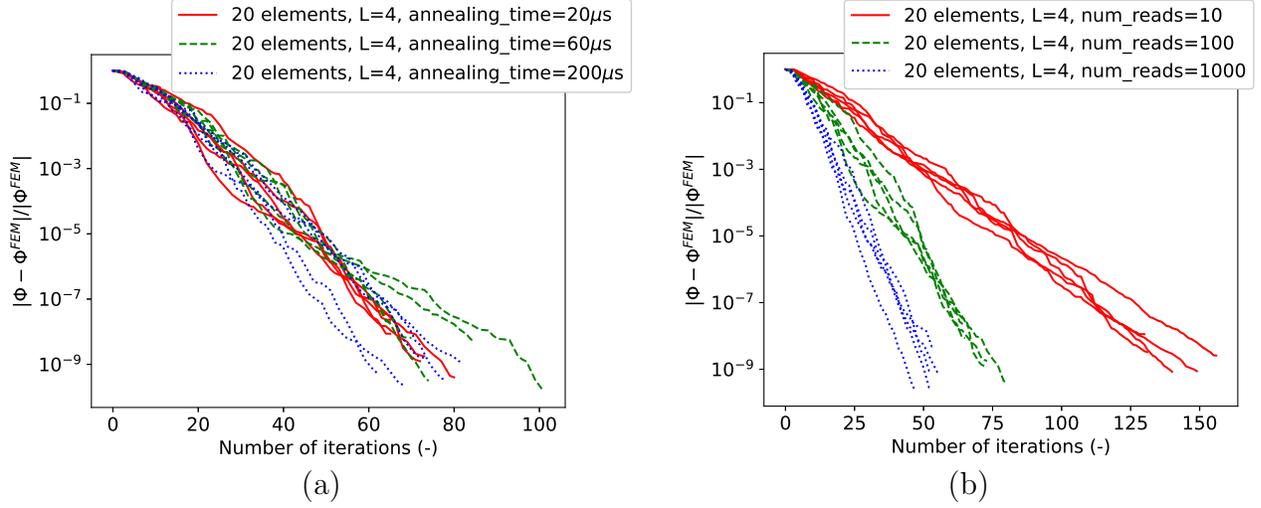


Figure 3: Uniaxial-strain tensile test with  $b_0 = 100 \text{ MPa} \cdot \text{mm}^{-1}$ : (a) effect of the annealing time (annealing\_time) for number\_reads=100; and (b) effect of the number of reads (number\_reads) for annealing\_time=20  $\mu\text{s}$  on the convergence history.

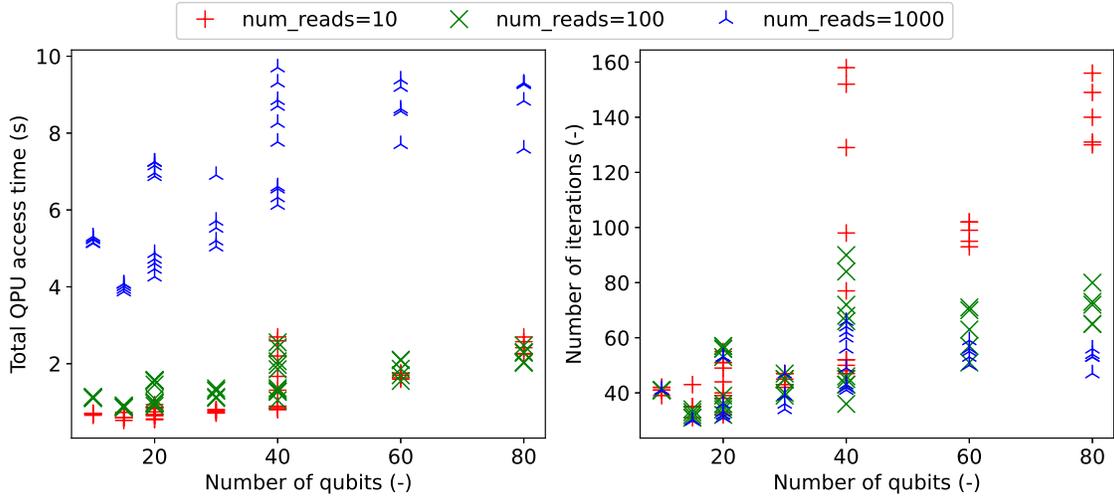


Figure 4: Uniaxial-strain tensile test with  $b_0 = 100 \text{ MPa} \cdot \text{mm}^{-1}$ : total QPU access time and number of iterations for each realisation in function of the number of qubits and number of reads (number\_reads). The default annealing time equal to 20  $\mu\text{s}$  is considered.

field and of the internal variables field, respectively lines 4 and 5 of Alg. 3, are solved through local iterations during the QA-SQP process following Alg. 2. The number of local iterations required for each local minimisation problem to reach a given accuracy of  $1 \times 10^{-9}$  is reported in Fig. 6(b), in which it can be seen that the number of local iterations decreases with the convergence of the double-minimisation problem. The convergence histories of the local displacement field optimisation problem and of the internal variables field optimisation at different double-minimisation iteration steps are shown in Fig. 7, showing that relatively low errors of  $1 \times 10^{-9}$  can be reached during the iterative process.

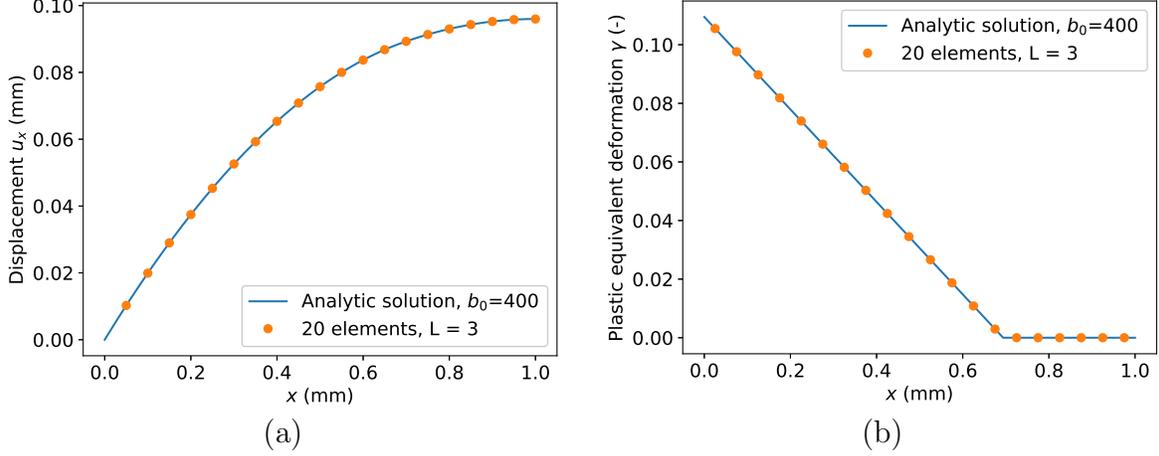


Figure 5: Uniaxial-strain tensile test with  $b_0 = 400 \text{ MPa} \cdot \text{mm}^{-1}$ : (a) displacement distribution; and (b) equivalent plastic strain distribution over the bar.

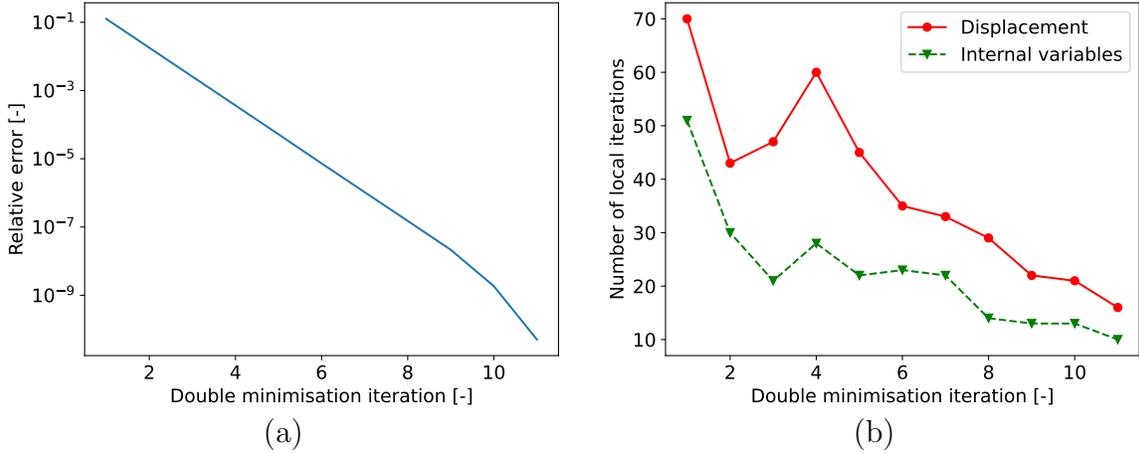


Figure 6: Uniaxial-strain tensile test with  $b_0 = 400 \text{ MPa} \cdot \text{mm}^{-1}$ : (a) convergence history in terms of the relative error *vs.* the number of double-minimisation iterations and; (b) number of local iterations required at each double-minimisation iteration. The relative error is defined by  $|(\Phi_{\mathbf{U}} - \Phi_{\mathbf{Q}})/\Phi_{\mathbf{U}0}|$  where  $\Phi_{\mathbf{U}}$  and  $\Phi_{\mathbf{U}0}$  are respectively the objective function values after minimisation with respect to the displacement unknowns and the one at the first iteration, and  $\Phi_{\mathbf{Q}}$  is the objective function value after minimisation with respect to the internal variable unknowns.

## 5.2. Elasto-plastic plate under tensile loading

In this section, we consider the proposed hybrid framework to study the cyclic loading of a two-dimensional plate under plane-strain condition, in order to demonstrate the capability of the framework to account for history-dependency. The finite element mesh and the boundary conditions are sketched in Fig. 8(a). The cyclic prescribed loading displacement is applied on the right edge of the plate as shown in Fig. 8(b). The material of the plate obeys a  $J_2$ -elasto-plastic model with a Swift isotropic hardening law as reported in Tab. 2. The current solution is compared to the one obtained with a conventional finite element method (FEM).

The current double-minimisation framework assisted by quantum annealing is conducted at four different times, *i.e.* four increments are considered to solve the problem. Since the

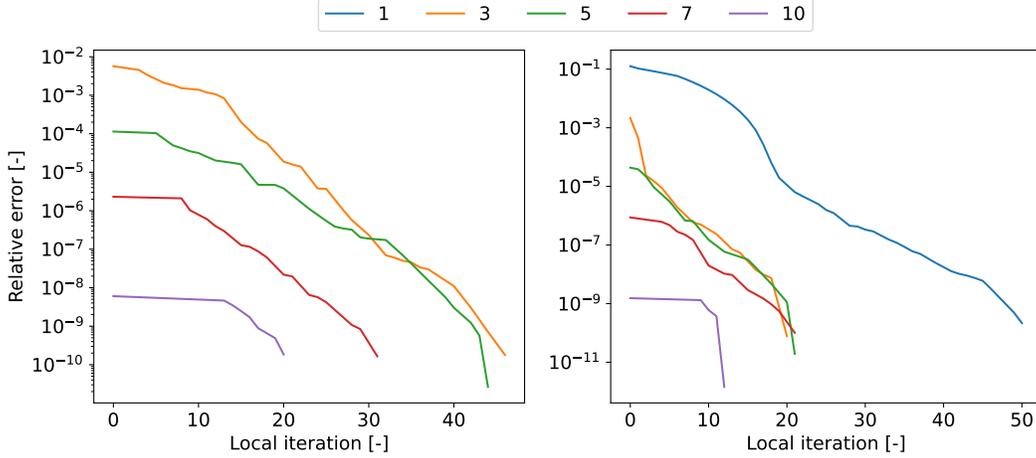


Figure 7: Uniaxial-strain tensile test with  $b_0 = 400 \text{ MPa} \cdot \text{mm}^{-1}$ : local iteration error convergence for the different double-minimisation iterations of (left) the displacement field optimisation; and (right) the internal variables field optimisation. The relative error is defined by  $|(\Phi - \Phi_{-1})/\Phi_0|$  where  $\Phi$  is the objective function value, and  $\Phi_0$  and  $\Phi_{-1}$  denote respectively the initial and last values of the objective function for each local minimisation.

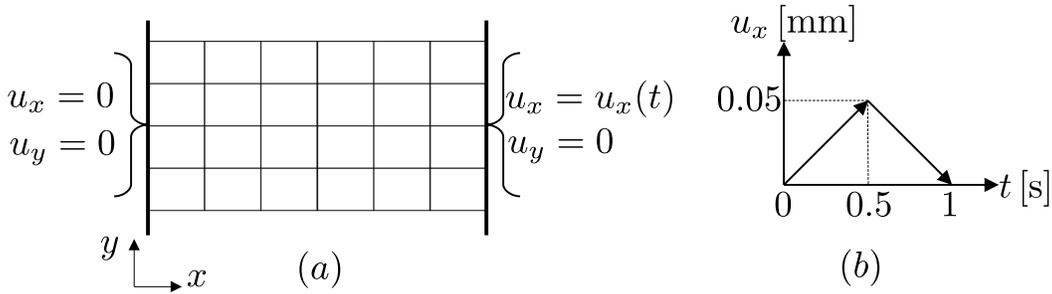


Figure 8: Plate under cyclic loading: (a) the finite element mesh of the  $1\text{mm} \times 0.5\text{mm}$ -plate under prescribed displacement boundary conditions; and (b) the cyclic displacement path applied to the right edge of the plate.

double-minimisation process can handle large increments (this is due to the fact that the elastic and plastic energies are correctly introduced), the whole simulation can be conducted using the developed QA-SQP in two increments only. Four increments are however considered in order to have intermediate solutions that can be compared with the classical finite element results. The considered times are  $t = 0.25\text{s}$ ,  $t = 0.5\text{s}$ ,  $t = 0.75\text{s}$  and  $t = 1\text{s}$ . Each degree of freedom is discretised with three qubits. Figures 9 and 10 compare the current framework with a classical FEM, respectively in terms of the reaction force on the left boundary of the plate *vs.* the prescribed displacement and of the distribution of the equivalent plastic strain at two configurations corresponding to  $t = 0.5\text{s}$  and  $t = 1\text{s}$ , see Fig. 8(b), demonstrating that the quantum annealing hybrid framework is able to model history-dependent behaviours.

Figure 11(a) shows the error evolution with the number of double-minimisation iterations, *i.e.* the number of iterations at line 3 of Alg. 3, at the four successive configuration resolutions. The numbers of local iterations required for each local minimisation problem, *i.e.* the QA-SQP resolution following Alg. 2 at lines 4 and 5 of Alg. 3, are reported in

Table 2: Material properties considered in the plate under cyclic loading.

Parameter	Notation/Expression	Value/Expression	Unit
Young's modulus	$E$	20	GPa
Poisson's ratio	$\nu$	0.3	-
Initial yield stress	$\sigma_y^0$	150	MPa
Isotropic hardening	$R$	$\sigma_y^0 \left[ \left( 1 + \frac{\gamma}{0.05} \right)^{0.1} - 1 \right]$	MPa

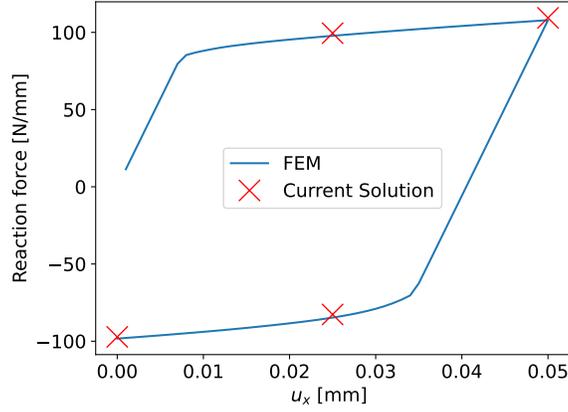


Figure 9: Plate under cyclic loading: comparison between the current framework and the classical FEM in terms of the reaction force on the left boundary of the plate *vs.* the prescribed displacement.

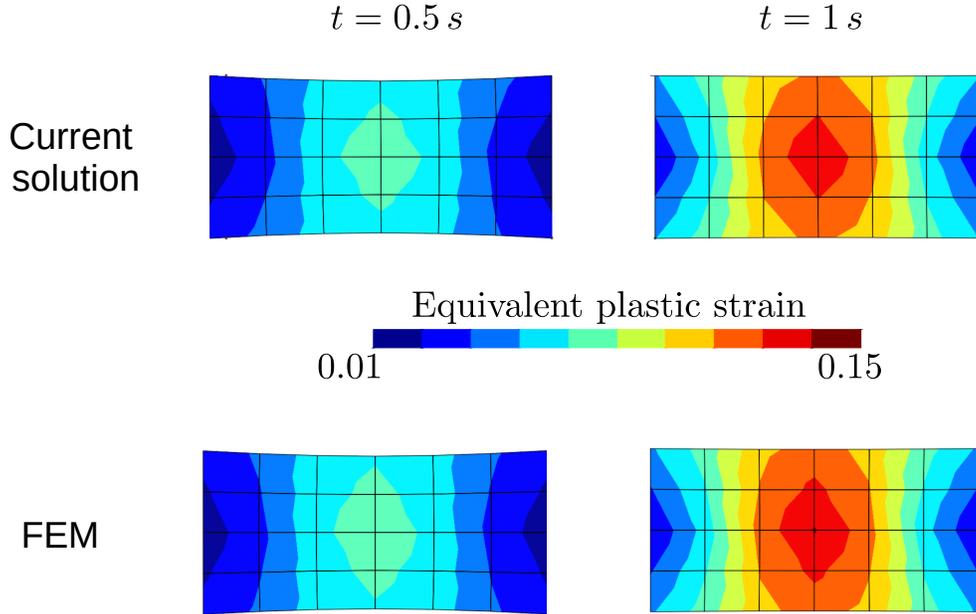


Figure 10: Plate under cyclic loading: comparison between the current framework and the classical FEM in terms of the distribution of the equivalent plastic strain at  $t = 0.5$  s and at  $t = 1$  s.

Fig. 11(b) for each of the double-minimisation iteration. The number of iterations for the

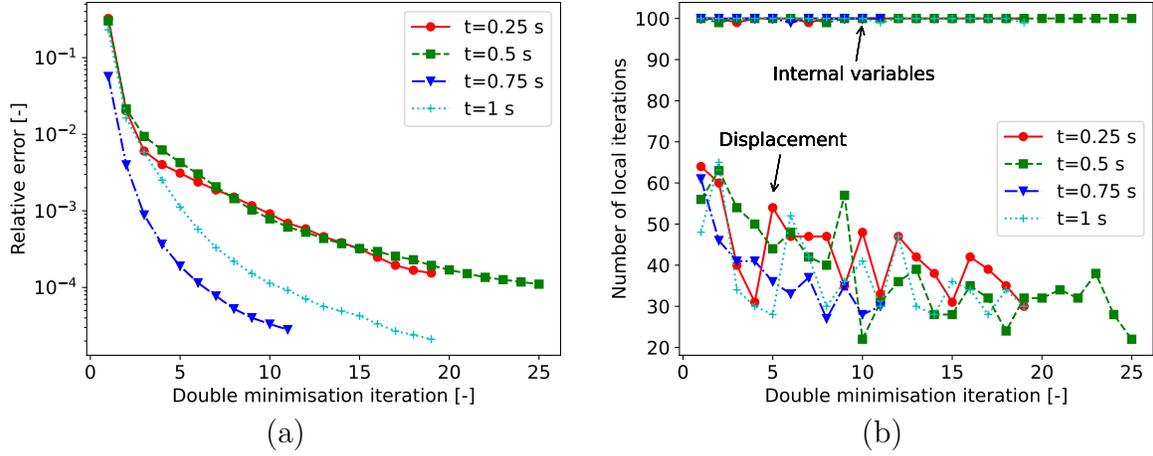


Figure 11: Plate under cyclic loading: (a) convergence history in terms of the relative error *vs.* the number of double-minimisation iterations and; (b) number of local iterations required at each double-minimisation iteration. The relative error is defined by  $|(\Phi_U - \Phi_Q)/\Phi_{U0}|$  where  $\Phi_U$  and  $\Phi_{U0}$  are respectively the objective function values after minimisation with respect to the displacement unknowns and the one at the first iteration, and  $\Phi_Q$  is the objective function value after minimisation with respect to the internal variable unknowns.

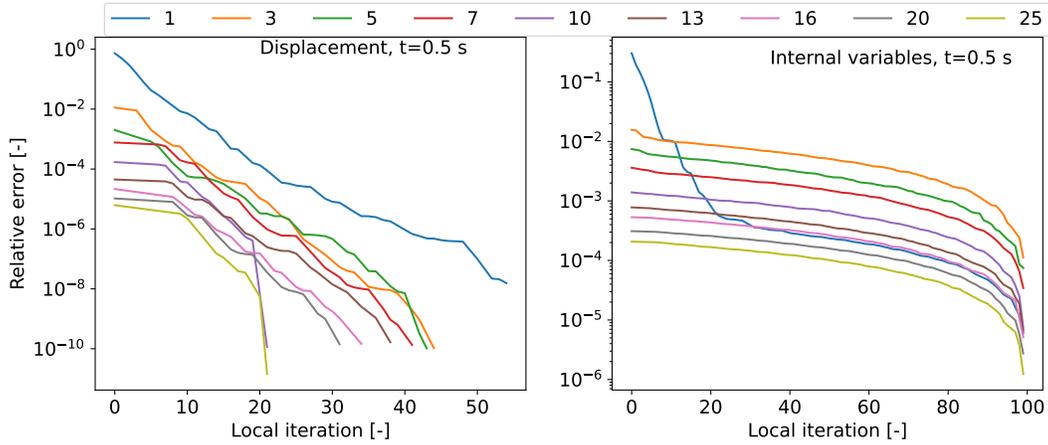


Figure 12: Plate under cyclic loading: local iteration error convergence for the different double-minimisation iterations of (left) the displacement field optimisation; and (right) the internal variables field optimisation. The relative error is defined by  $|(\Phi - \Phi_{-1})/\Phi_0|$  where  $\Phi$  is the objective function value, and  $\Phi_0$  and  $\Phi_{-1}$  denote respectively the initial and last values of the objective function for each local minimisation.

minimisation problem with respect to the internal variables field is limited to 100 in order to limit the queue waiting time. It is however noted that the double-minimisation strategy sketched in Alg. 3 does not require each local minimisation to be exact in order to reach the convergence of the double-minimisation problem, but only a solution improvement after each local minimisation. The convergence histories of the local displacement field optimisation problem and of the internal variables field optimisation at  $t = 0.5$  s are shown in Fig. 12 for different double-minimisation iteration steps, demonstrating the decrease of the objective functions after each minimisation problem resolution.

## 6. Conclusions

Quantum annealing has the potential to enhance finite element simulations in computational solid mechanics to study larger problem sizes, thus offering the prospect of more efficient and accurate solutions when analysing complex mechanical problems. Our study explores the possibility of quantum annealing techniques to be used as a solver in the context of the finite element method in order to tackle, in the future, computationally demanding problems, thus enabling researchers and engineers to explore new avenues in the design, analysis, and optimisation of mechanical structures and systems.

Towards this end, the weak form was reformulated as a multi-optimisation problem solving both the material constitutive models and the structure balance equations at the structural level of the finite element problem. After transforming the non-linear optimisation problems into series of quadratic ones, and after binarisation of the continuous degrees of freedom, the minimisation steps were shown to be compatible with the quantum annealer. As a result, the system resolution steps, included the ones related to the non-linear system of equations of the material model, are conducted on the quantum annealer, whilst the classical computer is used for the finite element assembly. The hybrid formulation was then applied on elasto-plastic 1D and 2D problems, in which it was shown that, on the one hand, error lower than  $1^{-9}$  can be achieved by considering 4 qubits to binarise a degree of freedom and that, on the other hand, that history-dependent behaviours can be modelled.

Although simple constitutive material models such as the presented J2-plasticity model do not present difficulty in their resolution with classical computers, this is not true for more complex material models, *e.g.* such as for finite-strain visco-elasticity visco-plasticity with pressure dependent yield surface, which involve several internal variables and for which deriving and evaluating the material operator require many algorithmic operations that slow down the resolution process. It is expected that with the development of quantum computers, the developed double-minimisation QA-SQP strategy, in which all the resolution steps are conducted on the quantum annealer whilst only assembly steps remain on the classical hardware, will present a clear advantage in the context of complex finite element simulations since quantum annealing is particularly efficient in evaluating the minimum point of a multi-variate quadratic function. Additionally, the double-minimisation strategy offers two other advantages: (i) as shown in Section 4.2, the minimisation problem in terms of the displacement field considers an elastic behaviour at a given internal variables state, which makes the material operator either easier to be evaluated or constant when defining the quadratic functional, and which is beneficial in terms of computational efficiency, and (ii) the minimisation problem in terms of the internal variables is simply the combination of independent constrained minimisation problems at each integration point since the displacement field is fixed; it thus does not require all the qubits to be connected and can therefore exploit the parallelism advantage of quantum computers.

We note that currently the size of the problem that can be solved is limited because of the current limitation in terms of available physical qubits number and of their interactions. Nevertheless, it is expected that with the progress in quantum computer hardware, problems of size larger than what can currently be handled with classical computers will become solvable.

## Acknowledgments

V.-D.N acknowledges the support of the Fonds National de la Recherche (F.R.S.-FNRS, Belgium).

F.R. acknowledges the support of the Fonds National de la Recherche (F.R.S.-FNRS, Belgium), #T0205.20.

This work is partially supported by a ‘‘Strategic Opportunity’’ grant from the University of Liege.

## Data availability

The code and raw/processed data required to reproduce these findings is available on [57] under the Creative Commons Attribution 4.0 International (CC BY 4.0) licence

## Appendix A. Finite element approximation

The domain  $V$  is discretised into a finite element mesh of  $N_{\text{ele}}$  elements, *i.e.*

$$V = \bigcup_{e=0}^{N_{\text{ele}}-1} V^e. \quad (\text{A.1})$$

The Neumann boundary  $\partial_N V$  is discretised accordingly into  $N_{\text{ele}}^{\text{neu}}$  elements with a lower dimension as

$$\partial_N V = \bigcup_{e=N_{\text{ele}}}^{N_{\text{ele}}+N_{\text{ele}}^{\text{neu}}-1} \Gamma^e. \quad (\text{A.2})$$

On an arbitrary element  $V^e$  (or  $\Gamma^e$ ), we approximate the displacement field  $\mathbf{u}$  and internal variables field  $\mathbf{q}$  under the following interpolants<sup>7</sup>

$$\mathbf{u}(\mathbf{x}) = \sum_{a=0}^{n_e} N_a^e(\mathbf{x}) \mathbf{u}_a^e, \text{ and} \quad (\text{A.3})$$

$$\mathbf{q}(\mathbf{x}) = \sum_{\eta=0}^{m_e} M^{\eta}(\mathbf{x}) \mathbf{q}^{\eta}, \quad (\text{A.4})$$

where  $N_a^e$  and  $M_\eta^e$  are respectively the displacement and internal variables shape functions,  $n_e$  and  $m_e$  are respectively the number of nodes and quadrature points on element  $e$ , and where  $\mathbf{u}_a^e$  and  $\mathbf{q}^{\eta}$  are the displacement and internal variables at the sampled points. On the one hand, the displacement field needs to be continuous in  $V$ , *i.e.*  $\mathbf{u} \in C_0(V)$ . This can be easily achieved using the isoparametric finite elements with Lagrange shape functions, in

---

<sup>7</sup>For ease of readability, the notation  $\bullet_a^e$  refers to the quantity  $\bullet$  on the node  $a$  of element  $e$  whilst  $\bullet^{\eta}$  refers to the quantity  $\bullet$  on the quadrature point  $\eta$  of element  $e$ .

which the Kronecker delta property is satisfied at all nodes of the elements, *i.e.*

$$N_a^e(\mathbf{x}_b^e) = \delta_{ab}, \quad \forall e, a, b, \quad (\text{A.5})$$

where  $\mathbf{x}_b^e$  for  $b = 0, \dots, n_e - 1$  are the coordinates of the nodes of element  $e$ . By contrast, since the internal variables field needs to be bounded, *i.e.*  $\mathbf{q} \in L^\infty(V)$ , the shape functions are assumed to satisfy the Kronecker delta property at quadrature points as

$$M^{e,\eta}(\Xi^{e,\zeta}) = \delta_{\eta\zeta}, \quad \forall e, \eta, \zeta, \quad (\text{A.6})$$

where  $\Xi^{e,\zeta}$  for  $\zeta = 0, \dots, m_e - 1$  are the quadrature points in element  $e$ . We note that the internal variables field  $\mathbf{q}$  is only used at the quadrature points, so that the interpolation (A.4) is licit.

Clearly, using finite element interpolants (A.3, A.4), the displacement unknowns vector (denoted by  $\mathbf{U}$ ) includes all displacements at the nodes of the mesh except the ones at the nodes on the Dirichlet boundary  $\partial_D V$  where the displacement field is prescribed, whilst the internal variable unknowns vector (denoted by  $\mathbf{Q}$ ) includes all the internal variables at the quadrature points of the elements.

Eq. (A.3) evaluated at the quadrature points can be organised in a matrix-vector operation as

$$\mathbf{u}^{e,\eta} = \mathbf{u}(\Xi^{e,\eta}) = \mathbf{N}^{e,\eta} \mathbf{U}^e, \quad (\text{A.7})$$

where  $\mathbf{N}^{e,\eta}$  is the elementary shape function matrix evaluated at the quadrature point  $\Xi^{e,\eta}$  and  $\mathbf{U}^e$  is the elementary unknowns vector gathering all elementary nodal displacements. Additionally, we introduce the operator  $[\bullet]$  to convert, respectively, a second-order tensor and a fourth-order tensor to a vector and a square matrix as follows

$$[\bullet]_{id+j} = \bullet_{ij} \quad \text{and} \quad [\bullet]_{id+j, kd+l} = \bullet_{ijkl}, \quad (\text{A.8})$$

where  $d$  is the dimension. As a result, Eq. (A.7) yields

$$[\boldsymbol{\varepsilon}^{e,\eta}] = [\nabla \otimes^s \mathbf{u}|_{\Xi^{e,\eta}}] = \mathbf{B}^{e,\eta} \mathbf{U}^e, \quad (\text{A.9})$$

where  $\mathbf{B}^{e,\eta}$  is the elementary matrix of the gradient of the shape functions associated with the displacement vector field at the quadrature point  $\Xi^{e,\eta}$ .

The energy function (30) is estimated at a quadrature point, leading to

$$\Delta \Phi_{n+1} = \Delta W_{n+1}^{\text{int}} - \Delta W_{n+1}^{\text{ext}}, \quad \text{with} \quad (\text{A.10})$$

$$\Delta W_{n+1}^{\text{int}} = \sum_{e=0}^{N_{\text{ele}}-1} \sum_{\eta=0}^{m_e-1} \omega^{e,\eta} \Delta \mathcal{E}_{n+1}(\Delta \boldsymbol{\varepsilon}_{n+1}^{e,\eta}, \Delta \mathbf{q}_{n+1}^{e,\eta}), \quad \text{and} \quad (\text{A.11})$$

$$\Delta W_{n+1}^{\text{ext}} = \sum_{e=0}^{N_{\text{ele}}-1} \sum_{\eta=0}^{m_e-1} \omega^{e,\eta} \mathbf{b}_0^{e,\eta} \cdot \Delta \mathbf{u}_{n+1}^{e,\eta} + \sum_{e=N_{\text{ele}}}^{N_{\text{ele}}+N_{\text{ele}}^{\text{neu}}-1} \sum_{\eta=0}^{m_e-1} \omega^{e,\eta} \mathbf{t}_0^{e,\eta} \cdot \Delta \mathbf{u}_{n+1}^{e,\eta}, \quad (\text{A.12})$$

where  $\bullet^{e,\eta}$  denotes a quantity  $\bullet$  at the quadrature point  $\eta$  of element  $e$ , and  $\omega^{e,\eta}$ ,  $\forall e, \eta$ ,

denotes the weights of the corresponding quadrature points  $\Xi^{e,\eta}$ . Using Eq. (A.9), Eqs. (A.11, A.12) become

$$\Delta W_{n+1}^{\text{int}} = \sum_{e=0}^{N_{\text{ele}}-1} \sum_{\eta=0}^{m_e-1} \omega^{e,\eta} \Delta \mathcal{E}_{n+1} (\mathbf{B}^{e,\eta} \Delta \mathbf{U}_{n+1}^e, \Delta \mathbf{q}_{n+1}^{e,\eta}), \text{ and} \quad (\text{A.13})$$

$$\Delta W_{n+1}^{\text{ext}} = \Delta \mathbf{U}_{n+1}^T \mathbf{f}_{n+1}^{\text{ext}}, \quad (\text{A.14})$$

where  $\mathbf{f}_{n+1}^{\text{ext}}$  is the external force vector

$$\mathbf{f}_{n+1}^{\text{ext}} = \bigwedge_{V^e \subset V} \sum_{\eta=0}^{m_e-1} \omega^{e,\eta} (\mathbf{N}^{e,\eta})^T \mathbf{b}_0^{e,\eta} + \bigwedge_{\Gamma^e \subset \partial_N V} \sum_{\eta=0}^{m_e-1} \omega^{e,\eta} (\mathbf{N}^{e,\eta})^T \mathbf{t}_0^{e,\eta}. \quad (\text{A.15})$$

with  $\bigwedge$  being used to indicate the assembling process.

## Appendix B. Analytical solution of an elasto-plastic bar under uniaxial strain test

The plastic deformation is deviatoric, implying the following plastic deformation tensor

$$\boldsymbol{\varepsilon}^{\text{p}} = \begin{bmatrix} \varepsilon_{xx}^{\text{p}} & 0 & 0 \\ 0 & -\frac{1}{2}\varepsilon_{xx}^{\text{p}} & 0 \\ 0 & 0 & -\frac{1}{2}\varepsilon_{xx}^{\text{p}} \end{bmatrix}, \quad (\text{B.1})$$

where  $\varepsilon_{xx}^{\text{p}} > 0$  as the result of the monotonic tensile loading. The equivalent plastic strain is given as

$$\dot{\gamma} = \sqrt{\frac{2}{3} \dot{\boldsymbol{\varepsilon}}^{\text{p}} : \dot{\boldsymbol{\varepsilon}}^{\text{p}}} = \dot{\varepsilon}_{xx}^{\text{p}}, \quad (\text{B.2})$$

implying  $\gamma = \varepsilon_{xx}^{\text{p}}$ . As a result, the Hooke's law can be rewritten as

$$\boldsymbol{\sigma} = K \varepsilon_{xx} \mathbf{I} + 2\mu \begin{bmatrix} \frac{2}{3}\varepsilon_{xx} - \gamma & 0 & 0 \\ 0 & \frac{\gamma}{2} - \frac{\varepsilon_{xx}}{3} & 0 \\ 0 & 0 & \frac{\gamma}{2} - \frac{\varepsilon_{xx}}{3} \end{bmatrix}, \quad (\text{B.3})$$

where  $K$  and  $\mu$  are the shear and bulk moduli, respectively. The last equation leads to

$$\sigma_{xx} = \left( K + \frac{4}{3}\mu \right) \varepsilon_{xx} - 2\mu\gamma, \text{ and} \quad (\text{B.4})$$

$$\sigma_{\text{eq}} = \sqrt{\frac{3}{2} \text{dev}(\boldsymbol{\sigma}) : \text{dev}(\boldsymbol{\sigma})} = 3\mu \left( \frac{2}{3}\varepsilon_{xx} - \gamma \right). \quad (\text{B.5})$$

Equations (94) lead to the solution of  $\sigma_{xx}$  as

$$\sigma_{xx} = b_0 (l - x) . \quad (\text{B.6})$$

The Kuhn-Tucker condition reads

$$\sigma_{\text{eq}} - \sigma_y^0 - H\gamma \leq 0, \gamma \geq 0, \text{ and } [\sigma_{\text{eq}} - \sigma_y^0 - H\gamma] \gamma = 0, \quad (\text{B.7})$$

where  $\sigma_y^0$  is the onset of plasticity and  $H$  is the linear hardening modulus.

Finally, one has the following equations for  $\varepsilon_{xx}$ ,  $\gamma$ , and  $u_x$ :

$$\left\{ \begin{array}{l} (K + \frac{4}{3}\mu) \varepsilon_{xx} - 2\mu\gamma = b_0 (l - x) , \\ 3\mu \left( \frac{2}{3}\varepsilon_{xx} - \gamma \right) - (\sigma_y^0 + H\gamma) \leq 0, \gamma \geq 0, \\ \left[ 3\mu \left( \frac{2}{3}\varepsilon_{xx} - \gamma \right) - (\sigma_y^0 + H\gamma) \right] \gamma = 0, \text{ and} \\ u_x = \int_0^x \varepsilon_{xx} dx . \end{array} \right. \quad (\text{B.8})$$

The system of equations above yields the following solution

$$\left\{ \begin{array}{l} \text{if } x \geq x_c, \left\{ \begin{array}{l} \varepsilon_{xx} = \frac{b_0 (l - x)}{K + \frac{4}{3}\mu} \\ \gamma = 0 \\ u_x = \begin{cases} \frac{b_0}{K + \frac{4}{3}\mu} \left( lx - \frac{x^2}{2} \right) & \text{if } x_c < 0, \text{ and} \\ u_c + \frac{b_0}{K + \frac{4}{3}\mu} \left[ l(x - x_c) - \frac{x^2 - x_c^2}{2} \right] & \text{if } x_c \geq 0 \end{cases} \end{array} \right. \\ \text{if } x < x_c, \left\{ \begin{array}{l} \varepsilon_{xx} = \frac{1}{K + \frac{4}{3}\mu - \frac{4\mu^2}{3\mu + H}} \left[ b_0 (l - x) - \frac{2\mu\sigma_y^0}{3\mu + H} \right] \\ \gamma = \frac{2\mu}{3\mu + H} \varepsilon_{xx} - \frac{\sigma_y^0}{3\mu + H} \\ u_x = \frac{1}{K + \frac{4}{3}\mu - \frac{4\mu^2}{3\mu + H}} \left[ b_0 \left( lx - \frac{x^2}{2} \right) - \frac{2\mu\sigma_y^0 x}{3\mu + H} \right] \end{array} \right. \end{array} \right. \quad (\text{B.9})$$

where

$$x_c = l - \left(K + \frac{4}{3}\mu\right) \frac{\sigma_y^0}{2\mu b_0}, \text{ and} \quad (\text{B.10})$$

$$u_c = \frac{1}{K + \frac{4}{3}\mu - \frac{4\mu^2}{3\mu + H}} \left[ b_0 \left( l x_c - \frac{x_c^2}{2} \right) - \frac{2\mu\sigma_y^0 x_c}{3\mu + H} \right]. \quad (\text{B.11})$$

It is noted that  $x_c$  is the transition point between the elastic and the elasto-plastic regions. There is no plastic deformation inside the bar when  $x_c \leq 0$ , implying

$$b_0 \leq \left(K + \frac{4}{3}\mu\right) \frac{\sigma_y^0}{2\mu l}. \quad (\text{B.12})$$

## References

- [1] P. Wriggers, *Nonlinear finite element methods*, Springer Science & Business Media, 2008.
- [2] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell, et al., Quantum supremacy using a programmable superconducting processor, *Nature* 574 (2019) 505–510.
- [3] E. Pednault, J. Gunnels, D. Maslov, J. Gambetta, On “quantum supremacy”, IBM Research Blog (2019). URL: <https://go.nature.com/34Qh40P>.
- [4] D. Castelvecchi, Ibm quantum computer passes calculation milestone, *Nature* 618 (2023) 656–657. doi:[10.1038/d41586-023-01965-3](https://doi.org/10.1038/d41586-023-01965-3).
- [5] M. A. Nielsen, I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*, Cambridge University Press, 2010. doi:[10.1017/CB09780511976667](https://doi.org/10.1017/CB09780511976667).
- [6] National Academies of Sciences, Engineering, and Medicine, *Quantum Computing: Progress and Prospects*, The National Academies Press, Washington, DC, 2019. doi:[10.17226/25196](https://doi.org/10.17226/25196).
- [7] T. Kadowaki, H. Nishimori, Quantum annealing in the transverse ising model, *Physical Review E* 58 (1998) 5355.
- [8] J. Brooke, D. Bitko, Rosenbaum, G. Aeppli, Quantum annealing of a disordered magnet, *Science* 284 (1999) 779–781.
- [9] S. Tanaka, R. Tamura, B. K. Chakrabarti, *Quantum spin glasses, annealing and computation*, Cambridge University Press, 2017.
- [10] G. Tosti Balducci, B. Chen, M. Möller, M. Gerritsma, R. De Breuker, Review and perspectives in quantum computing for partial differential equations in structural mechanics, *Frontiers in Mechanical Engineering* 8 (2022). URL: <https://www.frontiersin.org/articles/10.3389/fmech.2022.914241>. doi:[10.3389/fmech.2022.914241](https://doi.org/10.3389/fmech.2022.914241).

- [11] A. M. Childs, R. Kothari, R. D. Somma, Quantum algorithm for systems of linear equations with exponentially improved dependence on precision, *SIAM Journal on Computing* 46 (2017) 1920–1950.
- [12] A. W. Harrow, A. Hassidim, S. Lloyd, Quantum algorithm for linear systems of equations, *Physical review letters* 103 (2009) 150502.
- [13] A. Montanaro, S. Pallister, Quantum algorithms and the finite element method, *Physical Review A* 93 (2016) 032324.
- [14] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, et al., Variational quantum algorithms, *Nature Reviews Physics* 3 (2021) 625–644.
- [15] D. O’Malley, V. V. Vesselinov, Toq. jl: A high-level programming language for d-wave machines based on julia, in: 2016 IEEE High Performance Extreme Computing Conference (HPEC), IEEE, 2016, pp. 1–7.
- [16] A. Borle, S. J. Lomonaco, Analyzing the quantum annealing approach for solving linear least squares problems, in: WALCOM: Algorithms and Computation: 13th International Conference, WALCOM 2019, Guwahati, India, February 27–March 2, 2019, Proceedings 13, Springer, 2019, pp. 289–301.
- [17] P. Date, D. Arthur, L. Pusey-Nazzaro, Qubo formulations for training machine learning models, *Scientific reports* 11 (2021) 10029.
- [18] J. C. Criado, M. Spannowsky, Qade: solving differential equations on quantum annealers, *Quantum Science and Technology* 8 (2022) 015021. URL: <https://dx.doi.org/10.1088/2058-9565/acaa51>. doi:10.1088/2058-9565/acaa51.
- [19] R. Conley, D. Choi, G. Medwig, E. Mroczko, D. Wan, P. Castillo, K. Yu, Quantum optimization algorithm for solving elliptic boundary value problems on d-wave quantum annealing device, in: *Quantum Computing, Communication, and Simulation III*, volume 12446, SPIE, 2023, pp. 53–63.
- [20] S. Srivastava, V. Sundararaghavan, Box algorithm for the solution of differential equations on a quantum annealer, *Phys. Rev. A* 99 (2019) 052355. URL: <https://link.aps.org/doi/10.1103/PhysRevA.99.052355>. doi:10.1103/PhysRevA.99.052355.
- [21] O. M. Raisuddin, S. De, Feqa: Finite element computations on quantum annealers, *Computer Methods in Applied Mechanics and Engineering* 395 (2022) 115014. URL: <https://www.sciencedirect.com/science/article/pii/S0045782522002444>. doi:<https://doi.org/10.1016/j.cma.2022.115014>.
- [22] K. Endo, Y. Matsuda, S. Tanaka, M. Muramatsu, A phase-field model by an ising machine and its application to the phase-separation structure of a diblock polymer, *Scientific Reports* 12 (2022) 10794. URL: <https://doi.org/10.1038/s41598-022-14735-4>. doi:10.1038/s41598-022-14735-4.

- [23] M. Ortiz, L. Stainier, The variational formulation of viscoplastic constitutive updates, *Computer methods in applied mechanics and engineering* 171 (1999) 419–444.
- [24] R. Radovitzky, M. Ortiz, Error estimation and adaptive meshing in strongly nonlinear dynamic problems, *Computer Methods in Applied Mechanics and Engineering* 172 (1999) 203–240.
- [25] K. Weinberg, A. Mota, M. Ortiz, A variational constitutive model for porous metal plasticity, *Computational Mechanics* 37 (2006) 142–152.
- [26] W. W. Hager, H. Zhang, A survey of nonlinear conjugate gradient methods, *Pacific journal of Optimization* 2 (2006) 35–58.
- [27] A. Ruszczyński, *Nonlinear optimization*, Princeton university press, 2011.
- [28] J.-F. Bonnans, J. C. Gilbert, C. Lemaréchal, C. A. Sagastizábal, *Numerical optimization: theoretical and practical aspects*, Springer Science & Business Media, 2006.
- [29] A. J., A. Adedoyin, J. Ambrosiano, P. Anisimov, W. Casper, G. Chennupati, C. Coffrin, H. Djidjev, D. Gunter, S. Karra, N. Lemons, S. Lin, A. Malyzhenkov, D. Mascarenas, S. Mniszewski, B. Nadiga, D. O’malley, D. Oyen, S. Pakin, L. Prasad, R. Roberts, P. Romero, N. Santhi, N. Sinitsyn, P. J. Swart, J. G. Wendelberger, B. Yoon, R. Zamora, W. Zhu, S. Eidenbenz, A. Bärtschi, P. J. Coles, M. Vuffray, A. Y. Lokhov, Quantum algorithm implementations for beginners, *ACM Transactions on Quantum Computing* 3 (2022). URL: <https://doi.org/10.1145/3517340>. doi:10.1145/3517340.
- [30] J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li, E. Grant, L. Wossnig, I. Rungger, G. H. Booth, et al., The variational quantum eigensolver: a review of methods and best practices, *Physics Reports* 986 (2022) 1–128.
- [31] M. Born, V. Fock, Beweis des adiabatensatzes, *Zeitschrift für Physik* 51 (1928) 165–180.
- [32] T. Albash, D. A. Lidar, Adiabatic quantum computation, *Reviews of Modern Physics* 90 (2018) 015002.
- [33] M. W. Johnson, M. H. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. J. Berkley, J. Johansson, P. Bunyk, et al., Quantum annealing with manufactured spins, *Nature* 473 (2011) 194–198.
- [34] S. Yarkoni, E. Raponi, T. Bäck, S. Schmitt, Quantum annealing for industry applications: Introduction and review, *Reports on Progress in Physics* (2022).
- [35] C. C. McGeoch, Theory versus practice in annealing-based quantum computing, *Theoretical Computer Science* 816 (2020) 169–183. URL: <https://www.sciencedirect.com/science/article/pii/S0304397520300529>. doi:<https://doi.org/10.1016/j.tcs.2020.01.024>.
- [36] A. Rajak, S. Suzuki, A. Dutta, B. K. Chakrabarti, Quantum annealing: An overview, *Philosophical Transactions of the Royal Society A* 381 (2023) 20210417.

- [37] D-Wave System, The Advantage System: Performance Update, Technical Report, 2021. URL: [https://www.dwavesys.com/media/kjtlcemb/14-1054a-a\\_advantage\\_system\\_performance\\_update.pdf](https://www.dwavesys.com/media/kjtlcemb/14-1054a-a_advantage_system_performance_update.pdf).
- [38] D. Willsch, M. Willsch, C. D. Gonzalez Calaza, F. Jin, H. De Raedt, M. Svensson, K. Michielsen, Benchmarking advantage and d-wave 2000q quantum annealers with exact cover problems, *Quantum Information Processing* 21 (2022) 141.
- [39] J. Mosler, O. Bruhns, On the implementation of rate-independent standard dissipative solids at finite strain–variational constitutive updates, *Computer Methods in Applied Mechanics and Engineering* 199 (2010) 417–429.
- [40] L. Stainier, M. Ortiz, Study and validation of a variational theory of thermo-mechanical coupling in finite visco-plasticity, *International journal of solids and structures* 47 (2010) 705–715.
- [41] Q. Yang, L. Stainier, M. Ortiz, A variational formulation of the coupled thermo-mechanical boundary-value problem for general dissipative solids, *Journal of the Mechanics and Physics of Solids* 54 (2006) 401–424.
- [42] E. Fancello, J.-P. Ponthot, L. Stainier, A variational framework for nonlinear viscoelastic models in finite deformation regime, *Journal of Computational and Applied Mathematics* 215 (2008) 400–408.
- [43] J. Vassoler, L. Reips, E. Fancello, A variational framework for fiber-reinforced viscoelastic soft tissues, *International Journal for Numerical Methods in Engineering* 89 (2012) 1691–1706.
- [44] O. Kintzel, J. Mosler, A coupled isotropic elasto-plastic damage model based on incremental minimization principles, *Technische Mechanik-European Journal of Engineering Mechanics* 30 (2010) 177–184.
- [45] O. Kintzel, J. Mosler, An incremental minimization principle suitable for the analysis of low cycle fatigue in metals: A coupled ductile–brittle damage model, *Computer methods in applied mechanics and engineering* 200 (2011) 3127–3138.
- [46] C. Miehe, Strain-driven homogenization of inelastic microstructures and composites based on an incremental variational formulation, *International Journal for numerical methods in engineering* 55 (2002) 1285–1322.
- [47] L. Brassart, L. Stainier, I. Doghri, L. Delannay, A variational formulation for the incremental homogenization of elasto-plastic composites, *Journal of the Mechanics and Physics of Solids* 59 (2011) 2455–2475.
- [48] B. Halphen, Q. S. Nguyen, Sur les matériaux standard généralisés, *Journal de mécanique* 14 (1975) 39–63.
- [49] E. A. de Souza Neto, D. Peric, D. R. Owen, *Computational methods for plasticity: theory and applications*, John Wiley & Sons, 2011.

- [50] J. Bleyer, Automating the formulation and resolution of convex variational problems: Applications from image processing to computational mechanics, *ACM Trans. Math. Softw.* 46 (2020). URL: <https://doi.org/10.1145/3393881>. doi:10.1145/3393881.
- [51] J. Bleyer, Applications of conic programming in non-smooth mechanics, *Journal of Optimization Theory and Applications* (2022). URL: <https://doi.org/10.1007/s10957-022-02105-z>. doi:10.1007/s10957-022-02105-z.
- [52] I. G. Rosenberg, Reduction of bivalent maximization to the quadratic case., *Cahiers Centre Etudes Rech. Oper.* 17 (1975) 71–74.
- [53] E. Boros, P. L. Hammer, Pseudo-boolean optimization, *Discrete applied mathematics* 123 (2002) 155–225.
- [54] S. Jiang, K. A. Britt, A. J. McCaskey, T. S. Humble, S. Kais, Quantum annealing for prime factorization, *Scientific reports* 8 (2018) 17667.
- [55] A. Mandal, A. Roy, S. Upadhyay, H. Ushijima-Mwesigwa, Compressed quadratization of higher order binary optimization problems, in: *Proceedings of the 17th ACM International Conference on Computing Frontiers*, 2020, pp. 126–131.
- [56] D-wave system documentation, <https://docs.dwavesys.com/docs/latest/index.html>, 2022. Accessed: (February 4, 2024).
- [57] V. D. Nguyen, L. Wu, F. Remacle, L. Noels, Data of “a quantum annealing-sequential quadratic programming assisted finite element simulation for non-linear and history-dependent mechanical problems”, 2024. URL: <dx.doi.org/10.5281/zenodo.10451584>. doi:10.5281/zenodo.10451584.