Master Thesis

Exploring Quantum Annealing for Coupled Structural Analysis Problems

carried out for the purpose of obtaining the degree of

Diplom-Ingenieur_in (Dipl.-Ing.in or DI),

submitted at TU Wien

Faculty of Mechanical and Industrial Engineering

by

Lukas FREINBERGER

Mat.No.: 11916883

under the supervision of

Projektass. Dr.-Ing. Fabian Key, MSc

Institute of Lightweight Design and Structural Biomechanics

Place, Date

Wien, 18.02.2025

[redacted]

Signature

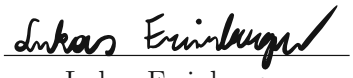
Affidavit

I declare in lieu of oath, that I wrote this thesis and carried out the associated research myself, using only the literature cited in this volume. If text passages from sources are used literally, they are marked as such.

I confirm that this work is original and has not been submitted for examination elsewhere, nor is it currently under consideration for a thesis elsewhere.

I acknowledge that the submitted work will be checked electronically-technically using suitable and state-of-the-art means (plagiarism detection software). On the one hand, this ensures that the submitted work was prepared according to the high-quality standards within the applicable rules to ensure good scientific practice "Code of Conduct" at the TU Wien. On the other hand, a comparison with other student theses avoids violations of my personal copyright.

Wien, 18.02.2025



Lukas Freinberger

Acknowledgements

I would like to thank my advisor, Dr. Fabian Key, for his professional guidance and continuous support during the research and composition of this thesis. His expertise and insightful feedback have been extremely helpful throughout the entire process. I also appreciate the freedom he gave me to explore my own ideas.

A special thank you goes to Prof. Mayu Muramatsu and the entire research group at Keio University in Japan. My time there was not only a remarkable academic experience but also a deeply enriching personal journey. The warmth, openness, and inspiring environment within the group made my stay unforgettable.

I am also sincerely grateful to Prof. Stefanie Elgeti, who laid the foundation for this opportunity abroad. Without her support, this chapter of my academic path would not have happened.

Contents

English Abstract

In this work, we develop and explore novel methods for using quantum annealing (QA) to solve coupled structural analysis problems, specifically focusing on fluid structure interaction (FSI) problems. These problems involve the interaction between a fluid and a structural domain, making them computationally demanding. When combined with design optimization tasks, they form complex optimization problems that are difficult to solve using classical methods, creating the need to explore alternative strategies such as QA.
QA is an emerging computational technique that utilizes quantum mechanical effects to solve complex optimization problems much faster and more efficiently than classical approaches. To optimize FSI problems using QA, it is first necessary to develop methods that allow using QA to solve FSI problems themselves, which is the focus of this work.
Our novel method is based on the partitioned approach, where the fluid and structural subproblems are treated separately. Using energy principles, we formulate the structural subproblem as an optimization problem and express the system's energy in a quadratic unconstrained binary optimization (QUBO) model, allowing the problem to be solved by QA. We validate our proposed method on a state-of-the-art D-Wave quantum annealer, with the results demonstrating the feasibility of QA for solving simple FSI problems.
Additionally, we analyze the capabilities and limitations of current QA hardware for these applications and explore strategies to overcome existing constraints. This study should lay the groundwork for future advancements in applying QA to engineering problems involving coupled structural analysis and optimization.

Deutsche Kurzfassung

Diese Arbeit befasst sich mit der Entwicklung und Untersuchung neuartiger Methoden zur Anwendung von Quantum Annealing (QA) auf gekoppelte Probleme der Strukturmechanik. Dabei liegt der Fokus auf Fluid-Struktur-Interaktion (FSI)-Problemen. Diese beschreiben die Interaktion zwischen einer Fluid- und einer Strukturdomäne und sind rechnerisch anspruchsvoll zu lösen. Insbesondere in Kombination mit Designoptimierung entstehen komplexe Optimierungsprobleme, deren Lösung mit klassischen Methoden herausfordernd ist. Dies verdeutlicht den Bedarf der Erforschung alternativer Rechenverfahren wie QA.
QA ist eine auf quantenmechanischen Effekten basierende Methode, die komplexe Optimierungsprobleme schnell und effizient lösen kann. Um FSI-Probleme in Kombination mit Designoptimierung mittels QA lösen zu können, muss zunächst eine Methode entwickelt werden, die eine Anwendung von QA auf FSI-Probleme erlaubt. Die vorliegende Arbeit widmet sich der Entwicklung einer solchen Methodik.
Der entwickelte Ansatz basiert auf der partitionierten Lösungsmethode, bei der Fluid- und Strukturprobleme separat behandelt werden. Unter Anwendung von Energieprinzipien wird das Strukturproblem als Optimierungsproblem formuliert und die Energie des Systems als Quadratic Unconstrained Binary Optimization (QUBO)-Modell dargestellt, wodurch eine Lösung mittels QA ermöglicht wird. Zur Validierung der entwickelten Methode erfolgt eine Implementierung auf einem D-Wave-Quantenannealer. Die Ergebnisse zeigen die Eignung von QA zur Lösung einfacher FSI-Probleme.
Darüber hinaus werden die Leistungsfähigkeit und die derzeitigen Einschränkungen aktueller QA-Hardware für derartige Anwendungen analysiert sowie Strategien zur Überwindung bestehender Limitationen untersucht. Die Erkenntnisse dieser Arbeit sollen die Grundlage für weiterführende Forschungsarbeiten zur Anwendung von QA auf gekoppelte Probleme der Strukturmechanik sowie deren Designoptimierung bilden.

List of Symbols

S	Action
A, A^s, A^f	Cross-sectional area, cross-sectional area of the piston, cross-sectional area of the chamber
Φ_i	Nodal basis function
q_i	Binary variable in QUBO
Γ, Γ^s, Γ^f, Γ^{fs}	Boundary, boundary of the structural domain, boundary of the fluid domain, interface between the structural and fluid domain
Γ^u, Γ^σ	Boundary with prescribed displacements and tractions, respectively
h_i, J_{ij}	Linear and quadratic coefficients in an Ising problem
$h_{\min}$, $J_{\min}$	Minimum linear and quadratic coefficient in an Ising problem
Q_{ij}	QUBO coefficient
ρ	Density
u_i, u_i^s, u_i^f	Displacement, displacement in the structural domain, displacement in the fluid domain
$u_{\text{con}}(x)$	Converged displacement
$\hat{u}_i$	Nodal displacement
$\bar{u}_i$	Prescribed displacement
Ω, Ω^s, Ω^f	Domain, structural domain, fluid domain
C_{ijkl}	Elasticity tensor
e_i	Element
ϵ_{con}, ϵ_{L^2}, ϵ_{H^1}	Convergence error, relative error in the L^2 norm, relative error in the H^1 norm
$\epsilon_{h,\max}$, $\epsilon_{J,\max}$	Maximum DAC quantization errors
F, $\bar{F}$	Force, prescribed force
R	Specific gas constant

g	Gravitational acceleration
H, H_i, H_p	Hamiltonian, initial Hamiltonian, problem Hamiltonian
H_{Ising}, H_{QUBO}	Hamiltonian in Ising and QUBO formulation
k	Specific heat ratio
K, K_0	Kinetic energy, kinetic energy per unit volume
L	Lagrangian function
L_0^f	Initial length of the chamber
L^s, L^f	Piston length, chamber length
m	Mass
n_j	Normal vector
N_e	Number of elements
N_q	Number of qubits per node
τ, λ, c	Offset, scaling, and representation coefficient for real number representation
$\hat{x}_i$	Nodal position
Π	Potential energy
W, W^V, W^S	Potential of external forces, potential of volume forces, potential of surface forces
W_0^V, W_0^S	Potential per unit volume due to external volume and surface forces
p	Pressure
ν	Specific volume
s_i	Spin variable
U, U_0	Strain energy, strain energy density
ϵ_{ij}	Strain tensor
σ_{ij}	Stress tensor
T	Temperature
Δt	Time interval
t	Time
t_i, t_i^s, t_i^f	Traction, structural traction, fluid traction

$\bar{t}_i$	Prescribed traction
$\mathcal{U}$	Uniform distribution
v_i	Velocity
V	Gas volume
f_i	Volume force
$\bar{x}(t)$, $\tilde{x}(t)$	Linear and quadratic approximations of the path
E	Young's modulus
*	Superscript denoting the analytical reference solution

List of Abbreviations

BESO bi-directional evolutionary structural optimization.

DAC digital-to-analog converter.

FSI fluid structure interaction.

ICE integrated control errors.

LEA least energy approximation.

PDE partial differential equation.

QA quantum annealing.

QPU quantum processing unit.

QUBO quadratic unconstrained binary optimization.

SDK software development kit.

CHAPTER 1

Introduction

1.1 Motivation

Coupled structural analysis problems are a fundamental engineering discipline, as they provide important understanding into the interactions between multiple physical fields that affect the behavior of a solid structure. One major example is fluid structure interaction (FSI), where fluid forces introduce structural deformation, which in turn affects the fluid behavior. Typical FSI applications include airplane components under aerodynamic loads, the behavior of bridges under wind forces, or blood flow in arteries. These problems are typically modeled through partial differential equations (PDEs) and solved using numerical methods. Multiple systems of equations, which depend on each other, are required to be solved, which makes them computationally challenging.
Even more complex are FSI problems combined with optimization tasks, such as size, shape, or topology optimization. These optimization tasks typically have complex design spaces with many variables, constraints, and objectives. Additionally, such problems often have many local optima, which makes it challenging for traditional methods to explore the design space and find the best overall solution.

To address these challenges, quantum annealing (QA) could have great potential. QA is a special form of quantum computing, a technology that uses the principles of quantum mechanics to perform certain computations far more efficiently than classical computers. Specifically, QA can solve optimization problems by finding the global minimum of an optimization problem quickly and effectively, even in complex solution spaces. When problems are reformulated into a form that quantum annealers can process, QA may offer enhanced computational efficiency and potentially superior solutions. These capabilities make QA an attractive approach for addressing the challenges of traditional solution methods. However, QA also has certain limitations. Existing quantum annealers are still restricted to relatively small problems that can be formulated and solved. In addition, external disturbances can introduce noise, which affects the quality of the solution.
Currently, some of the most advanced quantum annealers are from *D-Wave Systems*. To solve a problem on D-Wave QA hardware, it must be provided in a specific mathematical representation called Ising model. This model describes the problem as a quadratic objective function, where each decision variable, called a spin, can take values of $+1$ or -1. A mathematically equivalent and easily convertible formulation

is quadratic unconstrained binary optimization (QUBO), where the problem is represented as a quadratic objective function with binary variables taking values of 0 or 1. This representation is often more convenient for formulation and implementation. Therefore, problems are typically first modeled in QUBO and then converted into the equivalent Ising model to be solved on QA hardware. A detailed explanation of QA is given in Section 2.2.

In contrast to QA, FSI is a well-known engineering field with numerous established methods and applications [6, 4, 5, 3]. The complexity of optimizing FSI systems has led to various approaches for different applications. For the compliance minimization of structures subjected to design-dependent fluid pressure loads, one study [10] proposed an extended bi-directional evolutionary structural optimization (BESO) method. In the BESO method, the design variables are restricted to discrete values, resulting in solid or void regions without intermediate densities. Regarding FSI problems, this is a great benefit as it ensures a well-defined interface between the structural and fluid domain. By replacing void regions with fluid, the pressure loads can be modeled within the optimization process.
In [13], the author extended this approach by considering viscous fluid flows. The fundamental difference lies in the governing equations of the moving fluid, which are the steady-state incompressible Navier-Stokes equations.
Another study [11] introduced a topology optimization method for the frequency response of FSI systems. The objective was to minimize the amplitude of the frequency response by changing the topology of the structure. Once again, the BESO method was utilized to ensure well-defined boundaries. The proposed method can be applied to various problems such as noise reduction of aircraft or vibration control of submerged structures.

While FSI and design optimization have been extensively studied and developed, QA is a relatively new and emerging field of research. Recent studies have explored various aspects of QA.
A comprehensive review of QA, focusing on its theoretical foundations, practical implementation, and applications in traffic, logistics, machine learning, finance, and other fields, can be found in [21]. In structural mechanics, one study [17] introduced a "box algorithm" that utilized QA to solve the differential equations of an elastic bar by iteratively minimizing a discrete approximation of its potential energy in each step.
Regarding structural optimization, one work [24] investigated the optimization of two-dimensional truss member sizes to minimize a stress-based objective function. The study focused on adapting the problem for quantum annealers by employing a symbolic FEM approach. Although the approach found the global optimum for small-scale problems, the study identified challenges in scaling to larger systems.
Another study [27] took a different approach to truss structure optimization using QA, focusing on both two-dimensional and three-dimensional systems. This method represented nodal displacements as random number sums, which allows the elastic strain energy to be formulated in the QUBO format. The optimization process was performed iteratively, alternating between deformation analysis and design updates. Computations were performed on a hybrid annealer. The hybrid solver combines

classical and quantum computing, allowing for the handling of much larger problem sizes compared to a pure quantum annealer.
The work in [25] explored the use of QA for topology optimization on continuum. The authors proposed a hybrid approach that split the problem into two parts, one solved classically and the other, with reduced variables, solved on a quantum annealer. This method was designed for minimum compliance problems.
Lastly, in [28], we developed a novel formulation to address structural analysis and design optimization problems using QA. By expressing the structural problem through energy minimization principles, we transformed it into a QUBO form. The specific formulation allowed us to combine the structural analysis and design optimization problems, which are typically solved separately in an iterative process, into a single-step solution. To evaluate our approach, we analyzed a one-dimensional bar under self-weight loading, tailoring the problem's complexity to match the capabilities of currently available QA hardware. Finally, using a D-Wave quantum annealer, we demonstrated the applicability of this approach and validated its accuracy by comparing it to the analytical solution of the problem.

1.2 Problem Statement

Despite these advances in structural mechanics and design optimization, the application of QA to optimize FSI problems, or even to solve FSI problems alone, remains unexplored in current research. To enable the optimization of FSI problems using QA, it is essential to first establish how QA can be integrated into solving FSI problems themselves. This foundational step forms the motivation of this work. Thus, our aim is to develop a formulation of the FSI problem that allows the integration of QA. Specifically, a QUBO formulation has to be developed to allow the problem to be processed and solved on a D-Wave quantum annealer. Furthermore, this research seeks to deepen the understanding of the capabilities and limitations of current QA hardware, while exploring potential strategies to overcome these limitations and enhance the quality of the solution. The resulting research questions of the present study are the following:

- How can QA be incorporated into solving FSI problems, and what are the relevant considerations in developing a suitable QUBO formulation for this purpose?
- What are the capabilities and limitations of current QA hardware in the context of FSI problems and which strategies can we develop to improve the results?

1.3 Contributions

In this study, a novel method is derived and explored to incorporate QA into solving FSI problems. A QUBO formulation is created, making such problems solvable with QA hardware like D-Wave's systems. By testing this method on state-of-the-art QA hardware, the study shows how QA can be used to solve simple FSI problems.

The work also investigates the strengths and weaknesses of current QA hardware when solving these problems, providing insights transferable to other fields. As a result, an adaptive number representation method is introduced, which improves the accuracy of solutions in iterative processes.

1.4 Outline

The thesis is structured as follows: Chapter 2 provides an introduction to the fundamental concepts of FSI and QA, which form the theoretical basis of this work. In Chapter 3, a novel approach is developed to solve static FSI problems using QA. The chapter introduces the problem formulation, describes the methods used, and presents the results of various test cases. Additionally, a modification of the method is proposed to enhance solution accuracy. In Chapter 4, a formulation to address a dynamic structural problem as a basis to solve dynamic FSI problems is developed and analyzed. The limitations of the approach are also discussed. Finally, Chapter 5 summarizes the findings, discusses the constraints of the study, and outlines possible future research.

CHAPTER 2

Theoretical Background

This chapter focuses on basic concepts on which this work is built, specifically FSI and QA. In the first part, the key aspects of FSI problems are discussed, with a closer look at the relevant theory for the fluid model and the structural model. The structural model is divided into static and dynamic problem descriptions. Two energy principles, the principle of minimum potential energy and Hamilton's principle, are explained for these problem descriptions. The second part of the chapter introduces the concept of QA and key steps required to solve problems using this approach.

2.1 Fluid Structure Interaction

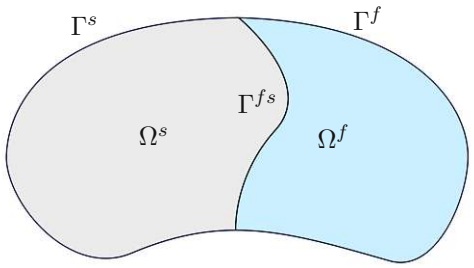


Figure 2.1: A FSI problem with the structural domain Ω^s and the fluid domain Ω^f. Each domain has its own boundary Γ^s and Γ^f, respectively, and a common interface Γ^{fs}.

As mentioned in Chapter 1, FSI problems involve the interaction between fluid and structures. Figure 2.1 shows schematically the corresponding domains: the structural domain Ω^s with its boundary Γ^s and the fluid domain Ω^f with its boundary Γ^f. The interface between these two domains is denoted by Γ^{fs}. The governing equations for the fluid are typically the Navier-Stokes equations, while the structural model is formulated on the principles of solid mechanics based on balance equations and constitutive laws. The solution process is generally numerical, utilizing discretization of these equations.

There are certain coupling conditions at the interface between the fluid and structural domains that must be satisfied [23]. The kinematic condition states the con-

tinuity of displacement across the interface:

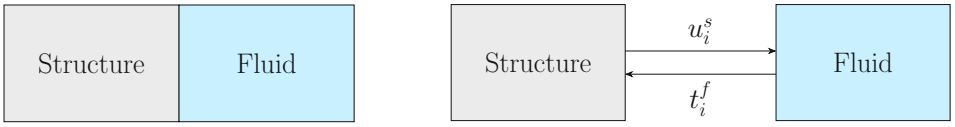
$$u_i^s = u_i^f \quad \text{on } \Gamma^{fs}. \tag{2.1}$$

Here, u_i^s and u_i^f denote the displacement field of the structure and the fluid, respectively. The dynamic condition ensures that the tractions are equal across the interface:

$$t_i^s = -t_i^f \quad \text{on } \Gamma^{fs}, \tag{2.2}$$

where t_i^s and t_i^f are the solid and fluid tractions.

The numerical methods to solve FSI problems can be categorized into two approaches: the monolithic approach and partitioned approach.
In the monolithic approach, the fluid and structural equations are combined into a single system of equations and solved simultaneously. The coupling conditions are incorporated implicitly. Figure 2.2a provides a schematic representation of this approach. This method offers advantages in terms of accuracy and stability since the coupling between the domains is handled inherently. However, it is computationally expensive, challenging to implement, and less flexible for use with existing solvers [16].
In contrast, the partitioned approach solves the fluid and structural equations separately, with information exchanged iteratively as boundary conditions at the interface to enforce the coupling conditions [23]. The type of boundary conditions exchanged between the fluid and structural solvers determines the specific partitioning scheme. The most common is the Dirichlet-Neumann scheme. Here, the structural displacement u_i^s at the interface is passed as a Dirichlet boundary condition to the fluid, and the fluid tractions t_i^f are passed as a Neumann condition to the structure. This procedure is illustrated in Figure 2.2b. Fixed-point methods are typically employed for the solution process [12]. While this method may lack accuracy and stability as errors can accumulate during the iterative process [16], it provides more modularity. This modularity allows the use of specialized solvers for each domain, making it easier to implement and adapt to different problems [12].



(a) Monolithic approach. **(b)** Partitioned approach (Dirichlet-Neumann).

Figure 2.2: A schematic representation of the monolithic and the partitioned approach for solving FSI problems.

In this study, we employ the partitioned approach for solving FSI problems, as we can use its modularity to treat the fluid and structural model separately. To adapt the problem complexity to current QA hardware, we integrate QA only into the structural subproblem. For this purpose, we reformulate the structural subproblem differently from conventional solvers, using the system's energy as the basis. By utilizing energy principles, we represent the structural analysis as an optimization problem, which can then be reformulated into the QUBO format required for QA.

In the following, we provide the relevant theoretical background for both the fluid and structural model of the FSI problem. Since the fluid and structural models are addressed separately, the subscripts s and f are dropped to simplify the notation.

2.1.1 Fluid Model

As already mentioned, the behavior of fluid is governed by the Navier-Stokes equations, which describe the conservation of mass, momentum, and energy in a fluid. A comprehensive treatment of these equations can be found in standard fluid mechanics textbooks like [19]. In this thesis, we focus on a highly simplified fluid model solely based on the ideal gas law, which establishes the relationship between pressure, volume, and temperature for an ideal gas. The ideal gas law is expressed as [18]:

$$p\nu = RT, \tag{2.3}$$

where p is the pressure, $\nu = \frac{1}{\rho}$ is the specific volume (the reciprocal of density ρ), R is the specific gas constant for a particular gas, and T is the temperature.
With additional assumptions introduced in Section 3.1.1, this law allows us to compute the pressure, which defines the Neumann boundary condition for the structure.

This simplified fluid model is sufficient for our study, as its primary purpose is to enable the coupling with the structural subproblem. Since our focus lies on the structural formulation and its adaptation for QA, a detailed fluid model is not required. However, because of the modularity of the partitioned approach, the fluid model could be replaced with a more complex one if needed, without affecting the structural formulation or the integration of QA.

2.1.2 Structural Model

For the structural model, we use energy principles to provide a form suitable for QA. To achieve this, we must distinguish between two cases: for static systems, we can apply the principle of minimum potential energy, while for dynamic systems, we can use Hamilton's principle.

Principle of Minimum Potential Energy

The principle of minimum potential energy is a fundamental concept, which can be derived from the principle of virtual work for the special case of elastic bodies [14]. It can be used to determine the static equilibrium configuration of a structure under external loads. A detailed description of the principle can be found in textbooks [15, 9, 14], which also serve as the basis for the following description.

To formulate the principle, we consider an elastic body Ω, illustrated in Figure 2.3. The principle is stated with respect to the displacement field of the body u_i. A volume force density f_i may act on the entire domain. The boundary of the body is denoted by Γ, where surface tractions $\bar{t}_i$ and displacements $\bar{u}_i$ are prescribed on the portions Γ^σ and Γ^u, respectively.

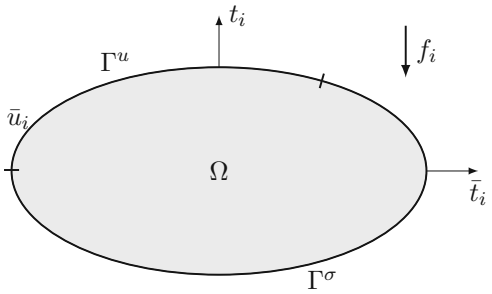


Figure 2.3: An elastic body Ω with the boundary portions Γ^u, where displacements $\bar{u}_i$ are prescribed and a boundary portion Γ^σ, where surface traction $\bar{t}_i$ are prescribed. The force f_i is a volume force density.

We assume that the body is elastic, which implies the existence of a strain energy density function U_0. Furthermore, we only consider the case of small displacements, which implies geometric linearity. The strain energy density is related to the stress tensor σ_{ij} and the strain tensor ϵ_{ij} by the following expression:

$$\sigma_{ij} = \frac{\partial U_0}{\partial \epsilon_{ij}}, \tag{2.4}$$

where U_0 is a function of the strain components ϵ_{ij}.

For external forces, we assume that they are conservative, which means that they can be derived from corresponding potentials. These include the potential of the volume forces, denoted as W_0^V, and the potential of the surface forces, denoted as W_0^S. The forces can then be expressed as:

$$f = -\frac{\partial W_0^V}{\partial u_i}, \quad t_i = -\frac{\partial W_0^S}{\partial u_i}. \tag{2.5}$$

For a linear elastic body, the strain energy density U_0 can be written as:

$$U_0 = \frac{1}{2} C_{ijkl} \epsilon_{ij} \epsilon_{kl}, \tag{2.6}$$

where C_{ijkl} is the elasticity tensor, which describes the material's response to deformation.

For external volume and surface forces, which do not depend on the displacement of the body they act on, the potential can be described as

$$W_0^V = -f_i u_i, \quad W_0^S = -\bar{t}_i u_i. \tag{2.7}$$

The total strain energy, the potential of the volume forces, and the potential of the surface forces can now be defined as follows:

$$U = \int_\Omega U_0 \, \mathrm{d}\Omega, \quad W^V = \int_\Omega W_0^V \, \mathrm{d}\Omega, \quad W^S = \int_{\Gamma^\sigma} W_0^S \, \mathrm{d}\Gamma^\sigma. \tag{2.8}$$

The total potential of external forces W is the sum of the potential contributions from the volume forces and the surface forces:

$$W = W^V + W^S. \tag{2.9}$$

With the total strain energy and the total potential of external forces,we are now able to define the total potential energy of the system as the sum of these two contributions:

$$\Pi = U + W. \tag{2.10}$$

Before stating the principle of minimum potential energy, it is necessary to restrict our consideration to only kinematically admissible displacement fields u_i. A displacement field u_i is kinematically admissible, if it is continuously differentiable and it satisfies the displacement boundary conditions:

$$u_i = \bar{u}_i \quad \text{on } \Gamma^u. \tag{2.11}$$

Based on the principle of virtual work, it can be shown that the total potential energy Π attains a stationary value for the true displacement field. Mathematically, this condition is expressed as:

$$\delta\Pi = 0, \tag{2.12}$$

where $\delta\Pi$ represents the first variation of the total potential energy.

Due to our assumption of geometric linearity, the condition

$$\delta^2\Pi > 0 \tag{2.13}$$

holds. Together with Equation (2.12), this represents the necessary and sufficient condition for Π to become a minimum, which directly leads to the principle of minimum potential energy.

In summary, the principle of minimum potential energy states that among all kinematically admissible displacement fields, the equilibrium configuration of an elastic body minimizes the total potential energy of the system.

Therefore, we can formulate the structural analysis problem as a minimization problem:

$$\min_{u_i \in \boldsymbol{U}} \{\Pi[u_i]\}, \tag{2.14}$$

with $\boldsymbol{U}$ representing the set of all kinematically admissible displacement fields. The solution $u_{i,\text{opt}}$ is the equilibrium state of the elastic body.

Hamilton's Principle

The principle of minimum potential energy is limited to the static equilibrium of solids. Hamilton's principle is more general and applicable to dynamic systems. In this context, the displacement u_i of a body becomes time-dependent, $u_i = u_i(t)$, and is often referred to as a path. A detailed description of the principle can be found in textbook [14, 20], which also serve as the basis for the following description.

For an elastic body subjected to conservative external forces, we recall from previous discussions the strain energy U and the potential energy of external forces W. The sum of these two components gives the total potential energy Π of the system. In dynamic systems, an additional energy contribution arises, the kinetic energy. For small displacements, the density ρ can be considered constant, and the kinetic energy per unit volume is given by

$$K_0 = \frac{1}{2}\rho v_i v_i, \tag{2.15}$$

where $v_i = \frac{\partial u_i}{\partial t}$ represents the velocity. Consequently, the total kinetic energy of the body is

$$K = \int_\Omega K_0 \, \mathrm{d}\Omega. \tag{2.16}$$

To formulate Hamilton's principle, it is necessary to restrict consideration to only admissible paths u_i. This means that the u_i must satisfy the displacement boundary conditions, as in the static case. Additionally, the configuration of the system at the initial time t_0 and the final time t_1 of the considered time interval must be known. Consequently, the variation δu_i vanishes at these time points:

$$\delta u_i(t_0) = \delta u_i(t_1) = 0. \tag{2.17}$$

Now, Hamilton's principle states that the difference between the kinetic energy K and the potential energy Π, integrated over time from t_0 to t_1, attains a stationary value for the true path of motion. This is expressed mathematically as:

$$\delta \int_{t_0}^{t_1} (K - \Pi) \, \mathrm{d}t = 0. \tag{2.18}$$

It is worth noting that for a static system, the kinetic energy becomes zero and Hamilton's principle reduces to the principle of minimum potential energy.

The integrand in Equation (2.18) is known as the Lagrangian function L, defined as:

$$L = K - \Pi. \tag{2.19}$$

Using this definition, the action S is defined as the time integral of the Lagrangian:

$$S = \int_{t_0}^{t_1} L \, \mathrm{d}t. \tag{2.20}$$

This leads to a compact formulation of Hamilton's principle, stating that the action S, defined as the time integral of the Lagrangian, is stationary for the actual path of the system:

$$\delta S = 0. \tag{2.21}$$

Note that the action S does not necessarily attain a minimum for the true path of the system, but rather an minimum or a saddle point (and never a maximum) [8]. However, for sufficiently short paths, the action S is always a minimum for the true path. Further insights regarding the conditions under which a path is considered sufficiently short are provided in [8].

Hamilton's principle and the principle of minimum potential energy provide a way to formulate structural problems as optimization problems, which is a requirement for solving them using QA. In the following chapter, the fundamentals of QA are introduced to provide a theoretical foundation.

2.2 Quantum Annealing

Quantum computing is an emerging field in computer science. It uses quantum mechanical phenomena, such as superposition, to address problems that can be very demanding for classical computers [22]. There are different approaches for quantum computing, the most famous one is gate-based quantum computing. In this approach, specially developed quantum algorithms, like Shor's algorithm [2] or Grover's algorithm [1], can be applied to solve specific problems much faster than classical algorithms.
Another approach to quantum computing is QA, which is especially useful for solving combinatorial optimization problems, where the goal is to find the best solution among many possibilities. These problems are often NP-hard, meaning they are very hard, or even impossible to solve efficiently with classical methods [22].
One of the most popular QA systems is built by *D-Wave Systems*. Since this thesis utilizes a quantum annealer from D-Wave, we will focus on the theoretical foundations specific to this system. A detailed description on the theoretical background of QA can be found on the official D-Wave website [31]. The following section will provide an overview of the fundamental principles and outlines the key steps required to solve mathematical problems using QA.

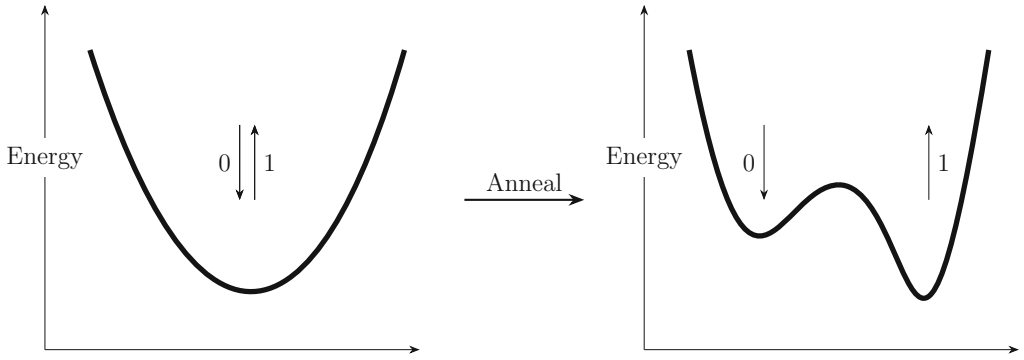


Figure 2.4: The evolution of an energy potential for a single qubit during the annealing process, starting in a superposition state (↓↑) and ending in a classical state, either spin-down (↓), corresponding to the binary variable 0, or spin-up (↑), corresponding to the binary variable 1. In this illustration, the qubit would end up in the spin-up (↑) state.

A fundamental component of QA is a quantum bit, or *qubit*. A quantum annealer consists of many qubits, each of which can be seen as the analogue of a classical bit. Unlike classical bits, which can only exist in one of two discrete states (0 or 1), a qubit can also exist in a superposition of both states simultaneously. There are several ways to create qubits, one of which is superconducting qubits, as used by D-Wave. In this approach, qubits are formed by superconducting loops of wire, where

the state of the qubit is determined by the direction of the circulating current with a corresponding magnetic field. The two classical states correspond to spin-down (↓) and spin-up (↑). During the annealing process, the qubit starts in a superposition state. By the end of the process, the qubit collapses into a classical state, either the spin-up (↑) or spin-down (↓) state. The probability of the qubit collapsing into one of these states can be controlled by applying an external magnetic field to the qubit. The external magnetic field is called a *bias*. The physics of this process can be represented in an energy diagram, as shown in Figure 2.4. By applying the external magnetic field, the energy valley of the preferred state is lowered, increasing the probability that the qubit will collapse into this state. Referring to Figure 2.4, this would be the spin-up (↑) state.

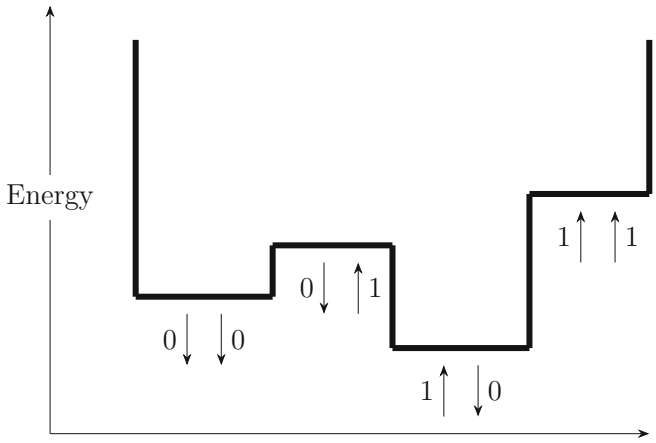


Figure 2.5: Energy landscape of two coupled qubits showing four possible states, with the lowest-energy state (↑, ↓) representing the optimal solution.

A quantum processing unit (QPU), in which we encode the problem to be solved, is composed of many qubits. For instance, D-Wave's Advantage system contains over 5000 qubits. A key aspect of QA lies in the ability to link these qubits, allowing them to influence each other using physical devices called *couplers*. Couplers create correlations between qubits, enabling them to end up in either the same classical state (both (↓) or both (↑)) or opposite states ((↓) and (↑)). These correlations, as well as individual qubit biases, are user-controlled and define the problem to be solved on a quantum annealer. Figure 2.5 illustrates the energy diagram for two coupled qubits, which form an energy landscape. For two qubits, there are four possible states: (↓, ↓), (↓, ↑), (↑, ↓), and (↑, ↑). The energy of each state is determined by the biases of the individual qubits and the couplers between them. During the annealing process, the qubits collapse into the lowest energy state, which in this example is the (↑, ↓) state.

As previously mentioned, a QPU can consist of thousands of qubits, which allows us to define the highly complex energy landscape representing the problem of interest. The goal of the QA process is to find the lowest-energy state, which represents the optimal solution of the problem being solved. This process is the foundation of energy optimization in QA.

Another fundamental term in QA is the *Hamiltonian*. The Hamiltonian can be seen as the mathematical description of the energy state. It is the sum of the initial Hamiltonian H_i and the final Hamiltonian, or problem Hamiltonian H_p. The annealing process begins in the lowest-energy state, also called *ground state*, of the initial Hamiltonian H_i, which can be easily prepared and represents a superposition of all possible states, as previously described. During the annealing process, H_i is gradually replaced by H_p, which represents the specific optimization problem using the qubit biases and couplers. The evolution of the Hamiltonian is expressed as:

$$H(t) = A(t)H_i + B(t)H_p \tag{2.22}$$

where $A(t)$ decreases from 1 to 0 and $B(t)$ increases from 0 to 1 over time. At the start, the system is in the ground state of H_i. By the end, the system ideally transitions into the ground state of H_p, which corresponds to the solution to the optimization problem. It is important to note that, in practice, thermal fluctuations and other phenomena can cause the system to jump from the ground state to a higher energy state. This may result in a solution that is not optimal. To improve the chances of finding the true ground state and obtaining the optimal solution, the annealing process must be repeated multiple times.

The problem Hamiltonian H_p can be mathematically formulated using either the Ising model or the QUBO model. These two models are equivalent and can be easily converted into one another. They describe how qubit biases and couplers, as previously introduced, define the energy landscape. In the Ising model, the spin variables $s_i \in \{-1, +1\}$ represent the spin-down and spin-up states of the qubits. The problem is expressed as:

$$H_{\text{Ising}} = \sum_i h_i s_i + \sum_{i<j} J_{ij} s_i s_j \tag{2.23}$$

where the linear coefficients h_i represent the qubit biases and the quadratic coefficients J_{ij} represents the coupling strength between the variables s_i and s_j.

In the QUBO model, the two classical states of the qubits are represented by binary variables $q_i \in \{0, 1\}$ and the problem is expressed as:

$$H_{\text{QUBO}} = \sum_i Q_{ii} q_i + \sum_{i<j} Q_{ij} q_i q_j \tag{2.24}$$

where Q is a matrix containing the linear terms Q_{ii} (biases) and the quadratic terms Q_{ij} (couplings).

The mathematical model of a problem can be represented as a *logical graph*, where the variables s_i or q_i are illustrated as nodes. The linear coefficients associated with these variables are included as weights on the nodes, while their interactions, which are weighted by the quadratic coefficients, are the edges of the graph.

In addition to the logical graph, there is the *hardware graph*, which represents the physical arrangement of qubits and couplers in a QPU. An example of a hardware

graph is illustrated in Figure 2.6, showing a section of the *Pegasus* architecture used in D-Wave's Advantage QPUs. In this graph, qubits are represented as green dots, and couplers are shown as gray lines. The blue squares highlight unit cells, which are the basic building blocks of the hardware graph. An Advantage QPU consists of a 16x16 grid of such unit cells.

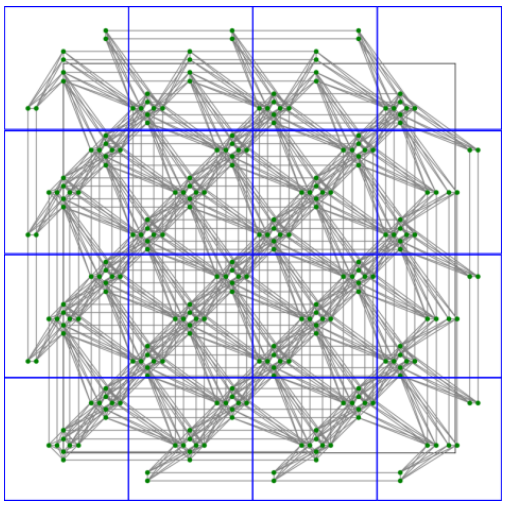


Figure 2.6: A section of the Pegasus hardware graph architecture used in D-Wave's Advantage QPUs. Image source: [33].

To solve a problem on a QPU, the logical graph, which represents the mathematical model of the problem, must be mapped onto the hardware graph. This process is called *minor embedding* and is necessary because the logical graph often has higher connectivity than the hardware graph can support. As can be seen in Figure 2.6, each qubit is connected to a limited number of neighboring qubits. In minor embedding, multiple physical qubits are grouped to represent a single logical qubit. One such group of qubits is called *chain*. In a chain, all qubits are required to be in the same state, that is the state of the logical qubit. Couplers with high negative weights need to be used to achieve this. The strength of these couplers is called *chain strength*.

In summary, the process of solving problems using QA can be outlined as follows:

1) *Definition of the Problem*

The optimization problem is represented mathematically using the Ising model or QUBO model. The problem can be visualized as a logical graph with variables as nodes and their relationships as edges, weighted by linear and quadratic coefficients.

2) *Minor-embedding*

The logical graph is mapped onto the QPU's hardware graph through minor embedding, using chains of physical qubits to preserve the correct interactions between the logical variables.

3) *Programming*

The bias for each qubit and the coupling strength between qubits are configured on the QPU.

4) *Initialization*

The system starts in the ground state of the initial Hamiltonian H_i, representing a superposition of all possible solutions.

5) *Annealing Process*

The Hamiltonian evolves from H_i to the problem Hamiltonian H_p by gradually adjusting the coefficients $A(t)$ and $B(t)$ over time.

6) *Readout of the Solution*

At the end of the annealing process, the system collapses into a final state, which represents a possible solution to the optimization problem. The state of each qubit is mapped back to the corresponding node in the logical graph.

7) *Resampling*

The annealing process is repeated multiple times to increase the possibility of finding the true ground state and achieving the optimal solution.

CHAPTER 3

Static Problem

In this chapter, we focus on the static FSI problem. First, we introduce both the structural and fluid model based on a specific one-dimensional (1D) problem. A key focus is the formulation of the structural subproblem as a QUBO problem, making it suitable for QA. To validate our method, we test it on a D-Wave quantum annealer, analyzing the iterative solution process and comparing the binary and random real number representations. We also investigate the sources of inaccuracies that appear in some solutions. To address these errors, we introduce an adaptive number representation and validate its effectiveness in improving accuracy.

3.1 Methods for the Static Problem

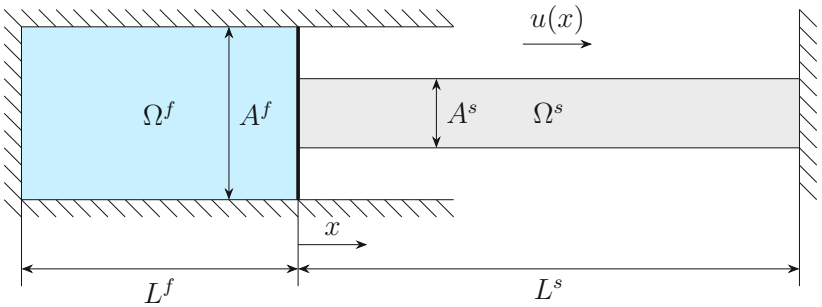


Figure 3.1: Schematic representation of the 1D piston problem consisting of the elastic structural domain Ω^s and the gas-filled chamber Ω^f.

For the static FSI problem, the so-called *piston problem* serves as a basis. This problem involves the interaction between a linearly elastic 1D piston, which represents the structural component, and a 1D chamber filled with gas, which represents the fluid domain. An illustration of the piston problem is provided in Figure 3.1.

The piston rod, with a length L^s and a cross-sectional area A^s, has an unknown displacement field $u(x)$. The chamber, which has a cross-sectional area A^f and a length L^f, contains an ideal gas at unknown pressure p. Based on the theory presented in Chapter 2, the piston forms the structural domain Ω^s with the boundary Γ^s, where a displacement boundary condition is applied on the right end Γ^u. The

gas-filled chamber forms the fluid domain Ω^f with its boundary Γ^f. The piston head is assumed to be a thin rigid body, forming the interface Γ^{fs} between Ω^s and Ω^f.

Solving the problem, i.e., finding the equilibrium displacement field of the piston, requires the coupling of two models:

- A fluid model, which takes the position of the piston head $u(0)$ as input and computes the fluid pressure p in the chamber.
- A structural model, which uses the force acting on the piston head, derived directly from the fluid pressure p, as input. The output of this model is the displacement field $u(x)$.

To find the equilibrium configuration, a partitioned approach is employed. As described in Chapter 2, in this approach, the fluid and structural models are solved independently and iteratively exchange information at the interface. Specifically, the fluid model provides the pressure based on the piston head displacement, while the structural model yields the displacement field based on the acting force derived from the fluid pressure. The problem is considered as solved once the difference between $u(x)$ of the current iteration and the previous iteration is small enough.

3.1.1 Fluid Model

In this study, the fluid model is kept simple. As a mathematical description of the model, the ideal state gas law (Equation (2.3)) is used. It is assumed that the pressure within the chamber is homogeneous and directly depends on the position of the piston head. Additionally, adiabatic conditions are assumed, meaning there is no heat exchange between the chamber and the external environment. The process is also assumed to be reversible, implying no permanent changes occur within or outside of the system. A reversible, adiabatic process is also known as isentropic. For an isentropic process, where the system moves from state 0 to state 1, the following relationship holds [18]:

$$p_1(V_1)^k = p_0(V_0)^k. \tag{3.1}$$

Here, p_0 and p_1 represent the gas pressure in state 0 and 1, respectively, while V_0 and V_1 denote the corresponding gas volume. Additionally, k is the specific heat ratio of the gas. For diatomic gases, k is typically assumed to be 1.4, which will also be used in this study.

Equation (3.1) can be rearranged to calculate the pressure at state 1:

$$p_1 = p_0 \left(\frac{V_0}{V_1} \right)^k \tag{3.2}$$

In the problem illustrated in Figure 3.1, the gas volume V is given by $V = A^f L^f$. Assuming that state 0 corresponds to the initial state ($u(x) = 0$) and state 1 corresponds to the desired state after the piston head has moved ($u(x) \neq 0$), the length

of the chamber L^f can be written as

$$L^f = L_0^f + u(0), \tag{3.3}$$

where L_0^f is the length of the chamber at state 0. Therefore, Equation (3.2) becomes

$$p_1 = p_0 \left(\frac{L_0^f}{L_0^f + u(0)} \right)^k. \tag{3.4}$$

As can be seen from Equation (3.4), in order to calculate the unknown gas pressure p_1, the displacement $u(0)$ must be provided. This is the coupling to the structural model where the displacement $u(x)$ is computed.

3.1.2 Structural Model

To determine the unknown displacement of the piston caused by the gas pressure force, the static structural model is formulated using the principle of minimum potential energy. To solve this problem using QA, the model must be reformulated into a QUBO form. The explanation below outlines the derivation of this QUBO form.

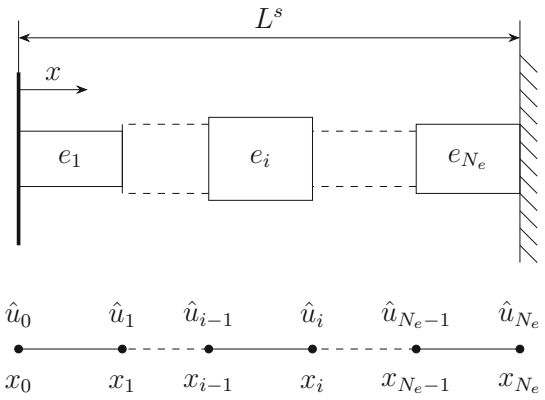


Figure 3.2: The piston rod composed into multiple elements e_i.

First, the piston rod is discretized into multiple elements, as shown in Figure 3.2. When using only identical elements, this discretization is not strictly necessary for a simple linear-elastic 1D problem. However, the discretization is performed to ensure the method could be applied to more complex problems where discretization is essential. Additionally, this approach allows varying the problem size by changing the number of elements.

The piston is divided into N_e elements $e_i = e_1, \ldots, e_{N_e}$, each spanning from x_{i-1} to x_i. The displacements at the element boundaries are treated as unknown nodal displacements $\hat{u}_i$, resulting in $N_e + 1$ nodes. The zero-displacement boundary condition on the fixed end of the piston is directly considered as $\hat{u}_{N_e} = 0$. We define

the displacement $u(x)$ as follows:

$$u(x) = \sum_{i=0}^{N_e} \hat{u}_i \Phi_i(x), \tag{3.5}$$

where $\Phi_i(x)$ are nodal interpolation functions. In this work, linear interpolation functions are chosen for simplicity:

$$\Phi_i(x) = \begin{cases} 0, & x < x_{i-1}, \\ \frac{x - x_{i-1}}{x_i - x_{i-1}}, & x_{i-1} \leq x < x_i, \\ \frac{x - x_{i+1}}{x_i - x_{i+1}}, & x_i \leq x < x_{i+1}, \\ 0, & x \geq x_{i+1}. \end{cases} \tag{3.6}$$

For the discrete model, the strain energy of an individual element e_i is given by:

$$U_i = \frac{1}{2} \int_{x_{i-1}}^{x_i} E_i A_i^s \left(u'(x)\right)^2 \, \mathrm{d}x, \tag{3.7}$$

where A_i^s and E_i are the cross-sectional area and Young's modulus of the element, respectively, and $u'(x)$ is the spatial derivative of the displacement, which is obtained as:

$$u'(x) = \sum_{i=0}^{N_e} \hat{u}_i \Phi_i'(x), \tag{3.8}$$

where $\Phi_i'(x) = \frac{d\Phi_i(x)}{dx}$ represents the derivative of the interpolation functions with respect to x. The total strain energy of the rod is determined by summing the contributions of all elements:

$$U = \sum_{i=1}^{N_e} U_i. \tag{3.9}$$

To compute the total potential energy, the potential energy of external forces W must also be considered. This consists of contributions from volume forces W^V and surface forces W^S. For the piston problem, volume forces are not considered ($W^V = 0$).

The surface forces result from the pressure p in the chamber acting on the piston head. For the 1D piston, this fluid pressure leads to an external force $\bar{F} = pA^f$, which acts on the first node of the piston u_0, representing the piston head. W^S is given by:

$$W^S = -\bar{F} u_0. \tag{3.10}$$

The total potential energy of the piston system is the difference between the strain energy and the potential energy of the surface forces:

$$\Pi = \sum_{i=1}^{N_e} U_i - \bar{F} u_0. \tag{3.11}$$

Since the QUBO problem requires binary variables, each real-valued nodal displacement $\hat{u}_i$ must be represented using a set of binary variables $q_{i,l} \in \{0, 1\}$. There are

multiple ways to represent a real-valued number using binary variables. In general, the mapping of a real-valued variable to binary variables can be expressed as follows [26]:

$$\hat{u}_i = \tau_i + \lambda_i \sum_{l=0}^{N_q-1} c_{i,l} q_{i,l}, \tag{3.12}$$

where N_q is the number of binary variables used to represent one real-valued variable $\hat{u}_i$. The coefficients τ_i and λ_i are real-valued scaling parameters that define the range of representable real-valued variables. The coefficient $c_{i,l}$ specifies the representation method for mapping the real-valued variable.
In this study, two representation methods are used and compared in Section 3.2: the classical binary representation (referred to as the *binary representation*) and a method involving random numbers (referred to as the *random representation*).

For the binary representation, $c_{i,l}$ is defined as:

$$c_{i,l} = 2^l, \tag{3.13}$$

while for the random representation, $c_{i,l}$ is defined as:

$$c_{i,l} \sim \mathcal{U}(0,1), \tag{3.14}$$

where $\mathcal{U}(0,1)$ denotes a uniform distribution in the range $[0,1]$.

Using Equation (3.12), the real-valued variables $\hat{u}_i$ are expressed in terms of binary variables, which is required for constructing the QUBO form. By collecting all binary variables $q_{i,l}$ into a vector $\boldsymbol{q}$, the structural problem can be written in QUBO form as:

$$\min_{\boldsymbol{q}} \{\Pi[\boldsymbol{q}]\}. \tag{3.15}$$

To provide a better understanding into the derivation of the QUBO form, a simple example is given below.

Let us consider a single element without boundary conditions or external forces. An illustration of this element is shown in Figure 3.3. The element spans from $x = 0$ to $x = 1$ and has two nodes with displacement variables $\hat{u}_0$ and $\hat{u}_1$.

Figure 3.3: One element with two nodes.

The displacement field of the element is defined using Equation (3.5) with

linear interpolation functions as defined in Equation (3.6):

$$u(x) = \sum_{i=0}^{N_e} \hat{u}_i \Phi_i(x) \tag{3.16}$$
$$= \hat{u}_0 \Phi_0(x) + \hat{u}_1 \Phi_1(x) \tag{3.17}$$
$$= \hat{u}_0(1-x) + \hat{u}_1 x. \tag{3.18}$$

The derivative with respect to x is:

$$u'(x) = \hat{u}_1 - \hat{u}_0. \tag{3.19}$$

With the derivative of the displacement field defined, we can use Equation (3.7) to derive the strain energy as a function of the nodal displacements. For simplicity, let us assume $E = A = 1$:

$$U = \frac{1}{2} \int_0^1 (\hat{u}_1 - \hat{u}_0)^2 \, \mathrm{d}x \tag{3.20}$$
$$= \frac{1}{2}\hat{u}_0^2 - \hat{u}_0\hat{u}_1 + \frac{1}{2}\hat{u}_1^2. \tag{3.21}$$

Since no external forces are applied, the potential energy of external forces W is zero. Therefore, the total potential energy is:

$$\Pi = U = \frac{1}{2}\hat{u}_0^2 - \hat{u}_0\hat{u}_1 + \frac{1}{2}\hat{u}_1^2. \tag{3.22}$$

To derive the QUBO form, the real-valued nodal displacement variables $\hat{u}_i$ must be represented using binary variables $q_{i,l} \in \{0, 1\}$ based on one of the representation methods discussed earlier. In this example, the binary representation method with $c_{i,l}$ defined in Equation (3.13) is used. Each nodal value is represented using two binary variables ($N_q = 2$). To represent real numbers in the range $[0, 1]$, the scaling parameters τ_i and λ_i are set as:

$$\tau_i = 0, \quad \lambda_i = \frac{1}{2^{N_q} - 1} = \frac{1}{3}. \tag{3.23}$$

Using Equation (3.12), the real-valued variables are mapped to binary variables as follows:

$$\hat{u}_i = 0 + \frac{1}{3} \sum_{l=0}^{1} 2^l q_{i,l} \tag{3.24}$$
$$= \frac{1}{3} q_{i,0} + \frac{2}{3} q_{i,1} \tag{3.25}$$

With this number representation, the total potential energy in terms of the binary variables becomes

$$\Pi[q_{i,l}] = \frac{1}{18}q_{0,0}^2 + \frac{2}{9}q_{0,0}q_{0,1} + \frac{2}{9}q_{0,1}^2 - \frac{1}{9}q_{0,0}q_{1,0} - \frac{2}{9}q_{0,0}q_{1,1} - \frac{2}{9}q_{0,1}q_{1,0} - \frac{4}{9}q_{0,1}q_{1,1} + \frac{1}{18}q_{1,0}^2 + \frac{2}{9}q_{1,0}q_{1,1} + \frac{2}{9}q_{1,1}^2. \tag{3.26}$$

To better understand the structure of $\Pi[q_{i,l}]$, we rewrite Equation (3.26) in matrix form. We define a vector $\boldsymbol{q}$ containing the binary variables:

$$\boldsymbol{q} = \begin{bmatrix} q_{0,0} \\ q_{0,1} \\ q_{1,0} \\ q_{1,1} \end{bmatrix} \tag{3.27}$$

Now, Equation (3.26) can be expressed as:

$$\Pi[\boldsymbol{q}] = \boldsymbol{q}^T \underbrace{\begin{bmatrix} \frac{1}{18} & \frac{2}{9} & -\frac{1}{9} & -\frac{2}{9} \\ 0 & \frac{2}{9} & -\frac{2}{9} & -\frac{4}{9} \\ 0 & 0 & \frac{1}{18} & \frac{2}{9} \\ 0 & 0 & 0 & \frac{2}{9} \end{bmatrix}}_{\boldsymbol{Q}} \boldsymbol{q}. \tag{3.28}$$

The upper diagonal matrix $\boldsymbol{Q}$ is called QUBO matrix and defines the problem. Equation (3.28) represents the QUBO model introduced in Equation (2.24), where the diagonal elements Q_{ii} are the linear coefficients (biases) and the nonzero off-diagonal elements Q_{ij} are the quadratic coefficients (couplings). The logical graph of this example is illustrated in Figure 3.4, with the blue nodes being the biases and the orange edges being the couplings. The left side of the graph corresponds to the first node, while the right side corresponds to the second node in Figure 3.3.

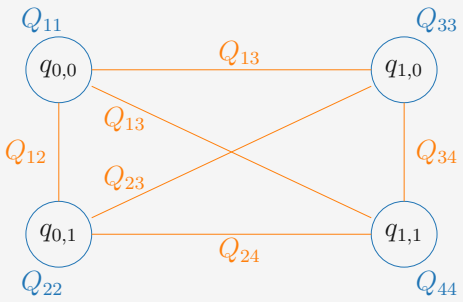


Figure 3.4: The logical graph of the example element.

As the potential energy in Equation (3.28) is now in QUBO form, the minimization problem (Equation (3.15)) can be solved using QA. In this specific example, only the trivial solution $u(x) = 0$ exists, as there are no forces or boundary conditions applied.

3.2 Results for the Static Problem

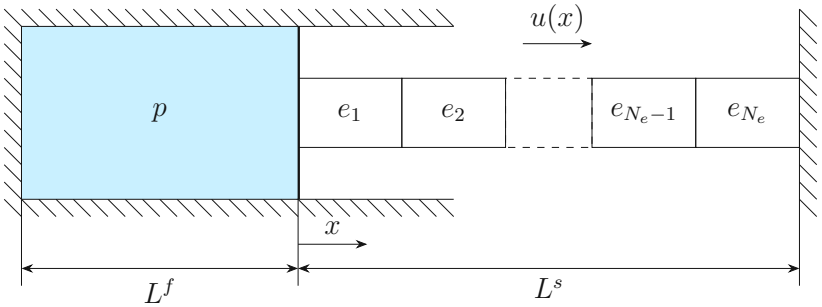


Figure 3.5: The piston problem with a composed rod.

In this section, the results for the static piston problem are presented and analyzed. Various test cases are examined to evaluate the capabilities and limitations of the proposed QA-based method for solving FSI problems. If not stated otherwise, the following physical parameters for the piston problem, which is illustrated in Figure 3.5, are considered. The piston rod has a length $L^s = 1.0$ with identical cross-sectional areas $A_i^s = A^s = 1.0$ and Young's moduli $E_i = E = 1.0$. The gas-filled chamber with a cross-sectional area $A^f = 2.0$ has an initial length $L_0^f = 1.0$. The gas has a specific heat ratio $k = 1.4$ and an initial pressure $p_0 = 0.5$. The values of these quantities are summarized in Table 3.1. Other settings, like the number of elements or the number of qubits per node vary for each test case and will therefore be stated individually.

Table 3.1: Physical parameters for the piston problem.

L^s	A^s	E	L_0^f	A^f	k	p_0
1.0	1.0	1.0	1.0	2.0	1.4	0.5

The results were obtained using a custom code based on the open-source software *EngiOptiQA* [35], which we have developed for solving structural and design optimization problems of a 1D rod. The formulation of the QUBO model was implemented with the *Fixstars Amplify* software development kit (SDK) [34] (amplify 0.11.2), while the solution process was carried out using *D-Wave's Ocean* SDK) [29] (`dwave-cloud-client 0.11.1`, `dwave-sampler 1.2.0`, `dwave-system 1.21.0`). All computations were executed on the `Advantage_system6.4` machine, which features the *Advantage performance update* QPU. As described in Section 2.2, this QPU is based on the Pegasus architecture illustrated in Figure 2.6. The QPU consists of $5,760$ qubits interconnected by $40,088$ couplers [30].

SP1: Iterative Solution Process and Error Criteria

First, we investigate the iterative solution process and validate the QA solutions with the analytical ones. Additionally, we analyze different error measures to evaluate

the accuracy of the results in the following experiments.

In test case *static problem 1* (SP1), the piston rod is divided into $N_e = 2$ elements. Each nodal value is represented using $N_q = 8$ qubits in the binary representation. To represent the real-valued nodal displacements in the range $[0, 1]$, the scaling parameters τ_i and λ_i are set to $\tau_i = 0$ and $\lambda_i = \frac{1}{2^{N_q}-1} = \frac{1}{255}$. The chosen values for this test case are summarized in Table 3.2.

Table 3.2: Settings for SP1.

N_e	N_q	Rep. Range	Rep. Method
2	8	$[0, 1]$	binary

To solve the coupled piston problem, we employ a fixed-point approach, where the fluid and the structural problem are solved iteratively until convergence is reached. This process is illustrated in Figure 3.6.
At each iteration:

- The fluid solver computes the pressure p inside the chamber based on the displacement u_0 of the first node (i.e., the displacement of the piston head).
- Using the computed pressure p, the resulting force $\bar{F}$ acting on the piston is determined and passed to the structural solver as a boundary condition.
- The structural solver computes the updated displacement field $u(x)$ using QA, which is then passed to the fluid solver to close the coupling loop.

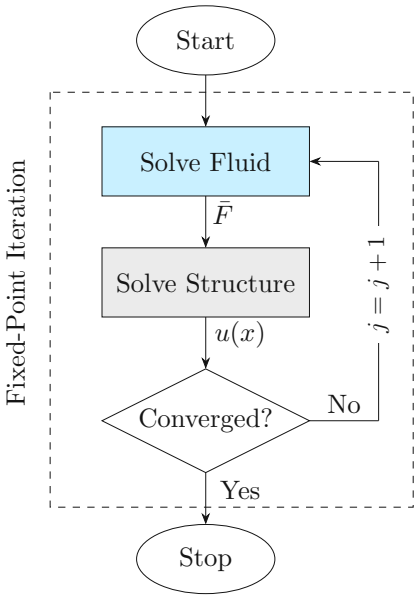
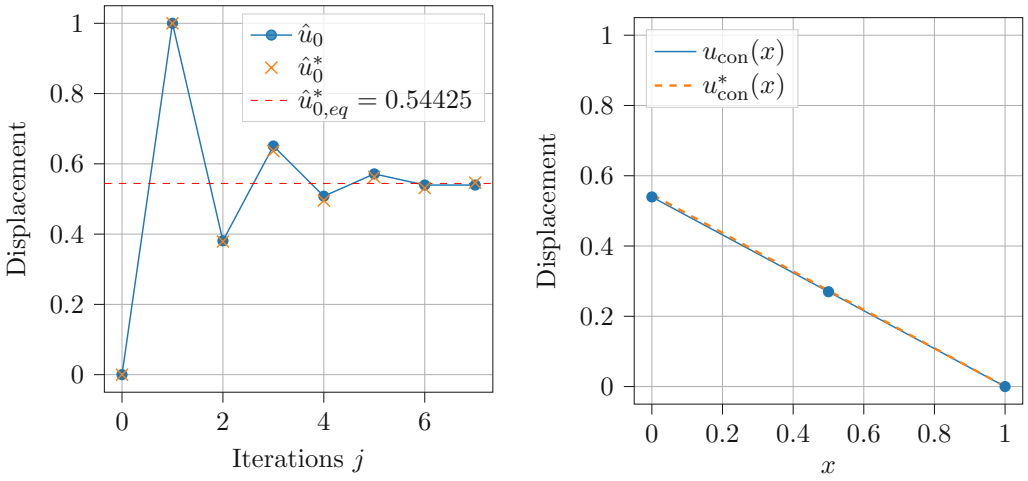


Figure 3.6: Illustration of the iterative fixed-point solution process.

The iterative coupling process continues until the solution is considered converged. To determine whether convergence is reached, we evaluate the relative change in the nodal displacement vector $\hat{\boldsymbol{u}}$, which contains the displacements $\hat{u}_i$ of all nodes. Convergence is achieved when the difference between two successive iterations falls below a predefined threshold:

$$\frac{\|\hat{\boldsymbol{u}}^{(j+1)} - \hat{\boldsymbol{u}}^{(j)}\|}{\|\hat{\boldsymbol{u}}^{(j+1)}\|} < \epsilon_{\text{con}}. \tag{3.29}$$

Here, j denotes the iteration index, and ϵ_{con} represents the predefined threshold, which is set to $\epsilon_{\text{con}} = 10^{-2}$.



(a) Nodal displacement $\hat{u}_0$ over iterations.

(b) Displacement function $u_{\text{con}}(x)$.

Figure 3.7: Results of test case SP1.

The results of SP1 are shown in Figure 3.7. The iterative solution process is visualized in Figure 3.7a by showing the nodal displacement $\hat{u}_0$ of the piston head over the iterations. Additionally, in each iteration the analytical solution of the displacement of the piston head $\hat{u}_0^*$ is indicated as orange cross. This solution is based on the fluid pressure of the current iteration and can be calculated with basic mechanical equations. As can be seen, the QA solution $\hat{u}_0$ closely matches $\hat{u}_0^*$ in every iteration. After seven iterations, the convergence criteria (Equation (3.29)) is fulfilled, and the solution process comes to an end. This means the equilibrium position of the piston is considered found. For comparison, the red dashed line indicates the analytical solution for the equilibrium position of the piston head $\hat{u}_{0,eq}^*$. This value is obtained by equating the force caused by the fluid pressure p with the corresponding counteracting force of the elastic piston. The converged QA solution is close to the actual equilibrium position $\hat{u}_{0,eq}^*$, confirming the validity of the method for accurately solving this problem.

Figure 3.7b shows the displacement function $u_{\text{con}}(x)$ of the converged QA solution, i.e., from the final iteration. The blue dots indicate the nodal displacements of the

piston. As can be observed, the QA solution is visually very close to the analytical solution $u^*_{\text{con}}(x)$. The error between the converged QA solution $u_{\text{con}}(x)$ and the corresponding analytical solution $u^*_{\text{con}}(x)$ will be a key metric in the following test cases, evaluating the solution accuracy. Therefore, an appropriate error measure for comparing these two solutions must be selected.

We consider two error norms: the L^2 norm and the H^1 norm. The relative error in the L^2 norm is given by:

$$\epsilon_{L^2} = \frac{\|u_{\text{con}}(x) - u^*_{\text{con}}(x)\|_{L^2}}{\|u^*_{\text{con}}(x)\|_{L^2}} = \frac{\left(\int_\Omega \left(u_{\text{con}}(x) - u^*_{\text{con}}(x)\right)^2 dx\right)^{1/2}}{\left(\int_\Omega \left(u^*_{\text{con}}(x)\right)^2 dx\right)^{1/2}}. \tag{3.30}$$

The second is the relative error in the H^1 norm, defined as:

$$\epsilon_{H^1} = \frac{\|u_{\text{con}}(x) - u^*_{\text{con}}(x)\|_{H^1}}{\|u^*_{\text{con}}(x)\|_{H^1}} = \frac{\left(\int_\Omega \left(u_{\text{con}}(x) - u^*_{\text{con}}(x)\right)^2 + \left(u_{\text{con}}{}'(x) - u^*_{\text{con}}{}'(x)\right)^2 dx\right)^{1/2}}{\left(\int_\Omega \left(u^*_{\text{con}}(x)\right)^2 + \left(u^*_{\text{con}}{}'(x)\right)^2 dx\right)^{1/2}}. \tag{3.31}$$

This norm considers not only the difference in the displacement values $u_{\text{con}}(x)$ but also the difference in their derivatives $u_{\text{con}}{}'(x)$.

The error measure we choose should correspond with the objective function, i.e., the potential energy, of the minimization problem. To evaluate this using our test case, we can take advantage of the fact that we do not have just one result for the converged solution but several, because the annealing process is executed multiple times to increase the chances of finding the true optimum. In this test case, the annealing process was repeated 500 times, resulting in 500 samples. Each sample can have a different solution, with high-quality outcomes characterized by solutions that cluster near the true minimum.

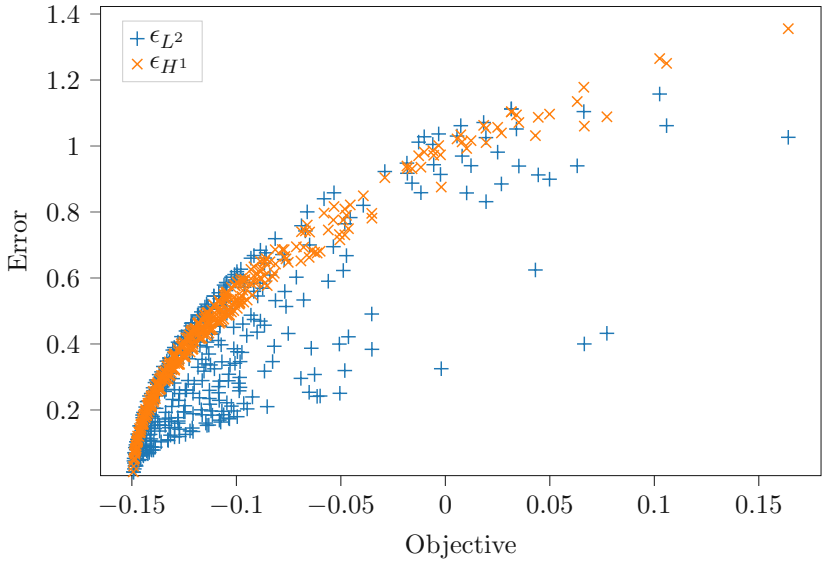


Figure 3.8: Comparison of L^2 norm error ϵ_{L^2} and H^1 norm error ϵ_{H^1} plotted against the objective function values for 500 annealing samples.

In Figure 3.8, the error of each sample relative to the analytical solution against the objective function values can be seen. Both the L^2 norm error ϵ_{L^2} and the H^1 norm error ϵ_{H^1} are compared. From the plot, it is clear that ϵ_{H^1} correlates more closely with the objective function. This behavior can also be explained mathematically. The H^1 norm includes both $u_{\text{con}}(x)$ and ${u_{\text{con}}}'(x)$, just as the potential energy of the problem depends on the displacement $u_{\text{con}}(x)$ (through external forces) and its gradient ${u_{\text{con}}}'(x)$ (through the strain energy). As a result, minimizing the objective function naturally minimizes the error in the H^1 norm. For this reason, the H^1 norm error ϵ_{H^1} is chosen as the accuracy metric in the following investigations.

SP2: Real Number Representation

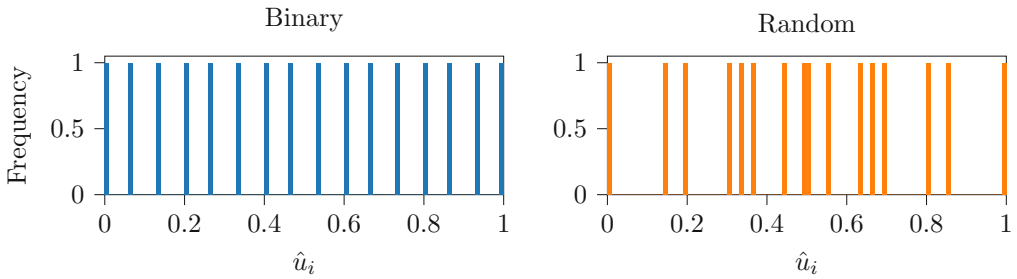
Next, we will compare the two real number representations introduced in Section 3.1.2, i.e., the binary representation and the random representation, to determine which one performs better. To test performance across a wide range of scenarios, we conduct multiple tests with different parameters.

Specifically, we vary the number of elements, i.e., the degrees of freedom, $N_e \in \{1, 2, 3, 4\}$, which influences the problem size, and the number of qubits per real number $N_q \in \{2, 4, 8, 16\}$, which affects both the representation precision of the real number, i.e., the nodal displacement, and the problem size. Due to the randomness in the random representation and the probabilistic nature of QA, eight independent trials are conducted for each combination of parameters, leading to $2 \times 4 \times 4 \times 8 = 256$ experiments. Table 3.3 summarizes the parameters.

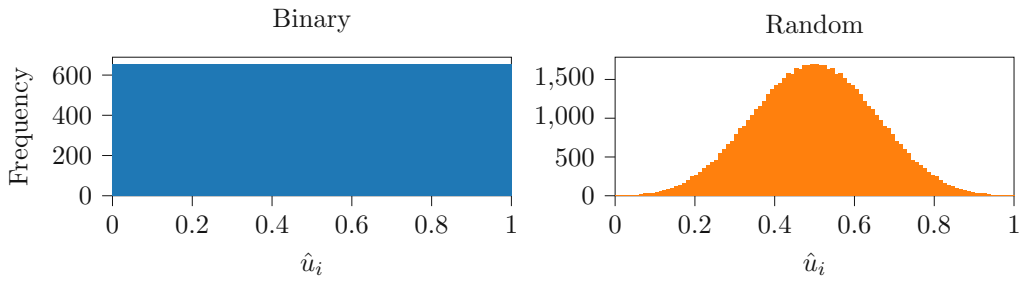
Table 3.3: Settings for SP2.

N_e	N_q	Rep. Range	Rep. Method	Trials
$\{1, 2, 3, 4\}$	$\{2, 4, 8, 16\}$	$[0, 1]$	{binary, random}	8

Before analyzing performance, we first illustrate the representable real-valued numbers for two specific cases, $N_q = 4$ and $N_q = 16$, to highlight key differences between the two representations. For $N_q = 4$ (Figure 3.9a), the binary representation provides evenly distributed and non-redundant representable numbers. In contrast, the random representation results in randomly distributed representable numbers.
For $N_q = 16$ (Figure 3.9b), the binary representation remains evenly distributed, while the random representation shows over-representation in the central region. This bias arises because the sum of uniformly distributed random numbers is taken in Equation (3.12). If the real-valued variable is not necessarily near the middle of the range, this bias becomes a disadvantage, as numbers in the outer regions cannot be represented accurately. However, the random representation has its benefits because it has properties that can lead to more accurate results in certain optimization problems [26, 7].



(a) Binary and random representation ($N_q = 4$).

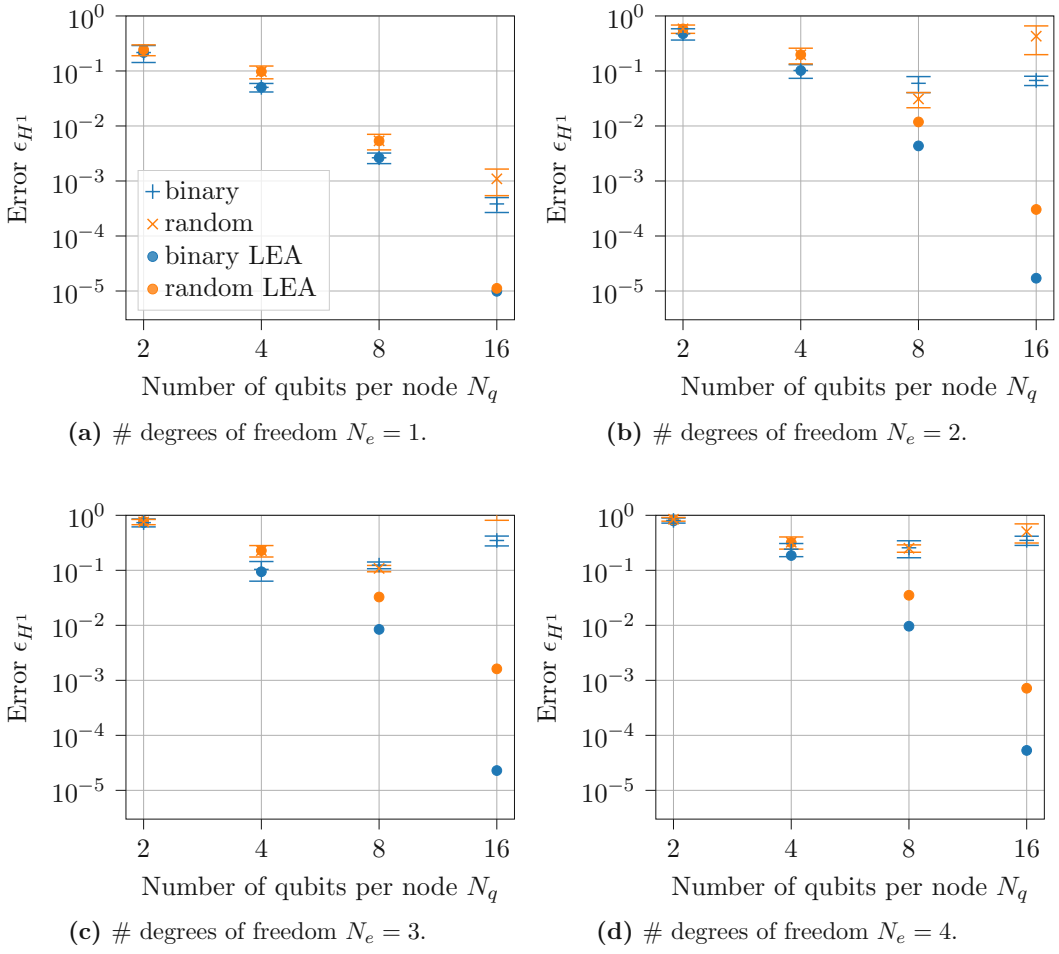


(b) Binary and random representation ($N_q = 16$).

Figure 3.9: The representable real numbers using the binary and random representations in the range $[0, 1]$ for two different values of N_q.

Since the random representation is biased toward the middle of the representation range and the binary representation has a structured distribution pattern, we randomize the initial fluid pressure $p_0 \sim \mathcal{U}(0.1, 0.5)$ in each trial to prevent unintended biases. This ensures that the converged solution does not always fall in the same part of the representation range. As a result, it prevents the bias of the random representation from affecting the findings and avoids cases where numbers in the binary representation end up in particularly favorable or unfavorable positions.

To compare the solution accuracy of both representations, we analyze the error ϵ_{H^1}, as defined in Equation (3.31), by averaging it over the 8 trials for each parameter combination. Figure 3.10 displays these results, where the vertical axis represents the average error. The horizontal axis corresponds to the number of qubits per node N_q. The error bars represent the standard error of each result. The four subplots vary in the number of degrees of freedom N_q.



(a) # degrees of freedom $N_e = 1$.

(b) # degrees of freedom $N_e = 2$.

(c) # degrees of freedom $N_e = 3$.

(d) # degrees of freedom $N_e = 4$.

Figure 3.10: Comparison of the binary and random representations for different numbers of degrees of freedom N_e and qubits per real number N_q. The plots show the average error of the QA solutions across 8 trials for each parameter combination. Blue and orange dots represent the ϵ_{H^1} error between the least energy approximation (LEA) and the analytical solution for the binary and random representations, respectively.

Various trends can be observed from these four subplots. First, the solution accuracy between the binary and random representations can be compared. Although individual experimental results are quite similar, the binary representation generally yields smaller errors. This suggests that for this type of problem, the binary representation is the better choice.

A second observation can be made from Figure 3.10a. As the number of qubits per node N_q increases, the error decreases. This is understandable, as a greater number of binary variables representing a real number allows for more precision.
However, this trend is only partly visible in Figure 3.10b. Up to $N_q = 8$, the error decreases, but at $N_q = 16$, the error is greater than at $N_q = 8$. This suggests that despite the theoretical increase in precision, other inaccuracies degrade the results. Another observation is that the result worsens for fixed N_q, e.g., $N_q = 2$, as the

number of degrees of freedom N_e increases, as seen in the 4 subplots.

To examine this situation more closely, the theoretically best possible solution was calculated for each experiment based on the given precision, which depends on N_q and the representation method. This solution is the approximation with the lowest possible objective value, i.e., the lowest potential energy, and is referred to as the *least energy approximation* (LEA). In Figure 3.10, the ϵ_{H^1} error between the LEA and the analytical solution for the binary and random representations is shown for each experiment as blue and orange dots, respectively.
In Figure 3.10a, for $N_q \in \{2, 4, 8\}$, the error of the QA solutions match the one of the LEA, indicating the best possible solution has been found. For $N_q = 16$, the error decreases, but a gap appears compared to the error of the LEA. This means the increased precision still provides some advantage but cannot be fully exploited. When increasing the problem size to $N_e = 2$ (Figure 3.10b), the error of the LEA cannot be reached even at $N_q = 8$. For even more complex problems (Figures 3.10c and 3.10d), the gap between the errors of the QA solution and LEA becomes larger.

These inaccuracies are apparently due to the annealing process, as increasing complexity with the use of more qubits causes solutions to deviate from the theoretically optimal solution due to various error sources, which will be examined next.

SP3: Error Sources

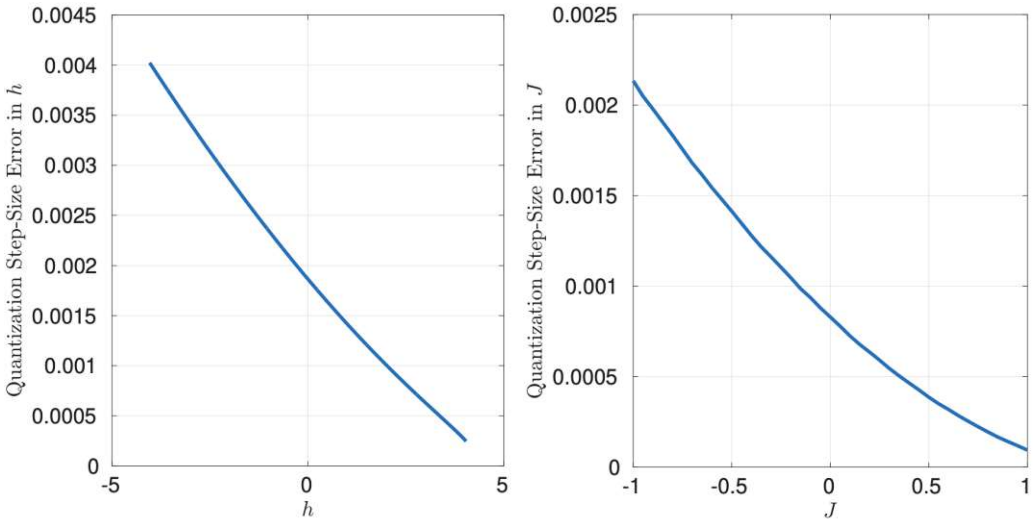
During a QA process, various sources of errors can occur. One significant category of these errors is known as integrated control errors (ICE) [32]. These errors arise from the issue to accurately represent the problem on the QPU in practice. While the QPU solves problems expressed in the Ising model (Equation (2.23)), the actual implementation on the QPU can involve small deviations in the h and J values, resulting in a slightly altered Hamiltonian:

$$H^{\delta}_{\text{Ising}} = \sum_i (h_i + \delta h_i) s_i + \sum_{i<j} (J_{ij} + \delta J_{ij}) s_i s_j, \tag{3.32}$$

where δh_i and δJ_{ij} represent deviations from the intended h_i and J_{ij} values, respectively.

Because δh_i and δJ_{ij} are summed over the number of qubits Equation (3.32), these errors can have a larger impact on results for problems involving many qubits. Several factors contribute to ICE, with dominant components including background susceptibility, flux noise in qubits, I/O system effects, and digital-to-analog converter (DAC) quantization errors [32]. DAC quantization errors arise when user-defined h and J values are converted into an analog signal for the QPU. Since the Ising spins on the QPU are controlled by magnetic fields, as described in Section 2.2, this conversion is necessary but introduces small inaccuracies.

While ICE consists of multiple sources of error, DAC quantization errors can be quantitatively analyzed using available data from D-Wave for specific QPUs, such as the `Advantage_system6.4`. Figure 3.11 illustrates typical DAC quantization errors for h and J values on the `Advantage_system6.4` QPU. For example, at $h = -4.0$, the error can be as large as ± 0.004.



(a) Quantization error in h. **(b)** Quantization error in J.

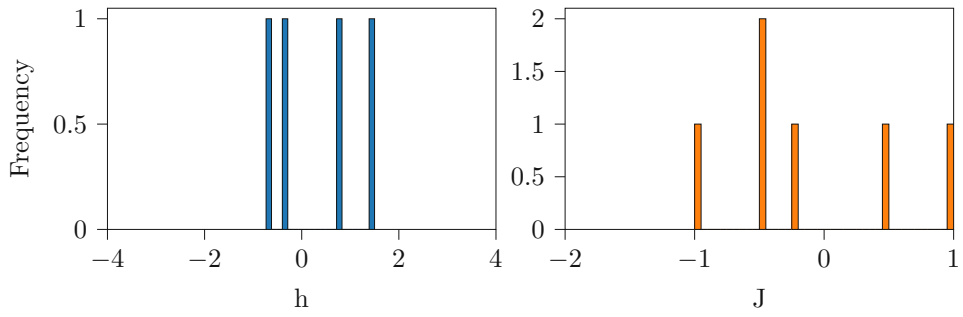
Figure 3.11: Typical DAC quantization errors of the `Advantage_system6.4` QPU. Image source: [30].

To investigate the impact of the DAC errors, we conduct a test case SP3. Similar to SP2, we perform multiple experiments with a different number of qubits per node N_q. This time, we exclusively use the binary representation method and fix the number of degrees of freedom to $N_e = 2$. The parameters are summarized in Table 3.4.

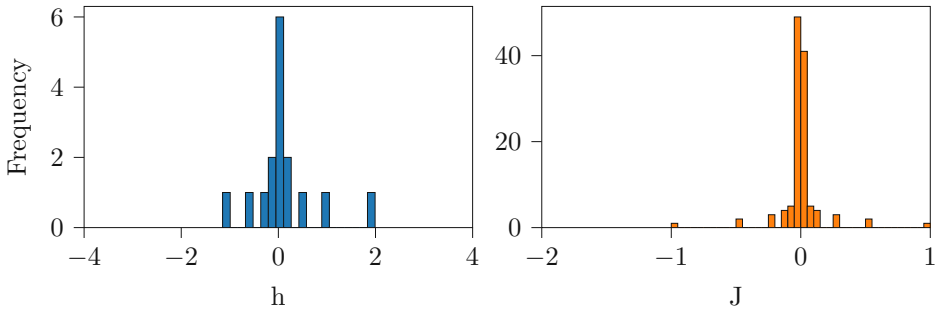
Table 3.4: Settings for SP3.

N_e	N_q	Rep. Range	Rep. Method	Trials	p_0
2	$\{2, 4, 5, 6, 8, 12\}$	$[0, 1]$	binary	8	$\mathcal{U}(0.1, 0.5)$

Before analyzing the results, we must discuss an important procedure called *auto-scaling*, which is done for every problem submitted to the QPU. Each QPU has a supported range of values for the biases and coupling strengths, i.e, the h and J coefficients. Before a specific problem is solved on a QPU, all user-defined coefficients are scaled up or down by a certain factor to fit the entire available range. For the `Advantage_system6.4` QPU, the supported ranges are $h \in [-4.0, 4.0]$ and $J \in [-2.0, 1.0]$.
This effect is illustrated in 3.12, which shows the scaled h and J values of two specific trials of SP3 with $N_q = 2$ qubits per node (Figure 3.12a) and $N_q = 8$ qubits per node (Figure 3.12b). The auto-scaling ensures that all coefficients are within the supported range. In these two cases, the largest J values are mapped to the J range limit of 1. Since the same scaling factor is applied to all coefficients, the negative J range limit and the h range cannot be fully utilized.



(a) Number of qubits per node $N_q = 2$.

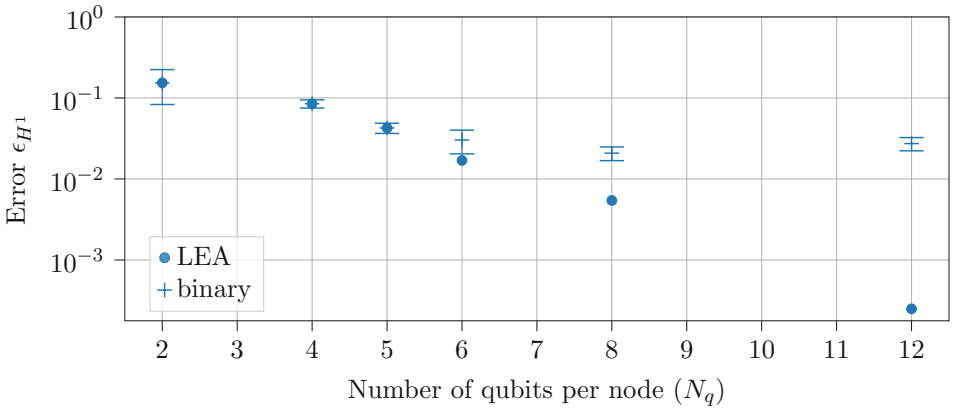


(b) Number of qubits per node $N_q = 8$.

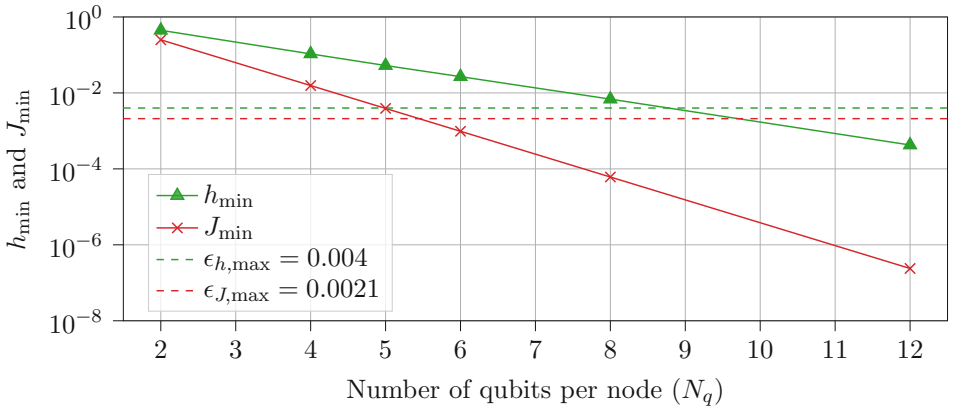
Figure 3.12: Scaled h and J values for two specific trials of SP3 with $N_q = 2$ and $N_q = 8$. In both trials, the initial gas pressure were $p_0 = 0.5$.

Comparing these two cases, it can be observed that when using $N_q = 8$ qubits per node, the resulting absolute values of the coefficients tend to be significantly smaller than when using $N_q = 2$ qubits per node. This is an important observation regarding the DAC errors, as this error is relatively larger for smaller coefficients. While increasing N_q improves precision in the real number representation, it also leads to smaller h and J values, which in turn increases the impact of DAC-induced deviations. This provides a possible explanation for why increasing the number of qubits per node does not always improve accuracy, as previously seen in SP2.

To further investigate this observation, we now discuss the results of SP3, where we analyze the impact of the DAC errors on the solution quality of our tests in greater detail. To achieve this, we compute ϵ_{H^1}, as defined in Equation (3.31), for different solutions that vary in the number of qubits per node N_q. Additionally, we compute the ϵ_{H^1} error between the LEA and the analytical solution to determine when the QA solution no longer matches the LEA, as we did in SP2. To draw conclusions about these deviations in relation to the DAC errors, we also determine the smallest absolute values of our coefficients, h_{min} and J_{min} obtained in our experiments and compare them with the magnitude of the DAC errors.



(a) Errors ϵ_{H^1} of the QA solution (binary) and the LEA for different number of qubits per node N_q.



(b) Minimum h and J coefficients in comparison to the maximum possible DAC quantization errors for different number of qubits per node N_q.

Figure 3.13: Results of SP3.

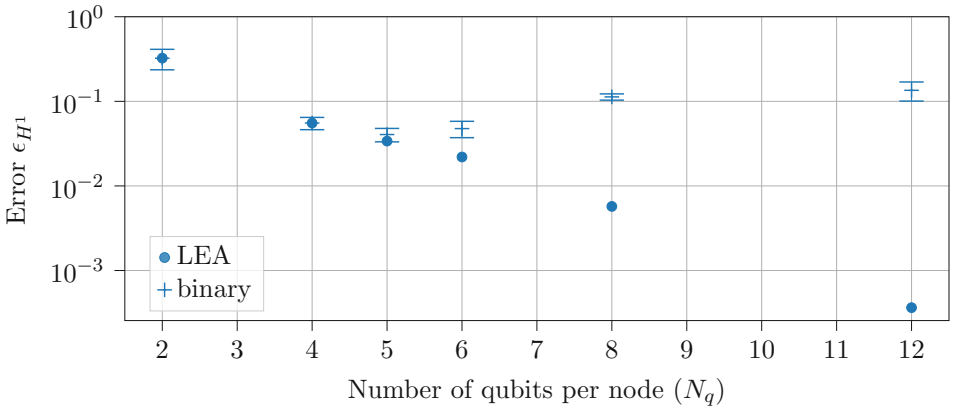
In Figure 3.13a, the error ϵ_{H^1} of the QA solution, as defined in Equation (3.31), averaged over the 8 trials, is shown. Additionally, the ϵ_{H^1} error of the LEA is presented for comparison. Both quantities are shown for different numbers of qubits per node N_q. Similar to the previous test case, for $N_q \in 2, 4, 5$, the QA solution matches the LEA. However, for $N_q \geq 6$, the LEA can no longer be reached, and a gap emerges between the errors of the two solutions. This gap grows larger as N_q increases.

Figure 3.13b displays the minimum coefficients h_{min} and J_{min} across the experiments. These values decrease with increasing N_q, confirming the observations before. The dashed lines in the second subplot represent the maximum DAC quantization errors $\epsilon_{h,\max}$ and $\epsilon_{J,\max}$ for the `Advantage_system6.4` QPU.

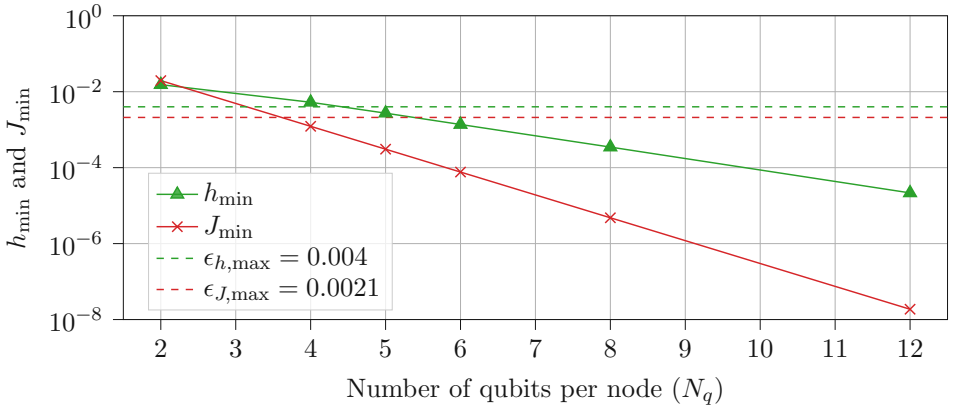
It can be observed that for $N_q \leq 5$, h_{min} and J_{min} remain above $\epsilon_{h,\max}$ and $\epsilon_{J,\max}$,

respectively. For $N_q \geq 6$, J_{min} falls below the DAC quantization error $\epsilon_{J,\max}$. At exactly $N_q = 6$, the QA solution can no longer match the LEA. This suggests that the inability to reach the LEA is at least partly due to the DAC quantization error.

To further test this hypothesis, we aim to decrease h_{min} and J_{min} in the problem setup and investigate how this affects the solution accuracy. To achieve this, a modified test is conducted where all parameters remained the same as before, except that the Young's moduli of the elements were set to $E_1 = 1.0$ and $E_2 = 5.0$, instead of being uniform. This results in reduced h_{min} and J_{min}. The results of this test are shown in Figure 3.14.



(a) Errors ϵ_{H^1} of the QA solution (binary) and the LEA for different number of qubits per node N_q.



(b) Minimum h and J coefficients in comparison to the maximum possible DAC quantization errors for different number of qubits per node N_q.

Figure 3.14: Results of SP3 with different Young's moduli of the elements. They were set to $E_1 = 1.0$ and $E_2 = 5.0$, instead of being uniform.

As seen in Figure 3.14b, the coefficients h_{min} and J_{min} decreased further, crossing the horizontal dashed lines at earlier N_q values. Figure 3.14a shows that the LEA

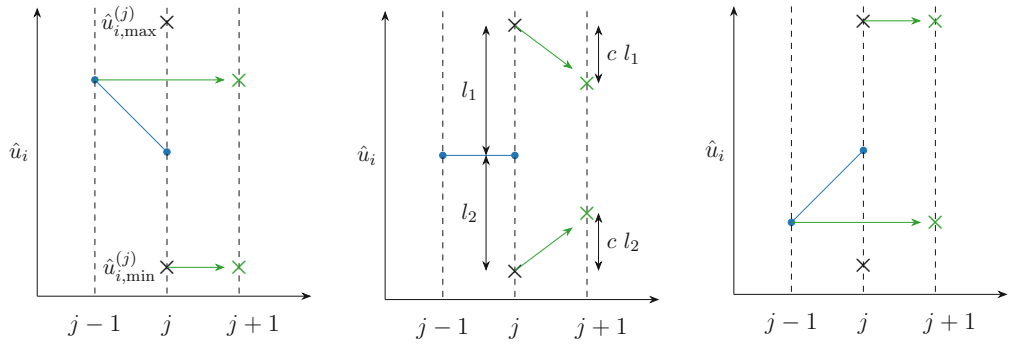
is no longer reached for $N_q \geq 5$, which means that the inaccuracy in the solution occurs at an earlier stage compared to the previous test. This example reinforces the finding that smaller h and J coefficients lead to a bigger impact of the DAC quantization errors.

Additionally, as shown in Figure 3.13a and Figure 3.14a, the error increases with N_q once the LEA can no longer be achieved. This might be explained by Equation (3.32), where the errors δh_i and δJ_{ij} accumulate. As N_q increases, the number of qubits also increases, amplifying the cumulative error and reducing the accuracy of the solution.

3.3 Adaptive Number Representation

As discussed in SP2, the quality of the solution generally decreases as the problem size increases. Moreover, in SP3 we observed that the solution cannot be arbitrarily improved by simply increasing the precision of the number representation, because this also leads to smaller h and J values, which in turn increases the impact of DAC-induced deviations.

To address this issue, we developed an adaptive number representation that progressively improves the accuracy of the solution, in line with the iterative nature of the partitioned approach. The key aspect of this method is that we use a variable representation range for the real numbers, unlike the previously fixed range of $[0, 1]$. In each iteration, the range limits for the next iteration are adjusted based on the solutions obtained from the current and previous iterations. This method relies on the assumption that the coupling progressively contracts toward convergence in each iteration.
For each real-valued nodal variable $\hat{u}_i$, the limits of the representable range are adjusted according to one of three cases, which are illustrated in Figure 3.15. In these sketches, the blue dots represent the current solution $\hat{u}_i^{(j)}$ and the previous solution $\hat{u}_i^{(j-1)}$. The black crosses indicate the upper and lower limits for the current iteration ($\hat{u}_{i,\max}^{(j)}$ and $\hat{u}_{i,\min}^{(j)}$), while the green crosses show the updated limits for the next iteration ($\hat{u}_{i,\max}^{(j+1)}$ and $\hat{u}_{i,\min}^{(j+1)}$).



(a) Case 1: $\hat{u}_i^{(j)} < \hat{u}_i^{(j-1)}$. The upper limit is updated to $\hat{u}_i^{(j-1)}$, and the lower limit remains unchanged.

(b) Case 2: $\hat{u}_i^{(j)} = \hat{u}_i^{(j-1)}$. Both limits are adjusted based on a factor c to refine the range around the current solution.

(c) Case 3: $\hat{u}_i^{(j)} > \hat{u}_i^{(j-1)}$. The lower limit is updated to $\hat{u}_i^{(j-1)}$, and the upper limit remains unchanged.

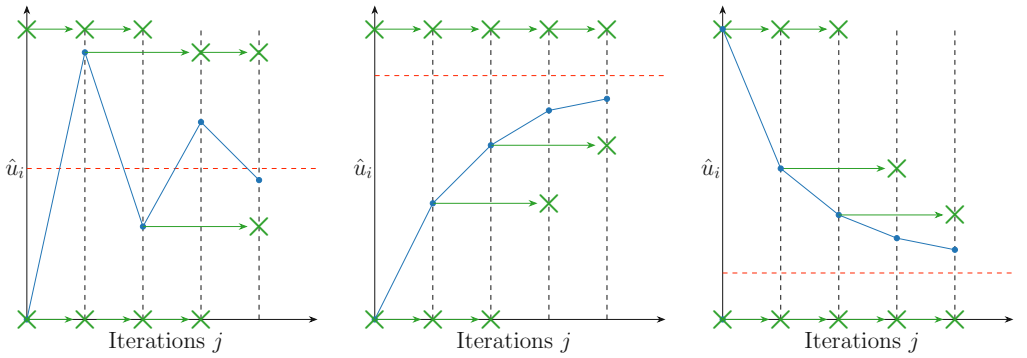
Figure 3.15: Principle of the adaptive number representation for updating the limits of the representation range.

Mathematically, the adjustment of the upper and lower limit can be written as:

$$\hat{u}_{i,\max}^{(j+1)} = \begin{cases} \hat{u}_i^{(j-1)}, & \hat{u}_i^{(j)} < \hat{u}_i^{(j-1)}, \\ \hat{u}_{i,\max}^{(j)} - c(\hat{u}_{i,\max}^{(j)} - \hat{u}_i^{(j)}), & \hat{u}_i^{(j)} = \hat{u}_i^{(j-1)}, \\ \hat{u}_{i,\max}^{(j)}, & \hat{u}_i^{(j)} > \hat{u}_i^{(j-1)}. \end{cases} \tag{3.33}$$

$$\hat{u}_{i,\min}^{(j+1)} = \begin{cases} \hat{u}_{i,\min}^{(j)}, & \hat{u}_i^{(j)} < \hat{u}_i^{(j-1)}, \\ \hat{u}_{i,\min}^{(j)} - c(\hat{u}_{i,\min}^{(j)} - \hat{u}_i^{(j)}), & \hat{u}_i^{(j)} = \hat{u}_i^{(j-1)}, \\ \hat{u}_i^{(j-1)}, & \hat{u}_i^{(j)} > \hat{u}_i^{(j-1)}. \end{cases} \tag{3.34}$$

Using the adaptive number representation, three different solution scenarios can be addressed, which are illustrated in Figure 3.16. The solution is expected to converge to the desired value, indicated by the red dashed line in the diagrams. The three scenarios include an oscillatory solution (Figure 3.16a), a monotonically increasing solution (Figure 3.16b), and a monotonically decreasing solution (Figure 3.16c).
For solutions that do not fulfill the underlying assumption of convergence and contraction, the adaptive number representation can be extended with an additional safety mechanism. If the current solution reaches either the upper or lower limit of the representation range, the corresponding limit can be adjusted to expand the representation range in that direction. The iteration is then repeated, ensuring that the solutions always remain within the representation range.



(a) Oscillatory solution. **(b)** Monotonically increasing solution. **(c)** Monotonically decreasing solution.

Figure 3.16: Adaptive number representation for three different scenarios.

SP4: Adaptive Number Representation vs. Fixed Representation

To analyze the effectiveness of the adaptive number representation in improving solution accuracy, we compare it against a fixed representation in this test case.

For this test, the number of elements was set to $N_e = 2$, and the number of qubits per node was set to $N_q = 8$. We conducted experiments for two scenarios: one with a fixed representation range of $[0, 1]$ and another using the adaptive representation. The relevant test parameters are provided in Table 3.5.

Table 3.5: Settings for SP4.

N_e	N_q	Rep. Range	Rep. Method	Trials	p_0
2	8	{fixed, adaptive}	binary	8	$\mathcal{U}(0.1, 0.5)$

For each representation method, we perform eight independent trials and evaluate the error ϵ_{H^1}, as defined in Equation (3.31), at each iteration. The average error across the trials is analyzed to compare the performance of both representation methods over the iterations.

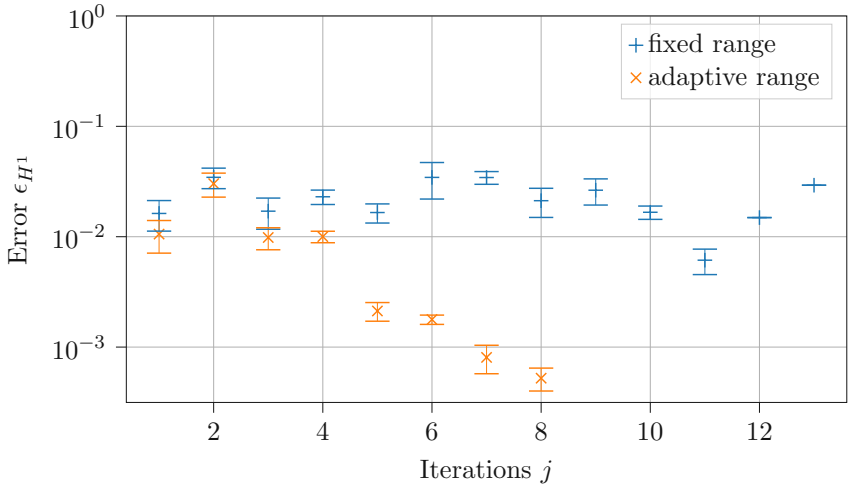


Figure 3.17: Comparing fixed and adaptive range methods with $N_q = 8$.

As seen in Figure 3.17, the solution accuracy for the fixed range case remains similar across all iterations. In contrast, for the adaptive number representation, the solution accuracy improves significantly over the iterations, demonstrating the effectiveness of the adaptive number representation. Employing the adaptive number representation in our test cases improves the accuracy of the solution with each iteration, as the representation range of individual nodal displacements $\hat{u}_i$ contracts. Additionally, the adaptive number representation trials converged on average after 8 iterations, whereas the fixed range trials required an average of 10 iterations to converge. This means that, in this test case, the improved solutions obtained in each iteration using the adaptive number representation led to faster convergence.
In a modified test, the number of qubits per node was reduced to $N_q = 3$. The results are presented in Figure 3.18. For the fixed range trials, the average number of iterations required for convergence was 4. However, it is important to note that the error of the converged solutions exceeded a value of 10^{-1}, indicating that these results were highly inaccurate. In contrast, the adaptive number representation trials required an average of 10 iterations to converge, but the accuracy of the converged solution was significantly improved, with the error reduced to just above 10^{-3}.
This result highlights that the adaptive number representation can be used effectively with far fewer qubits per node, which reduces the size of the problem on the QPU. As shown above, smaller problem sizes are easier for the QPU to process, resulting in fewer chains and better solution quality. Additionally, using a smaller N_q ensures that the h and J step sizes on the QPU do not approach the limits of its resolution, allowing the LEA to be consistently achieved. This will be demonstrated in the next example.

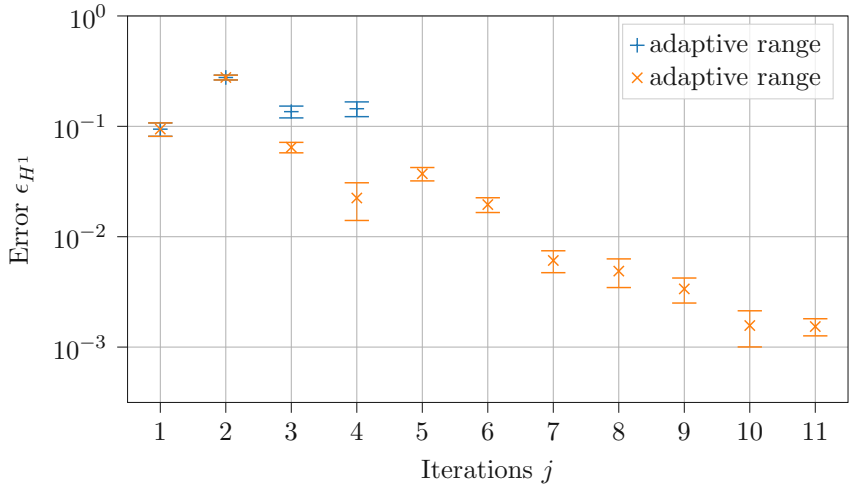


Figure 3.18: Comparing fixed and adaptive range methods with $N_q = 3$.

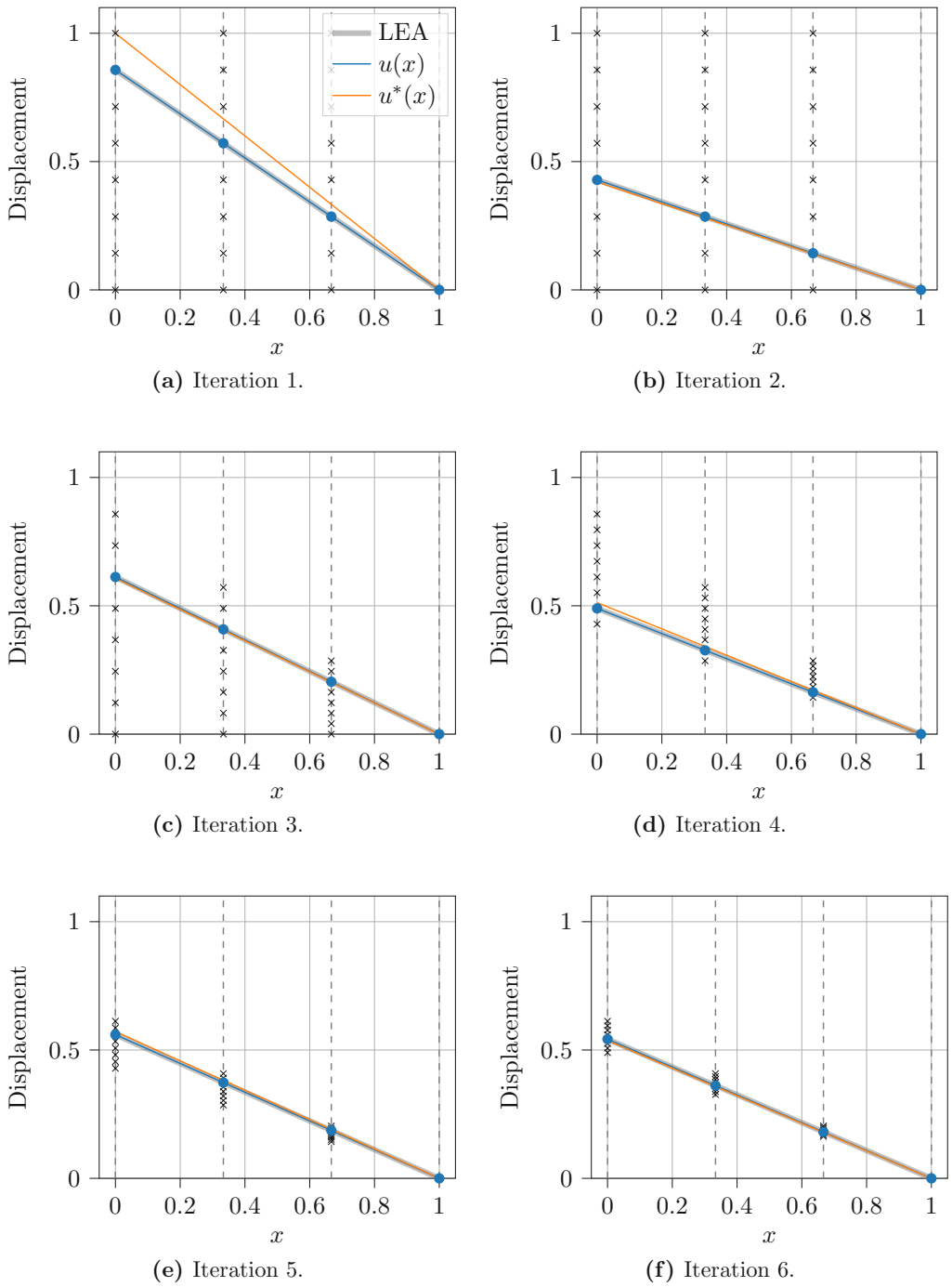
SP5: Convergence Behavior of the Adaptive Number Representation

In this test case, the solution process using the adaptive number representation will be discussed in detail. The test case settings are provided in Table 3.6. The displacement function $u(x)$ will be shown for each iteration to analyze the convergence behavior and other relevant aspects of the adaptive number representation.

Table 3.6: Settings for SP5.

N_e	N_q	Rep. Range	Rep. Method	Trials	p_0
3	3	adaptive	binary	1	0.5

The solution of this test converged after 9 iterations. The displacement over the length of the rod for the first 6 iterations is shown in Figure 3.19. In addition to the computed solution $u(x)$, the analytical solution $u^*(x)$ and the LEA are plotted for comparison. The black crosses indicate the possible values that the solution for each node can take. Because we have chosen $N_q = 3$, there are $2^3 = 8$ possible values for each node (except for the last one where the displacement boundary condition is applied).



(a) Iteration 1. **(b)** Iteration 2.

(c) Iteration 3. **(d)** Iteration 4.

(e) Iteration 5. **(f)** Iteration 6.

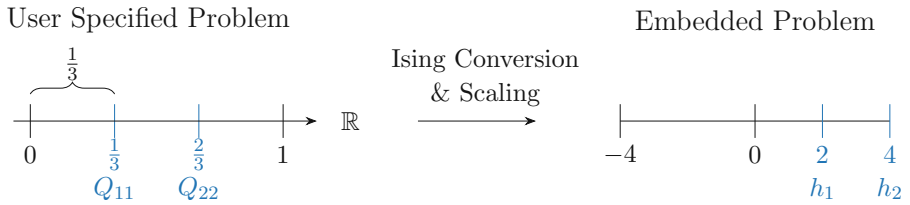
Figure 3.19: Displacement function $u(x)$ over the first six iterations of the adaptive number representation method. The analytical solution u^* and the LEA are shown for comparison.

Focusing on the first iteration (c.f. Figure 3.19a), we observe that the QA solution lies below the analytical solution. In terms of the nodewise distance between the two solutions, there are alternatives closer to the analytical solution. However,

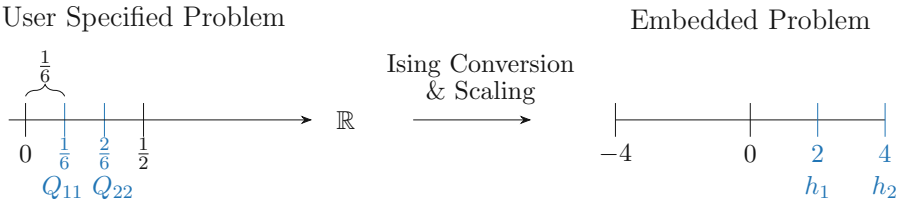
as discussed in Chapter 2, the potential energy (which is our objective function) depends not only on $u(x)$ but also on its derivative $u'(x)$. This means that the best solution is not simply the one closest to the analytical solution, which would minimize the error in the L^2 norm. This is confirmed by the fact that the LEA matches the QA solution.

Next, we will discuss the solution in iteration 3 (c.f. Figure 3.19c). Here, the first contraction of the representation range is performed. As discussed, this contraction is based on the two previous iterations (iteration 1 and 2). Due to the fact that in iteration 2 every nodal solution $\hat{u}_i^{(2)}$ was less than the corresponding solution of iteration 1, the upper limits are set to the solutions $\hat{u}_i^{(1)}$. The lower limits remain unchanged.

In iteration 6 (c.f. Figure 3.19f), we can see that the representation ranges have been contracted tightly around the converging solution. This allows for a high precision in the representation of the real-valued nodal solutions $\hat{u}_i^{(6)}$, resulting in a low error compared to the analytical solution.
It is important to note that, in contrast to the fixed number representation, improving the precision by adjusting the representation range does not lead to smaller h and J coefficients on the problem embedded in the QPU.



(a) Number representation with range $[0, 1]$.



(b) Number representation with range $[0, \frac{1}{2}]$.

Figure 3.20: Comparison of the user-specified problem and the embedded problem for two different representation ranges of a real number.

This can be demonstrated using a simple example where a real number representation is performed with $N_q = 2$ qubits in two different representation ranges. In Figure 3.20a, we choose a representation range of $[0, 1]$. According to Equation (3.12), this results in QUBO coefficients of $Q_{11} = \frac{1}{3}$ and $Q_{22} = \frac{2}{3}$, corresponding to a resolution of $\frac{1}{3}$. The right side of the figure shows the resulting h coefficients of the embedded Ising model after conversion and scaling. As seen, the scaling ensures that the supported range of the QPU is fully utilized, leading to $h_1 = 2$ and $h_2 = 4$.

In Figure 3.20b, we reduce the representation range to $[0, \frac{1}{2}]$, improving the resolution to $\frac{1}{6}$, thus allowing real numbers to be represented with higher precision. However, as shown on the right side of Figure 3.20b, the h coefficients of the embedded problem remain unchanged compared to Figure 3.20a. This demonstrates that improving the precision by reducing the representation range does not affect the coefficients of the embedded Ising problem.
This is a key advantage over the fixed representation range, where precision can only be improved by increasing the number of qubits per node N_q. A higher N_q leads to smaller coefficients in the embedded problem, making it more sensitive to minor deviations, such as DAC quantization errors, as previously discussed.

Returning to the analysis of Figure 3.19, we can see that in all plots, the QA solution matches the LEA, which means that the best possible solution can be found in each iteration. This indicates that, with the adaptive number representation, the problem can be effectively processed by the QPU.

CHAPTER 4

Dynamic Problem

In contrast to the static problems discussed so far, many real-world FSI problems involve dynamic processes. This makes it crucial to develop and analyze methods that incorporate QA for such problems as well. As discussed in Chapter 2, the partitioned approach combined with Hamilton's principle should serve as the basis for addressing these kinds of problems. However, before applying this methodology to FSI problems, it is essential to first evaluate whether Hamilton's principle is fundamentally suitable for formulating the structural subproblem as an optimization problem, which allows the solution via QA. Therefore, this chapter focuses solely on the structural problem. In Section 4.1, a simple example is introduced to illustrate the use of Hamilton's principle as a minimization technique, and the results of this approach are discussed in Section 4.2. The subsequent section addresses the limitations identified and introduces an adapted method with the aim of overcoming these issues.

4.1 Methods for the Dynamic Problem

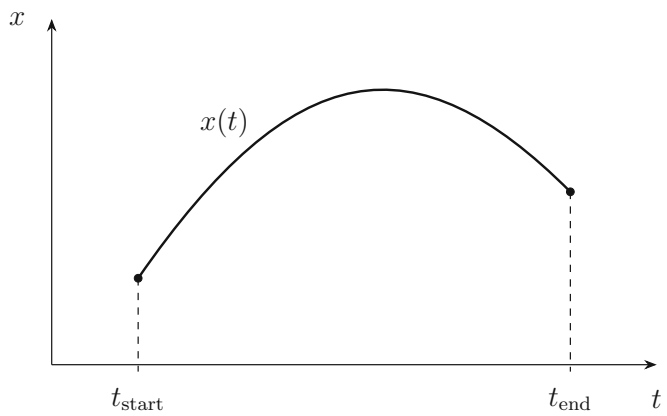


Figure 4.1: Path $x(t)$ of a particle with known initial and final positions.

For addressing the dynamic structural problem, we use a highly simplified example where performing the analysis still provides all the necessary findings to answer the

question from above. One of the main simplifications is the transition from a continuum to a discrete model, reducing the problem to a single-particle system. This means that the unknown is not the time-dependent displacement $u(t)$ as introduced in Chapter 2, but rather the position $x(t)$ of the particle. Specifically, we consider a single particle of mass m moving vertically in the earth's gravitational potential, defining this upward direction as the positive x-direction. As required by Hamilton's principle, we consider that the initial position $x(t_{\text{start}})$ and the final position $x(t_{\text{end}})$ of the particle are known quantities. We can imagine the particle as a ball thrown vertically upward at t_{start}, where x is the height above the ground, before descending back to a specific height at t_{end}. The goal is to determine the path or trajectory $x(t)$ of the ball. An example of such a path is illustrated in Figure 4.1.

To determine the motion of the particle (i.e., the correct path $x(t)$) using Hamilton's principle, we must first specify the kinetic and potential energy of the particle. The potential energy Π is defined as

$$\Pi = mgx(t), \tag{4.1}$$

where g is the gravitational acceleration. The kinetic energy is given by

$$K = \frac{1}{2}mv(t)^2, \tag{4.2}$$

where $v = \frac{dx}{dt}$ is the velocity of the particle. With the Lagrangian being $L = K - \Pi$, the action S is given by

$$S = \int_{t_{\text{start}}}^{t_{\text{end}}} L \, \mathrm{d}t = \int_{t_{\text{start}}}^{t_{\text{end}}} \frac{1}{2}mv(t)^2 - mgx(t) \, \mathrm{d}t. \tag{4.3}$$

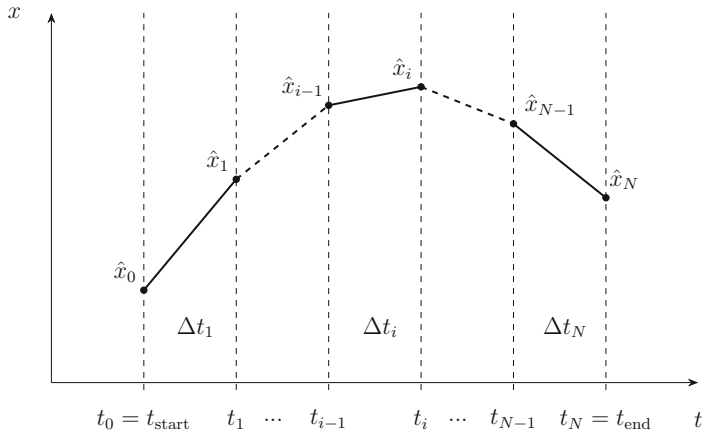


Figure 4.2: The path $x(t)$ composed into N_e time intervals.

To ensure that the method is also applicable to problems with more complex motions involved, we adopt a discretization approach. Unlike the static case, where discretization is performed in space, here we discretize in time. Note that for continuum systems a discretization in both space and time must be employed. We

compose the path $x(t)$ of the particle into N_e time intervals Δt_i, where $i = 1, ..., N_e$, as shown in Figure 4.2. Each time interval Δt_i spans from t_{i-1} to t_i. At the interval boundaries, we consider the path as unknown nodal position $\hat{x}_i$, resulting in $N_e + 1$ nodes. The first and the last nodal position x_0 and x_{N_e} are known quantities. We define the path $x(t)$ as follows:

$$x(t) = \sum_{i=0}^{N_e} \hat{x}_i \Phi_i(t), \tag{4.4}$$

where $\Phi_i(t)$ are nodal interpolation functions. As for the spatial discretization in the static problem formulation, linear interpolation functions are chosen for simplicity:

$$\Phi_i(t) = \begin{cases} 0, & t < t_{i-1}, \\ \frac{t - t_{i-1}}{t_i - t_{i-1}}, & t_{i-1} \leq t < t_i, \\ \frac{t - t_{i+1}}{t_i - t_{i+1}}, & t_i \leq t < t_{i+1}, \\ 0, & t \geq t_{i+1}. \end{cases} \tag{4.5}$$

The velocity of the particle is obtained as:

$$v(t) = \sum_{i=0}^{N_e} \hat{x}_i \dot{\Phi}_i(t), \tag{4.6}$$

where $\dot{\Phi}_i(t) = \frac{d\Phi_i(t)}{dt}$ represents the derivative of the interpolation functions with respect to t. By substituting Equations (4.4) and (4.6) into Equation (4.3), the action in one individual time interval Δt_i is given by:

$$S_i = \int_{t_{i-1}}^{t_i} \frac{1}{2} m (\sum_{i=0}^{N_e} \hat{x}_i \dot{\Phi}_i^2) - mg (\sum_{i=0}^{N_e} \hat{x}_i \Phi \, \mathrm{d}t) \tag{4.7}$$

$$= \int_{t_{i-1}}^{t_i} \frac{1}{2} m (\hat{x}_{i-1} \dot{\Phi}_{i-1} + x_i \dot{\Phi}_i)^2 - mg (\hat{x}_{i-1} \Phi_{i-1} + x_i \Phi_i) \, \mathrm{d}t \tag{4.8}$$

The total action is determined by summing the contributions of all intervals:

$$S = \sum_{i=1}^{N_e} S_i. \tag{4.9}$$

As discussed in Section 3.1, a QUBO problem requires binary variables. Therefore, the real-valued nodal positions $\hat{x}_i$ must be expressed using the number representation previously introduced to convert them into binary variables. However, for the dynamic problem, our focus is to verify whether Hamilton's principle is fundamentally suitable for formulating dynamic problems as optimization problems. To study this, we begin with solving the problem using classical methods that are capable of dealing with real-valued variables. In a subsequent step, the real number representation can then be applied to solve the optimization problem using QA.

As stated in Chapter 2, the action S attains generally a minimum or saddle point for the true path, and only for sufficiently short paths it is always a minimum.

Additionally, for 1D potentials $\Pi(x)$ where the condition $\frac{\partial^2 \Pi}{\partial x^2} \leq 0$ holds, the action S is always minimized for the true path, regardless of the path length [8]. Since this condition also applies to the potential energy in our case (c.f. Equation (4.1)), we can formulate the path problem as a minimization problem:

$$\min_{\hat{x}_i} \{S[\hat{x}_i]\}, \tag{4.10}$$

Therefore, minimizing the action S determines the true nodal positions $\hat{x}_{i,\text{opt}}$.

4.2 Results for the Dynamic Problem

In test case *dynamic problem 1* (DP1), we demonstrate the suitability of the presented formulation for solving the problem introduced in Section 4.1 in a single minimization step. The initial position $x(t_{\text{start}}) = x_{\text{start}}$ and the final position $x(t_{\text{end}}) = x_{\text{end}}$ are specified as boundary conditions.
To validate the presented minimization method, the following equations are used to compute the analytical reference solution. The path of the particle can be computed as

$$x^*(t) = x_{\text{start}} + v_{\text{start}}(t - t_{\text{start}}) - \frac{1}{2} g (t - t_{\text{start}})^2, \tag{4.11}$$

where v_{start} is the initial velocity, which can be computed using the boundary condition $x^*(t_{\text{end}}) = x_{\text{end}}$:

$$v_{\text{start}} = \frac{x_{\text{end}} - x_{\text{start}} + \frac{1}{2} g (t_{\text{end}} - t_{\text{start}})^2}{t_{\text{end}} - t_{\text{start}}}. \tag{4.12}$$

For the numerical result, the time span is discretized into $N_e = 8$ time intervals, and the problem is solved using the proposed minimization method. The settings for this test case are summarized in Table 4.1.

Table 4.1: Settings for DP1.

m	g	t_{start}	t_{end}	x_{start}	x_{end}	N_e
1	9.81	0	3	0	5	8

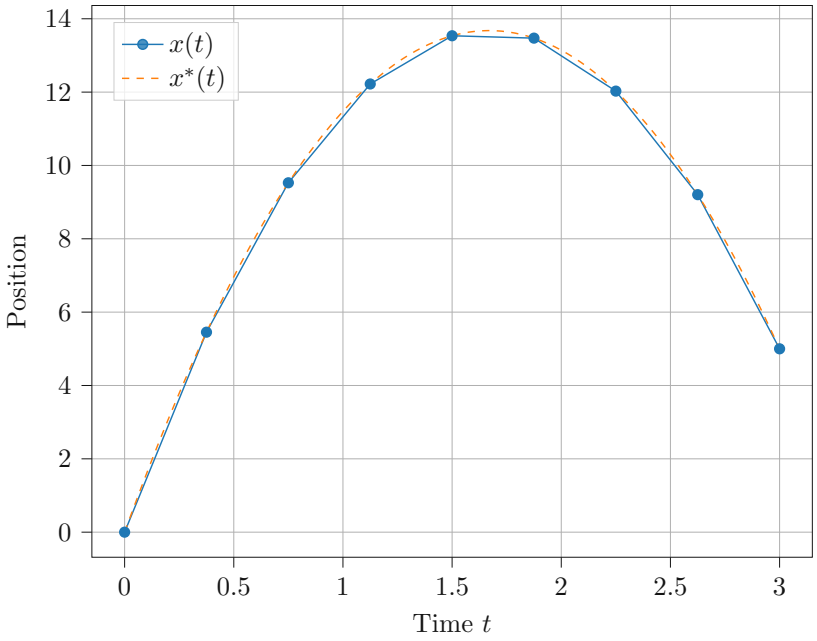


Figure 4.3: Comparison of the numerical solution $x(t)$ and the analytical solution $x^*(t)$ for the boundary value problem.

The resulting numerical solution $x(t)$ of the particle for the presented test case is shown in Figure 4.3 as the blue solid line. For comparison, the analytical solution $x^*(t)$ is displayed as the orange dashed line. It can be observed that the numerical solution $x(t)$ matches the analytical solution exactly at the nodal points (blue dots). In addition, linear interpolation between the nodes can be seen.

This test case demonstrates that the proposed method is capable of solving a dynamic problem formulated as a boundary value problem. However, there is a significant limitation to this approach. In typical dynamic problems, the position (or deformation, in the case of a continuous problem) at the end of a specific time-span is unknown. Instead, such problems are formulated as initial value problems, where the initial position and velocity are given, and the motion is computed based on these initial conditions.

To show that the current method is not capable of solving initial value problems, the previous test case is modified to prescribe initial values instead of boundary values. Specifically, instead of specifying the final position x_{end}, the initial velocity $v_{\text{start}} = \frac{\mathrm{d}x}{\mathrm{d}t}\Big|_{t=t_{\text{start}}}$ was additionally provided. The initial velocity was set to $v_{\text{start}} = 16.382$ to achieve the same analytical path as in the previous example.

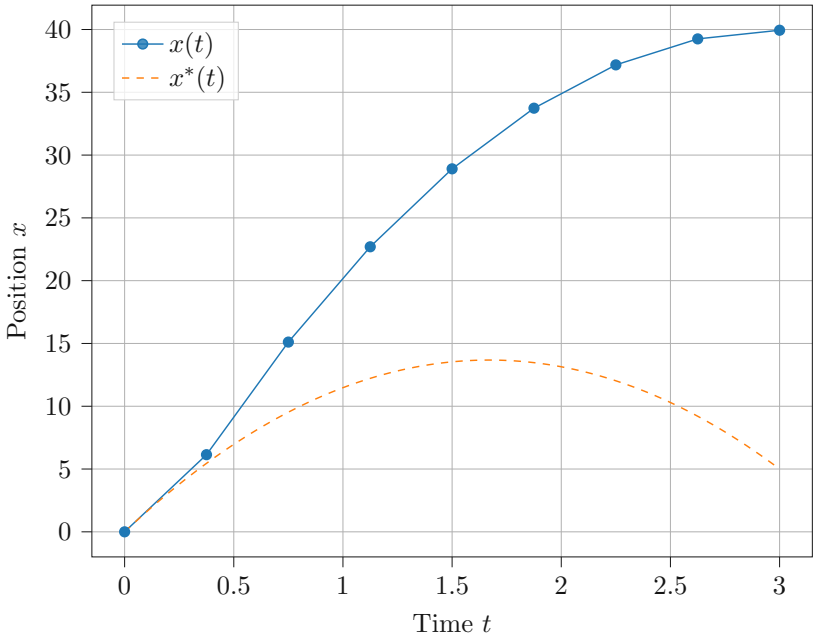


Figure 4.4: Comparison of the numerical solution $x(t)$ and the analytical solution $x^*(t)$ for the initial value problem.

Figure 4.4 visualizes the result of this test. It can be seen that the numerical solution $x(t)$ does not match the correct analytical solution.

To address this limitation and adapt Hamilton's principle as a minimization principle for solving initial value problems, the next chapter introduces a modification to the current method.

4.3 Modified Method for the Dynamic Problem

The modified method is based on the idea of transforming the initial value problem into a boundary value problem. Therefore, we first approximate the trajectory $x(t)$ in each time interval using a linear approximation. In the second step, this linear approximation is corrected by quadratic terms. The coefficients of the quadratic terms are determined using Hamilton's principle.

We introduce the method by considering one time interval Δt_i of a discretized path $x(t)$ (c.f. Figure 4.5). As can be seen in the sketch, the interval spans from t_{i-1} to t_i, where $\hat{x}_{i-1}$ and $\hat{x}_i$ are the nodal positions of the unknown path $x(t)$. We assume that $\hat{x}_{i-1}$ is known and $\hat{x}_i$ is unknown. The goal is to calculate the unknown path $x(t)$ in this time interval using Hamilton's principle as a minimization principle.

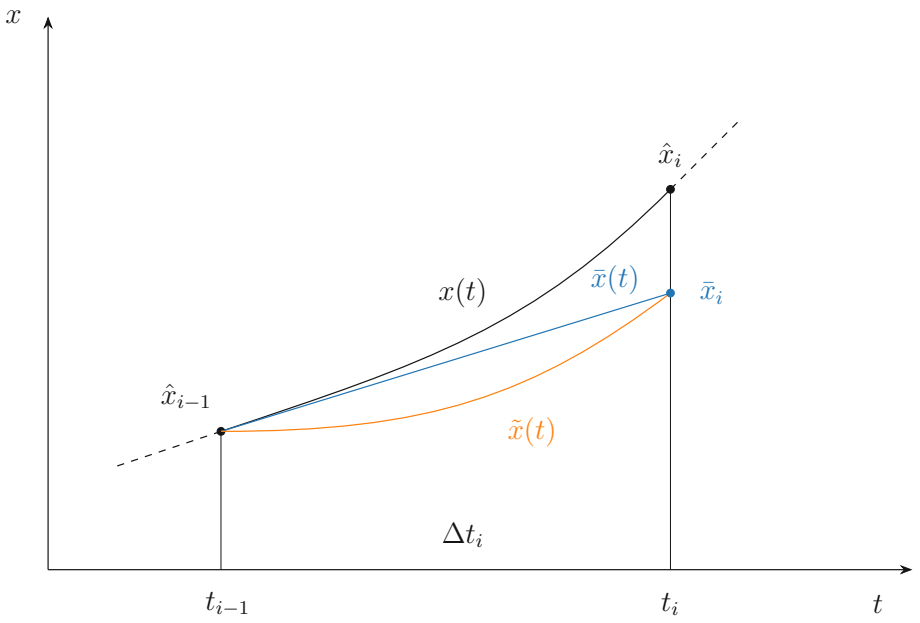


Figure 4.5: Time interval Δt_i with the nodal position values $\hat{x}_{i-1}$ and $\hat{x}_i$ and the path $x(t)$ in between. The linear approximation function $\bar{x}(t)$ and the unknown quadratic function $\tilde{x}(t)$ transform the initial value problem into a boundary value problem.

To accomplish this, we begin by computing a linear approximation $\bar{x}(t)$ of the unknown path in the interval Δt_i. This approximation can be written as

$$\bar{x}(t) = \hat{x}_{i-1} + \dot{x}(t_{i-1})(t - t_{i-1}), \tag{4.13}$$

where $\dot{x}(t_{i-1}) = \frac{\mathrm{d}x}{\mathrm{d}t}\big|_{t=t_{i-1}}$ can be computed from $x(t)$ of previous time interval Δt_{i-1}. We define $\bar{x}(t_i)$ as a new position $\bar{x}_i = \bar{x}(t_i)$. Next, we introduce a quadratic function $\tilde{x}(t)$, which also spans from t_{i-1} to t_i:

$$\tilde{x}(t) = \alpha_i + \beta_i(t - t_{i-1}) + \gamma_i(t - t_{i-1})^2. \tag{4.14}$$

Here, α_i, β_i and γ_i are unknown coefficients for the constant, linear and quadratic term of the function. As can be seen in the sketch Figure 4.5, this quadratic function has predefined endpoints that match the endpoints of the linear approximation function $\bar{x}(t)$. Thus, we can formulate two boundary conditions from which two of the three unknown coefficients of $\tilde{x}(t)$ can be calculated. Substituting $t = t_{i-1}$ into $\tilde{x}(t)$ and requiring it to be equal the position $\hat{x}_{i-1}$, we can directly determine the coefficient α_i:

$$\text{I)} \quad \tilde{x}(t_{i-1}) = \hat{x}_{i-1} \quad \Rightarrow \quad \alpha_i = \hat{x}_{i-1}. \tag{4.15}$$

The second boundary condition ensures that the quadratic function $\tilde{x}(t)$ matches the linear approximation $\bar{x}(t)$ at the endpoint $\bar{x}_i$. This leads to an expression for β_i depending on the coefficient γ_i:

$$\text{II)} \quad \tilde{x}(t_i) = \bar{x}_i \quad \Rightarrow \quad \beta_i = \dot{x}(t_{i-1}) - \gamma_i(t_i - t_{i-1}). \tag{4.16}$$

Now we can substitute the expressions for α_i and β_i into Equation (4.14), which leads to

$$\tilde{x}(t) = \hat{x}_{i-1} + \dot{x}(t_{i-1})(t - t_{i-1}) + \gamma_i(t - t_{i-1})(t - t_i). \tag{4.17}$$

The only remaining unknown coefficient in this quadratic function is γ_i. With Equation (4.14) to Equation (4.17), we have formulated a boundary value problem, where the correct value for γ_i must be found to determine the unknown function $\tilde{x}(t)$. To determine γ_i, we can use Hamilton's principle by defining the action dependent on $\tilde{x}(t)$:

$$S_i[\tilde{x}(t)] = \int_{t_{i-1}}^{t_i} K_i[\tilde{x}(t)] - \Pi_i[\tilde{x}(t)]\,\mathrm{d}t. \tag{4.18}$$

There are two possible ways to determine the unknown path $x(t)$ using the presented method.

The first approach involves minimizing the action for each time interval in an iterative manner. Specifically, for interval Δt_i, the action, as defined in Equation (4.18), is minimized with respect to the unknown coefficient γ_i:

$$\min_{\gamma_i} \{S_i[\tilde{x}(t)]\} \tag{4.19}$$

Once the optimal solution $\gamma_{i,\mathrm{opt}}$ is determined, the quadratic function $\tilde{x}_i(t)$ is fully defined. The path $x(t)$ in the interval Δt_i can then be obtained by correcting the linear approximation $\bar{x}(t)$ with the quadratic term:

$$x(t) = \bar{x}(t) + \gamma_{i,\mathrm{opt}}(t - t_{i-1})^2 \tag{4.20}$$

$$= \hat{x}_{i-1} + \dot{x}(t_{i-1})(t - t_{i-1}) + \gamma_{i,\mathrm{opt}}(t - t_{i-1})^2. \tag{4.21}$$

The solution for $x(t)$ also determines the values $\hat{x}_i$ and $\dot{x}(t_i)$, which are then used as inputs for the linear approximation $\bar{x}(t)$ of the next time interval Δt_{i+1}. Repeating this process for all intervals results in the complete path $x(t)$.

DP2: Results of the iterative Approach

Test case DP2 demonstrates the iterative results. The settings for this test case are provided in Table 4.2.

Table 4.2: Settings for DP2.

m	g	t_{start}	t_{end}	x_{start}	v_{start}	N_e
1	9.81	0	3	0	16	6

Figure 4.6 shows the computed path $x(t)$ obtained by applying the presented method iteratively for each time interval. The dashed green and red lines represent the functions $\bar{x}(t)$ and $\tilde{x}(t)$, respectively, for each interval. The analytical solution $x^*(t)$ is included for comparison. As seen, the two curves are identical. The perfect representation of $x^*(t)$ can be explained by the fact that $x^*(t)$ is a quadratic function, just like the quadratic approximation $x(t)$.

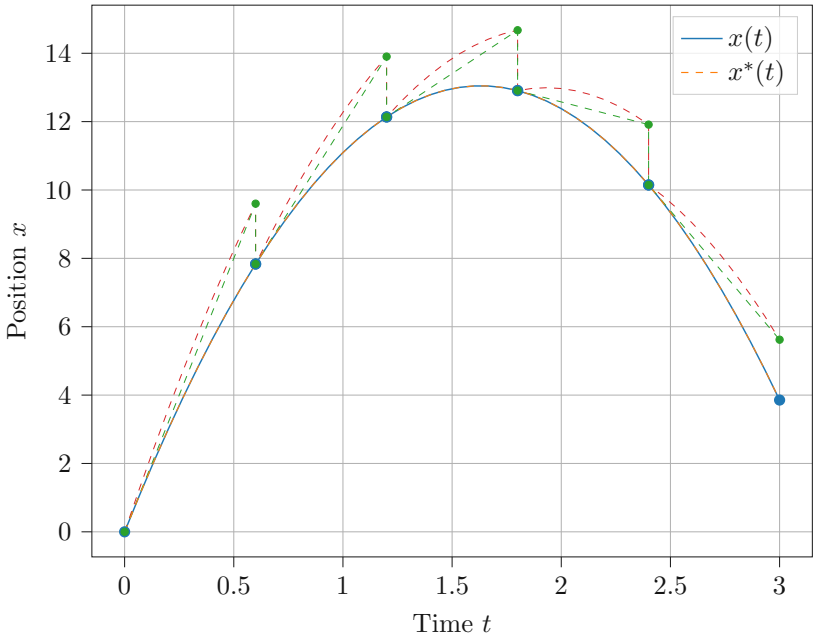


Figure 4.6: Path $x(t)$ computed iteratively compared with the analytical solution $x^*(t)$.

However, while the iterative method demonstrates correctness, it has limited practical significance for problems we consider. This is because the minimization in each time interval is not computationally demanding, and thus, the advantages of QA cannot be effectively utilized.

Thus, in a second approach, we aim to address all time intervals simultaneously, avoiding any iteration. Therefore, the action S_i of each interval is formulated and added to the total action S, with all coefficients γ_i treated as unknowns:

$$S = \sum_{i=1}^{N_e} S_i[\tilde{x}(t)]. \tag{4.22}$$

The complete action S is minimized in a single step to determine all $\gamma_{i,\text{opt}}$ at once.

DP3: Results of the Direct Approach

To test this direct approach, the test case settings of DP2 are used. The results are shown in Figure 4.7.

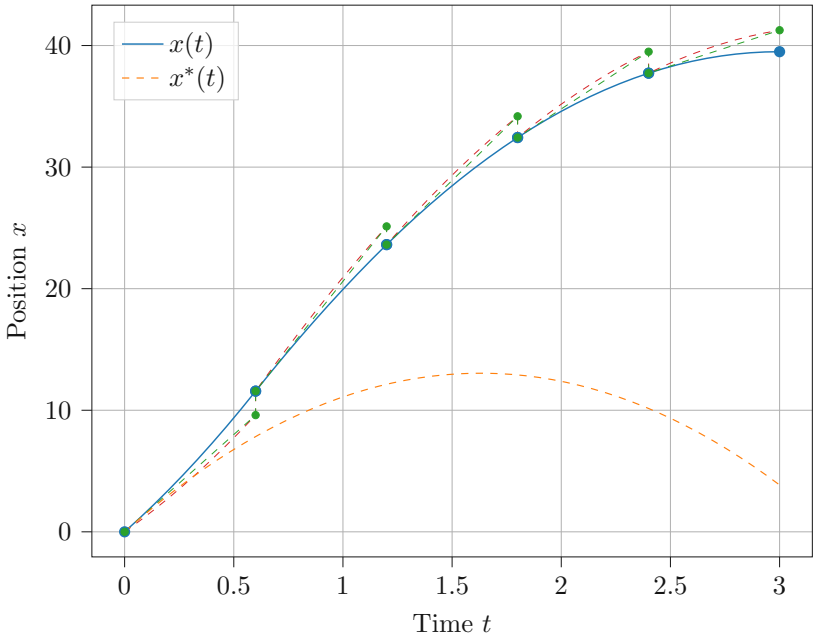
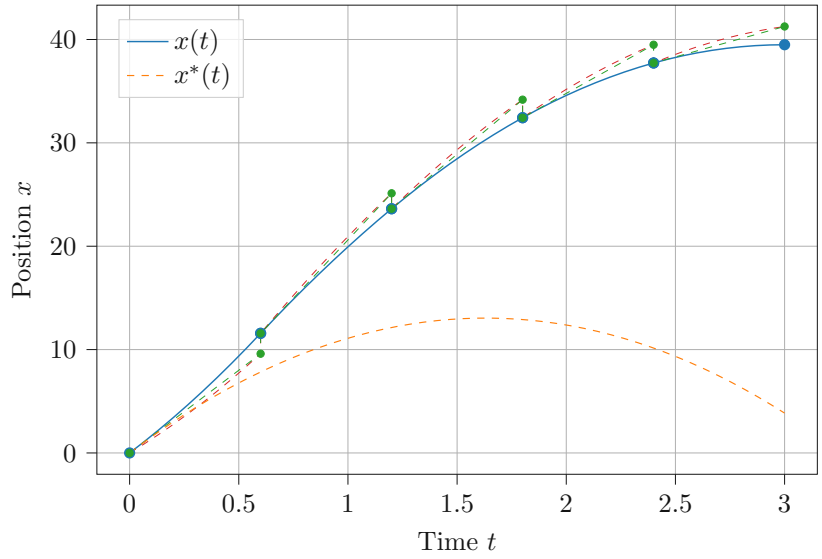
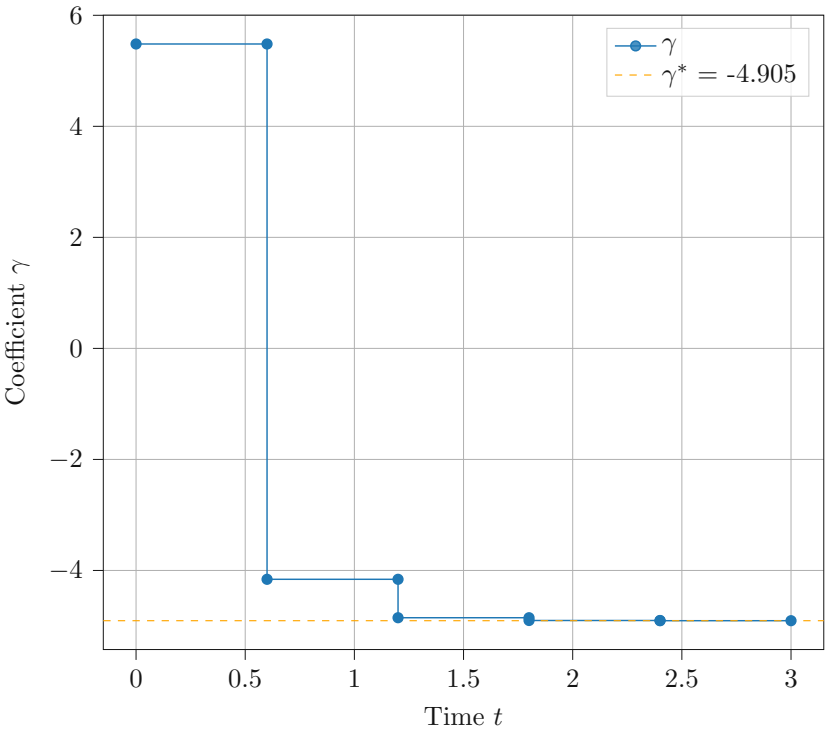


Figure 4.7: Path $x(t)$ computed using the direct approach compared with the analytical solution $x^*(t)$.

As seen in Figure 4.7, the direct method produces incorrect results. The failure of the direct method lies in the coupling of the coefficients γ_i across time intervals, which results from the boundary conditions Equations (4.15) and (4.16). Specifically, the path $x(t)$ in each interval Δt_i is influenced not only by its own coefficient γ_i, but also by the coefficients of all previous intervals. For example, the coefficient γ_1 of the first interval affects not only $x(t)$ in Δt_1, but also influences the paths of all subsequent intervals $\Delta t_2, \Delta t_3, ..., \Delta t_{N_e}$. This effect arises because the start point $\hat{x}_{i-1}$ and the slope $\dot{x}(t_{i-1})$ at the beginning of each interval are determined by the boundary conditions, which are influenced by earlier intervals.
As a result, when minimizing the total action S, the direct approach allows earlier coefficients (e.g., γ_1) to adjust the paths in all subsequent intervals in order to find the minimum value of the action S. This leads to a smaller action that is not physically correct.



(a) Path $x(t)$ computed using the direct approach.



(b) Coefficients γ_i in each time interval.

Figure 4.8: Path $x(t)$ and the coefficients γ_i in each time interval.

This issue is illustrated in (Figure 4.8), where the coefficients γ_i in the time intervals are displayed in addition to the result of the test case of the previous test. The coefficient γ_{N_e} of the last interval, which only influences its own path, matches

the analytical solution $\gamma^* = -\frac{g}{2} = -4.905$. In contrast, the coefficients in earlier intervals increasingly deviate from γ^*. These deviations occur because the earlier coefficients influence not only their assigned paths but also all subsequent ones. Consequently, the calculated path $x(t)$ has a total action of $S_{\text{opt}} = -413.5$, which is a significantly lower value than the physically correct value of $S^*_{\text{opt}} = -162.5$. This confirms that the direct method fails to find the physically correct solution.

In summary, the modified method so far is unsuitable for solving dynamic initial value problems as a one-step optimization problem utilizing Hamilton's principle. The only way to obtain a physically correct solution using this method is by solving the problem iteratively. This approach ensures that the boundary conditions in each interval are satisfied independently before moving on to the next, allowing each coefficient γ_i to be calculated based solely on its own path. However, the iterative process has limited practical significance for this thesis, as the individual minimizations in each interval are not computationally expensive and, therefore, the advantages of QA cannot be effectively utilized.
Consequently, this chapter emphasizes the need to develop alternative approaches to effectively handle dynamic problems using QA.

CHAPTER 5

Conclusion

In this study, we explored QA for coupled structural analysis problems. Specifically, we focused on developing methods that allow using QA to solve FSI problems. This should serve as a basis for novel optimization strategies for FSI problems through the efficient use of QA. We based our methods on the partitioned approach, where fluid and structural problems are treated separately. In this context, we investigated static and dynamic problems.

For the static problem, we reformulated the structural subproblem by applying the principle of minimum potential energy and expressing the system's energy in QUBO form. This method was validated on a state-of-the-art D-Wave quantum annealer, with the results demonstrating the feasibility of QA for solving simplified problems, such as the static piston problem.
The findings highlighted several key aspects regarding the practical application of QA for such problems. First, the real number representation method played a critical role, with the binary representation generally outperforming the random representation in terms of accuracy. However, increasing the precision of the number representations beyond a certain threshold resulted in a decrease in solution quality. This issue became more pronounced with increasing problem size, characterized by the number of elements. The decreasing solution quality is due to hardware-induced inaccuracies, such as DAC quantization errors.
To address these limitations, we introduced a method with an adaptive number representation. This method makes use of the iterative solution process of the partitioned approach by adjusting the range of representable real numbers in each step, resulting in significantly improved solution quality. Additionally, fewer qubits per node were required, which is particularly advantageous given the current hardware limitations of QA.

In the second part of the thesis, we addressed a dynamic structural problem as a basis for solving dynamic FSI problems. We proposed a method based on Hamilton's principle, formulating the problem as an optimization task suitable for QA. For boundary value problems, this approach produced correct results. However, real-world dynamic problems are typically initial value problems. To handle initial value problems, we developed a method that reformulates an initial value problem into element-wise boundary value problems through linear approximation. Iteratively applying this method to every element yielded correct results. However, the iterative

application is of limited practical significance to optimization problems we aim to solve efficiently using QA in one go. When using the method in a single step, coupling between the unknown coefficients led to incorrect results.
This circumstance creates a need for further research to adapt the method or develop new approaches to formulating dynamic problems suitable for QA. This could involve refining the approach based on Hamilton's principle or exploring entirely different principles.

Additionally, it is worth investigating to what extent a monolithic approach to solving FSI problems is suitable for QA. Although less modular than the partitioned approach, a monolithic approach could allow QA to solve the problem in a single step, potentially offering certain advantages. However, the monolithic approach comes with considerable challenges. Combining the structural and fluid subproblems into a single system increases problem complexity, requiring a larger number of qubits. Given the current hardware limitations of QA, this could impose significant constraints or even make practical implementation impossible. Therefore, it is important to evaluate the limitations against the potential benefits to determine whether a monolithic approach is an appropriate alternative.
The next step is to build on these methods and findings to develop approaches that use QA to solve FSI optimization problems. This would allow QA to fully exploit its strength in finding optimal solutions in complex solution spaces, fast and effectively.

Although current QA hardware is limited by the number of qubits and hardware-induced inaccuracies, it is essential to explore innovative strategies that enable its effective use for engineering problems such as FSI. This study serves as a foundation for developing methods, identifying limitations, and finding strategies to overcome these limitations, laying the groundwork for future advancements in using QA for this engineering field.

List of Figures

List of Tables

Bibliography

1. Grover LK. A fast quantum mechanical algorithm for database search. en. *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing - STOC '96*. Philadelphia, Pennsylvania, United States: ACM Press, 1996 :212–9. DOI: 10.1145/237814.237866. Available from: http://portal.acm.org/citation.cfm?doid=237814.237866

2. Shor PW. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. SIAM Journal on Computing 1997 Oct; 26. Publisher: Society for Industrial and Applied Mathematics:1484–509. DOI: 10.1137/S0097539795293172. Available from: https://epubs.siam.org/doi/10.1137/S0097539795293172

3. Bathe M and Kamm RD. A Fluid-Structure Interaction Finite Element Analysis of Pulsatile Blood Flow Through a Compliant Stenotic Artery. Journal of Biomechanical Engineering 1999 Aug; 121:361–9. DOI: 10.1115/1.2798332. Available from: https://doi.org/10.1115/1.2798332

4. Rifai SM, Johan Z, Wang WP, Grisval JP, Hughes TJR, and Ferencz RM. Multiphysics simulation of flow-induced vibrations and aeroelasticity on parallel computing platforms. Computer Methods in Applied Mechanics and Engineering 1999 May; 174:393–417. DOI: 10.1016/S0045-7825(98)00306-5. Available from: https://www.sciencedirect.com/science/article/pii/S0045782598003065

5. Frandsen JB. Numerical bridge deck studies using finite elements. Part I: flutter. Journal of Fluids and Structures 2004 Feb; 19:171–91. DOI: 10.1016/j.jfluidstructs.2003.12.005. Available from: https://www.sciencedirect.com/science/article/pii/S0889974603001841

6. Bessert N and Frederich O. Nonlinear airship aeroelasticity. Journal of Fluids and Structures 2005 Dec; 21:731–42. DOI: 10.1016/j.jfluidstructs.2005.09.005. Available from: https://www.sciencedirect.com/science/article/pii/S0889974605001738

7. Cheong CY, Chiam SC, and Goh CK. Eliminating Positional Dependency in Binary Representation via Redundancy. en. *2007 IEEE Symposium on Foundations of Computational Intelligence*. Honolulu, HI, USA: IEEE, 2007 Apr :251–8. DOI: 10.1109/FOCI.2007.372177. Available from: http://ieeexplore.ieee.org/document/4233915/

8. Gray CG and Taylor EF. When action is not least. American Journal of Physics 2007 May; 75:434–58. DOI: 10.1119/1.2710480

9. Sadd MH. Elasticity: Theory, Applications, and Numerics. en. Academic Press, 2009 Feb

10. Picelli R, Vicente W, and Pavanello R. Bi-directional evolutionary structural optimization for design-dependent fluid pressure loading problems. Engineering Optimization 2015 Oct; 47. Publisher: Taylor & Francis:1324–42. DOI: 10.1080/0305215X.2014.963069. Available from: https://doi.org/10.1080/0305215X.2014.963069

11. Vicente WM, Picelli R, Pavanello R, and Xie YM. Topology optimization of frequency responses of fluid–structure interaction systems. Finite Elements in Analysis and Design 2015 Jun; 98:1–13. DOI: 10.1016/j.finel.2015.01.009. Available from: https://www.sciencedirect.com/science/article/pii/S0168874X15000104

12. Ha ST, Ngo LC, Saeed M, Jeon BJ, and Choi H. A comparative study between partitioned and monolithic methods for the problems with 3D fluid-structure interaction of blood vessels. en. Journal of Mechanical Science and Technology 2017 Jan; 31:281–7. DOI: 10.1007/s12206-016-1230-2. Available from: https://doi.org/10.1007/s12206-016-1230-2

13. Picelli R, Vicente WM, and Pavanello R. Evolutionary topology optimization for structural compliance minimization considering design-dependent FSI loads. Finite Elements in Analysis and Design 2017 Nov; 135:44–55. DOI: 10.1016/j.finel.2017.07.005. Available from: https://www.sciencedirect.com/science/article/pii/S0168874X1730015X

14. REDDY JN. Energy Principles and Variational Methods in Applied Mechanics, 3rd Edition — Wiley. en. 2017. Available from: https://www.wiley.com/en-hk/Energy+Principles+and+Variational+Methods+in+Applied+Mechanics%2C+3rd+Edition-p-9781119087373

15. Mang HA and Hofstetter G. Festigkeitslehre. de. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018. DOI: 10.1007/978-3-662-57564-2. Available from: http://link.springer.com/10.1007/978-3-662-57564-2

16. Gibert M, Bel-Brunon A, Brun M, A. Gravouil, and Rochette M. A monolithic formulation of fluid-structure interaction for the co-simulation : application to the piston problem. eng. Accepted: 2020-06-04T15:24:21Z. CIMNE, 2019 :308–19. Available from: https://upcommons.upc.edu/handle/2117/190051

17. Srivastava S and Sundararaghavan V. Box algorithm for the solution of differential equations on a quantum annealer. Physical Review A 2019 May; 99. Publisher: American Physical Society:052355. DOI: 10.1103/PhysRevA.99.052355. Available from: https://link.aps.org/doi/10.1103/PhysRevA.99.052355

18. Potter MC and Somerton CW. Thermodynamics for engineers. en. Fourth edition. New York: McGraw-Hill Education, 2020

19. Spurk JH and Aksel N. Fluid Mechanics. en. Cham: Springer International Publishing, 2020. DOI: 10.1007/978-3-030-30259-7. Available from: http://link.springer.com/10.1007/978-3-030-30259-7

20. Santra R. Hamilton'sches Prinzip. de. *Einführung in den Lagrange- und Hamilton-Formalismus: Kanonische Theorie klassischer Teilchen und Felder*. Ed. by Santra R. Berlin, Heidelberg: Springer, 2022 :29–45. DOI: 10.1007/978-3-662-65359-3_2. Available from: https://doi.org/10.1007/978-3-662-65359-3_2

21. Yarkoni S, Raponi E, Bäck T, and Schmitt S. Quantum annealing for industry applications: introduction and review. en. Reports on Progress in Physics 2022 Sep; 85. Publisher: IOP Publishing:104001. DOI: 10.1088/1361-6633/ac8c54. Available from: https://dx.doi.org/10.1088/1361-6633/ac8c54

22. Meglio AD. A Practical Guide to Quantum Machine Learning and Quantum Optimization: Hands-on Approach to Modern Quantum Algorithms. en. 1st ed. Birmingham: Packt Publishing Limited, 2023

23. Spenke T, Make M, and Hosters N. A Robin-Neumann scheme with quasi-Newton acceleration for partitioned fluid-structure interaction. en. International Journal for Numerical Methods in Engineering 2023; 124:979–97. DOI: 10.1002/nme.7151. Available from: https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.7151

24. Wils K and Chen B. A Symbolic Approach to Discrete Structural Optimization Using Quantum Annealing. en. Mathematics 2023 Jan; 11. Number: 16 Publisher: Multidisciplinary Digital Publishing Institute:3451. DOI: 10.3390/math11163451. Available from: https://www.mdpi.com/2227-7390/11/16/3451

25. Ye Z, Qian X, and Pan W. Quantum Topology Optimization via Quantum Annealing. en. IEEE Transactions on Quantum Engineering 2023; 4:1–15. DOI: 10.1109/TQE.2023.3266410. Available from: https://ieeexplore.ieee.org/document/10099425/

26. Endo K, Matsuda Y, Tanaka S, and Muramatsu M. Novel real number representations in Ising machines and performance evaluation: Combinatorial random number sum and constant division. en. PLOS ONE 2024 Jun; 19. Publisher: Public Library of Science:e0304594. DOI: 10.1371/journal.pone.0304594. Available from: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0304594

27. Honda R, Endo K, Kaji T, Suzuki Y, Matsuda Y, Tanaka S, and Muramatsu M. Development of optimization method for truss structure by quantum annealing. en. Scientific Reports 2024 Jun; 14. Publisher: Nature Publishing Group:13872. DOI: 10.1038/s41598-024-64588-2. Available from: https://www.nature.com/articles/s41598-024-64588-2

28. Key F and Freinberger L. A Formulation of Structural Design Optimization Problems for Quantum Annealing. en. Mathematics 2024 Jan; 12. Number: 3 Publisher: Multidisciplinary Digital Publishing Institute:482. DOI: 10.3390/math12030482. Available from: https://www.mdpi.com/2227-7390/12/3/482

29. D-Wave Ocean Software Documentation. Available from: https://docs.ocean.dwavesys.com/en/stable/ [Accessed on: 2025 Jan 5]

30. D-Wave QPU-Specific Characteristics. Available from: https://docs.dwavesys.com/docs/latest/doc_physical_properties.html [Accessed on: 2025 Feb 11]

31. D-Wave System Documentation. Available from: https://docs.dwavesys.com/docs/latest/c_gs_2.html [Accessed on: 2025 Jan 8]

32. D-Wave System Documentation - Error Sources. Available from: https://docs.dwavesys.com/docs/latest/c_qpu_ice.html [Accessed on: 2025 Feb 11]

33. D-Wave QPU Architecture - Topologies. Available from: https://docs.dwavesys.com/docs/latest/c_gs_4.html [Accessed on: 2025 Jan 9]

34. Fixstars Amplify SDK. en. Available from: https://amplify.fixstars.com/en/sdk [Accessed on: 2025 Jan 5]

35. Key F. EngiOptiQA (v0.1.0). DOI: 10.5281/zenodo.10222618. Available from: https://zenodo.org/records/10222618 [Accessed on: 2025 Jan 5]