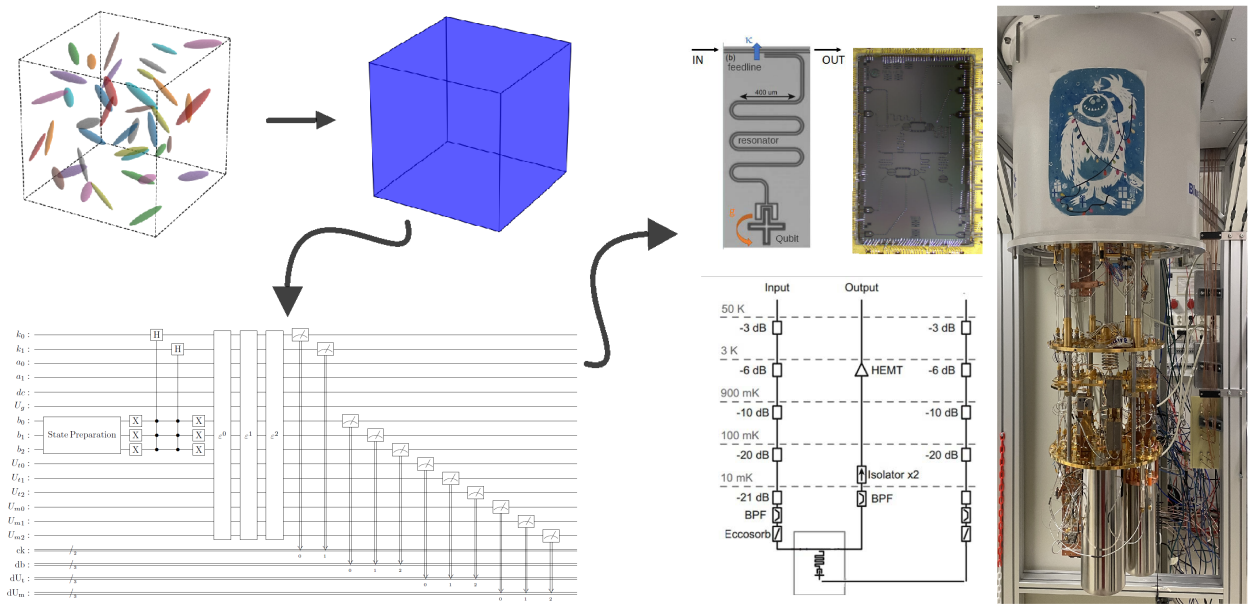
CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG



Quantum RVE solver in the NISQ regime

Implementation of a Quantum Fourier Transform based solution of Representative Volume Element problems on a NISQ device

Master's thesis in Physics

GIRONDEL BIMANA

DEPARTMENT OF PHYSICS

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026
www.chalmers.se

Master's thesis 2026

Quantum RVE solver in the NISQ regime

Implementation of a Quantum Fourier Transform based solution of Representative Volume Element problems on a NISQ device

GIRONDEL BIMANA

UNIVERSITY OF GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Physics
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2026

Quantum RVE solver in the NISQ regime
Implementation of a Quantum Fourier Transform based solution of Representative Volume Element problems on a NISQ device
GIRONDEL BIMANA

Supervisor: Mats Granath, Department of Physics, Gothenburg University
Examiner: Mohsen Mirkhalaf, Department of Physics, Gothenburg University

Master's thesis 2026
Department of Physics
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone +46 31 772 1000

Cover: Workflow of the proposed quantum computational homogenization with the generation of an RVE, encoding a quantum algorithm to solve it and a subsequent run on a superconducting QPU (*Pingu* at Chalmers MC2[1]).

Typeset in LaTeX
Gothenburg, Sweden 2026

[1]G. Tancredi, *Controlling a quantum processor*, Chalmers University of Technology, Gothenburg, Sweden, 2024, lecturer: Giovanna Tancredi; Supervisor: Anuj Aggarwal; Lab Location: F4205 - 4th floor, Fysik forskarhus.

Quantum RVE solver in the NISQ regime
Implementation of a Quantum Fourier Transform based solution of Representative Volume Element problems on a NISQ device

GIRONDEL BIMANA
Department of Physics
Chalmers University of Technology
University of Gothenburg

Abstract

Concurrent multiscale simulations in solid mechanics require repeated solutions of microscopic boundary-value problems. Quantum algorithms based on the quantum Fourier transform (QFT) offer a potential way to reduce this scaling exponentially. Previous work formulated a quantum homogenization algorithm for a one-dimensional Representative Volume Element (RVE) by adapting the Fast Fourier transform (FFT) homogenization approach to a quantum computing setting.
This work extends that formulation by investigating its feasibility on current quantum hardware. Three circuit models were implemented and compared: a full circuit following the original construction, a simplified circuit adapted to the geometry of the considered RVE, and a sequential circuit where the result of one quantum run is recovered and used as input for the next iteration. The circuits were first validated on a noiseless quantum circuit simulator and later executed on IBM quantum hardware using different error-suppression configurations.
The hardware results showed that the fully quantum implementations are strongly limited by noise accumulation. The full circuits produced mostly noisy outputs, while the simplified circuit showed significant improvement, with average errors in the interval $23.91 - 36.19\%$. The best hardware results were obtained with the sequential circuit, which reduced the circuit depth and gate count and reached an average relative error of 25.00% (11% when limited to two iterations).
The fully quantum models retain the expected $\mathcal{O}(\text{poly}(\log N))$ scaling. The sequential model, however, requires intermediate field recovery between QPU runs, making its complexity comparable to the classical FFT-based approach.

Keywords: quantum computing, composite materials, FFT, QFT, representative volume element, computational homogenization, NISQ, twirling, dynamical decoupling.

Acknowledgements

First and foremost, I want to express my sincere gratitude to my examiner Professor Mohsen Mirkhalaf, for his support, his advice and his feedback that helped me to improve my work.

I also want to thank my supervisor Professor Mats Granath, for providing suggestions and the necessary computational resources that were needed to complete the project.

This work acknowledges support from the WACQT Quantum Technology Testbed operated by Chalmers Next Labs, which is funded by the Knut and Alice Wallenberg Foundation.

I enjoyed my time at the Physics' department of the University of Gothenburg. I would like to thank the colleagues at the department and my fellow thesis writer for warmly welcoming me into their community.

Finally, I want to thank my parents and my sisters for their everlasting love and support throughout this process and beyond.

Girondel Bimana, Gothenburg, May 2026

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

DD	Dynamical Decoupling
DFT	Discrete Fourier Transform
JJ	Josephson Junction
FFT	Fast Fourier Transform
NISQ	Noise Intermediate-scale Quantum
QFT	Quantum Fourier Transform
QPU	Quantum Processing Unit
RVE	Representative Volume Element
SFRC	Short Fiber Reinforced Composite

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this thesis.

Indices

i	Index for iterations
j	Index for bit string
v	Index for discretized field points expressed as state probability amplitudes
w	Index for domain subintervals

Sets

$\mathcal{P}$	Pauli set for randomization
Ω	Set of points occupied by the RVE

Parameters

θ, φ	Polar - and azimuthal rotation angles
τ	Time intervall between pulses for DD
E	Macroscopic applied strain
n	Number of qubits
S	Number of iterations

Variables

υ	Grid points on RVE
μ	Scalar modulus
ε	Strain field

Contents

List of Acronyms **ix**

Nomenclature **xi**

List of Figures **xv**

List of Tables **xv**

1 Introduction **1**
1.1 Purpose and limitations 2
1.2 Outline 3

2 Theory **5**
2.1 Computational Homogenization 5
2.1.1 Representative Volume Elements 6
2.1.2 FFT-Based Computational Homogenization 6
2.2 Algorithmic Complexity 7
2.3 Mathematical Framework of Quantum Computation 8
2.4 Quantum Gates and Unitary Operations 9
2.4.1 Hadamard 9
2.4.2 Rotation 10
2.4.3 Controlled-NOT 10
2.4.4 Qubit Reordering (SWAP) 11
2.4.5 Universality 11
2.5 Quantum Representation of Numerical Algorithms 11
2.6 Quantum Hardware Implementations 12
2.6.1 Transpiling 12
2.6.2 NISQ Devices, Pulse Control, and Noise 12
2.6.2.1 Gate Errors 13

3 Methods **15**
3.1 Implementation Framework 15
3.1.1 Polynomial Encoding 15
3.1.2 Quantum Fourier Transform 16
3.1.3 Quantum Homogenization Algorithm 17
3.1.3.1 Model problem 18
3.1.3.2 Quantum algorithm 18

3.2 Simulation with Qiskit Aer . . . 20
3.3 Execution on IBM Hardware . . . 21
3.4 Evaluation Metrics . . . 23
3.4.1 Relative L_2 Error . . . 23
3.4.2 Shape Similarity . . . 24

4 Results and Discussion **27**
4.1 The 3-qubit Circuit . . . 27
4.2 2-qubits circuits . . . 29
4.2.1 Validation . . . 30
4.2.2 QPU runs . . . 30
4.3 Complexity . . . 37

5 Conclusion **41**

Bibliography **43**

A Supplementary material **I**
A.1 One-Dimensional FFT Formulation . . . I
A.2 Superconducting Quantum Computers . . . II
A.2.1 The Josephson Junction . . . III
A.2.2 The Transmon Circuit . . . III
A.2.3 Control, Coupling, and Readout . . . IV
A.2.4 Characterisation: T_1, T_2, and Fidelity . . . IV
A.2.5 IBM Processors . . . V
A.3 Circuits . . . V
A.3.1 Polynomial encoding . . . V
A.3.2 SWAP . . . VI
A.3.3 2-qubit circuit . . . VII
A.3.4 Transpiled and mapped circuit (sequential) . . . VIII

List of Figures

2.1 Illustration of the Bloch sphere 9
2.2 Illustration of a quantum circuit 11

3.1 Encoding of a polynomial function 16
3.2 Quantum Fourier transform 17
3.3 Stress computation with the circuit 20
3.4 DD (Hahn spin echo) illustration 23

4.1 The shear modulus $\mu(x)$ 28
4.2 Three qubits circuit solution 28
4.3 Two qubit circuit RVE solver (one iteration) 29
4.4 Aer simulator solutions 31
4.5 Ideal implementation relative L_2-norm error 32
4.6 Two qubits full circuit solution 32
4.7 Two qubits simplified circuit solution 33
4.8 Two qubit circuits relative errors 34
4.9 Two qubit circuits idle periods 36
4.10 Sequential two qubit circuit RVE solver 37
4.11 Two qubits sequential circuit solution 38
4.12 Two qubit sequential circuit relative errors 38
4.13 Complexity plot 39

List of Tables

2.1 FFT-based solver 7

4.1 Circuit depth 34
4.2 Basis gates' execution time 35
4.3 Time scales and gate errors of the solver's circuit 35

1

Introduction

Composite materials are becoming increasingly common across many applications thanks to their superior mechanical performance. Depending on the composition, orientation, and distribution of their constituents, they can deliver benefits such as weight reduction, improved tensile strength and stiffness. As a result, they are being adopted across numerous industries -including aerospace, automotive, and construction- , even more so because of growing cost and energy constraints [1, 2].

Short Fiber Reinforced Composites (SFRCs), in particular, are a widely used subtype that incorporates short fibers to enhance performance. They are versatile in use and well suited to injection molding. This enables faster and more economical production than continuous-fiber composites [3].

Manufacturing has a strong influence on SFRC behaviour. During processing, the viscous melt flow tends to align fibers with the flow direction, creating a preferred orientation. At the same time, the stresses imposed during processing can cause fiber breakage through contact with the surrounding environment or through fiber-fiber interactions. The resulting distributions of fiber lengths and orientations in injection-moulded parts strongly influence the mechanical response of the final component [4]. This opens the possibility of tailoring SFRC properties through design, provided that the effect of the resulting microstructure on the macroscopic response can be predicted accurately [5].

Directly resolving these local variations at the component scale is computationally inefficient. Homogenization schemes address this by replacing the heterogeneous material response with an effective response obtained from the properties of the constituents and relevant microstructural quantities [6].

One numerical method that does this is computational homogenization. It determines the macroscopic solution field by solving coupled problems at both macro and micro scales. The gradient of the macroscopic field provides the input to the microstructural problem, while the macroscopic fluxes are obtained by averaging the corresponding microscopic fluxes, thereby closing the multiscale loop [7]. This is usually done using numerical Representative Volume Elements (RVEs) to create statistical geometric realizations of the microstructure, combined with Finite Element

(FE)- or Fast Fourier Transform (FFT)-based computations to obtain an effective (homogenized) response [8, 9].

For two-component composites, the system comprises two phases: the fibers and the surrounding matrix. The RVE can be generated via a randomized fiber placement algorithm that targets a specified fiber volume fraction and orientation distribution. The RVE is usually assumed periodic, meaning it can be viewed as being surrounded by identical copies of itself. Periodic Boundary Conditions (PBCs) are then imposed to reduce boundary effects.

However, because each material point in the macroscopic finite-element model is associated with an RVE that resolves the underlying microstructure, solving all RVE problems simultaneously with the macroscopic finite-element analysis can be computationally very expensive [5]. This motivates interest in alternative computational approaches, including quantum computing.

In quantum computing, information is encoded in the state of a quantum system composed of (n) quantum bits(qubits). Each qubit is defined by two computational basis states -(0) and (1)- implemented using physical systems such as photons, trapped ions, or superconducting circuits. Unlike classical bits, which take values 0 or 1, qubits obey the laws of quantum mechanics and can exist in superpositions of the computational basis states. Consequently, the number of complex amplitudes required to describe the state of an (n)-qubit system scales exponentially with the number of qubits [10].

Classically, RVE problems can be solved efficiently using FFT-based methods. However, very large microstructures can still be resource intensive. A quantum counterpart based on the Quantum Fourier Transform (QFT), tackle this limitation by exploiting the dimensional advantage of the quantum representation, offering an exponential speed-up [11].

1.1 Purpose and limitations

The purpose of this work was to implement and run an RVE solver on a quantum computer. Today's noisy intermediate-scale quantum (NISQ) hardware imposes significant constraints: decoherence, dephasing, crosstalk between qubits, pulse-control imperfections, and other noise sources become more severe as circuit depth increases. To run on IBM's superconducting quantum processing units (QPUs), these challenges were addressed by adapting the quantum algorithm, and incorporating error-suppression techniques such as twirling and dynamical decoupling.

The project was limited to the implementation of a homogenization scheme in a one-dimensional setting. No physical experiments were performed. It was also outside the scope of this project to validate the developed model through classical solvers other than the FFT-method.

1.2 Outline

The following parts of the report are structured as follows. Section 2 introduces the theory of homogenization and quantum computing in the context of computational mechanics. Section 3 explains the mapped quantum algorithm and its implementation on the IBM hardware including the employed error suppression techniques. Section 4 presents the results obtained from these procedures and a comparative analysis with the classical algorithm. Section 5 provides related discussions and concludes this study.

2

Theory

Many engineering materials exhibit heterogeneous microstructures composed of multiple phases whose characteristic length scales are much smaller than those of the macroscopic structures in which the materials are used. This is the case for fiber-reinforced composites, polycrystalline metals, and porous materials. In such systems, the macroscopic response depends on the spatial distribution, geometry, and mechanical properties of the microscopic constituents.

2.1 Computational Homogenization

For a heterogeneous elastic medium occupying a domain $\Omega \subset \mathbb{R}^d$, the local mechanical state is described by the displacement field $\mathbf{u}(\mathbf{x})$, from which the strain tensor can be obtained as

$$\boldsymbol{\varepsilon}(\mathbf{x}) = \frac{1}{2}\left(\nabla\mathbf{u}(\mathbf{x}) + \nabla\mathbf{u}(\mathbf{x})^T\right). \tag{2.1}$$

The corresponding stress tensor $\boldsymbol{\sigma}(\mathbf{x})$ is related to the strain through the generalized Hooke's law

$$\boldsymbol{\sigma}(\mathbf{x}) = \mathcal{C}(\mathbf{x}) : \boldsymbol{\varepsilon}(\mathbf{x}), \tag{2.2}$$

where $\mathcal{C}(\mathbf{x})$ denotes the fourth order stiffness tensor, which varies in space due to the heterogeneous microstructure, and the **:** symbol refers to the double contraction $\sigma_{ij} = C_{ijkl}\varepsilon_{kl}$.

In the absence of body forces, mechanical equilibrium requires

$$\nabla \cdot \boldsymbol{\sigma}(\mathbf{x}) = 0. \tag{2.3}$$

Computational homogenization provides a model for determining effective material properties by solving boundary value problems on a domain that captures the statistical characteristics of the microstructure [6, 12].

For a given macroscopic strain tensor $\mathbf{E}$, the effective behaviour is defined through the volume averages

$$\langle\boldsymbol{\sigma}\rangle = \mathcal{C}^{\text{eff}} : \mathbf{E}, \tag{2.4}$$

where $\mathcal{C}^{\text{eff}}$ is the effective stiffness tensor, and the volume average operator is defined as

$$\langle f \rangle = \frac{1}{|\Omega|} \int_{\Omega} f(\mathbf{x})\, d\mathbf{x}. \tag{2.5}$$

2.1.1 Representative Volume Elements

The microscopic problem is typically solved on a Representative Volume Element (RVE), i.e., a domain chosen so that the relevant statistical properties of the microstructure are represented at the scale of interest [6, 13].

When periodic boundary conditions are assumed, the displacement field can be decomposed into a macroscopic and a fluctuating part,

$$\mathbf{u}(\mathbf{x}) = \mathbf{E}\mathbf{x} + \tilde{\mathbf{u}}(\mathbf{x}), \tag{2.6}$$

where $\mathbf{E}$ represents the imposed macroscopic strain and $\tilde{\mathbf{u}}(\mathbf{x})$ is a periodic displacement fluctuation satisfying

$$\tilde{\mathbf{u}}(\mathbf{x} + \mathbf{L}) = \tilde{\mathbf{u}}(\mathbf{x}).$$

This decomposition induces a corresponding decomposition of the strain field,

$$\boldsymbol{\varepsilon}(\mathbf{x}) = \mathbf{E} + \tilde{\boldsymbol{\varepsilon}}(\mathbf{x}), \tag{2.7}$$

with

$$\tilde{\boldsymbol{\varepsilon}}(\mathbf{x}) = \frac{1}{2}\left(\nabla\tilde{\mathbf{u}}(\mathbf{x}) + \nabla\tilde{\mathbf{u}}(\mathbf{x})^T\right). \tag{2.8}$$

The periodicity of the fluctuation implies that the volume average of the microscopic strains is equal to macroscopic strain:

$$\langle \boldsymbol{\varepsilon}(\mathbf{x}) \rangle = \mathbf{E}. \tag{2.9}$$

2.1.2 FFT-Based Computational Homogenization

For periodic media, the RVE problem can be solved efficiently using FFT-based methods [8, 9, 14, 15]. First, a reference medium with stiffness tensor $\mathcal{C}_0$ is introduced. The polarization field can then be defined as

$$\boldsymbol{\tau}(\mathbf{x}) = (\mathcal{C}(\mathbf{x}) - \mathcal{C}_0) : \boldsymbol{\varepsilon}(\mathbf{x}). \tag{2.10}$$

With this definition, the equilibrium problem can be written in Lippmann-Schwinger form [9, 15],

$$\boldsymbol{\varepsilon}(\mathbf{x}) = \mathbf{E} - \int_{\Omega} \boldsymbol{\Gamma}_0(\mathbf{x} - \mathbf{y}) : \boldsymbol{\tau}(\mathbf{y})\, d\mathbf{y}, \tag{2.11}$$

where $\boldsymbol{\Gamma}_0$ is the Green operator associated with the reference medium (a full derivation is given in [9, 15]).

The convolution structure of this equation is what makes the method efficient since it enables a numerical solution using Fourier transforms. In Fourier space, the relation becomes

$$\widehat{\boldsymbol{\varepsilon}}(\mathbf{0}) = \mathbf{E} \quad \text{and} \quad \widehat{\boldsymbol{\varepsilon}}(\mathbf{k}) = -\widehat{\boldsymbol{\Gamma}}_0(\mathbf{k}) : \widehat{\boldsymbol{\tau}}(\mathbf{k}) \quad \text{for} \quad \mathbf{k} \neq 0. \tag{2.12}$$

This formulation leads to the iterative FFT-based algorithm introduced by Moulinec and Suquet, which is summarized in Table 2.1.

Table 2.1: FFT-based solver: Iterative solution of the RVE, the discrete Fourier transform (DFT) is evaluated using the FFT routine.

Step	**Operation**	
1	Initialize $\boldsymbol{\varepsilon}(\mathbf{x})$ with $\mathbf{E}$	$\boldsymbol{\varepsilon}_0(\mathbf{x}) = \mathbf{E}$
2	Compute $\boldsymbol{\sigma}_i(\mathbf{x})$ and $\boldsymbol{\tau}_i(\mathbf{x})$	$\boldsymbol{\sigma}_i(\mathbf{x}) = \mathcal{C}_i(\mathbf{x}) : \boldsymbol{\varepsilon}_i(\mathbf{x}),$ $\boldsymbol{\tau}_i(\mathbf{x}) = (\mathcal{C}_i(\mathbf{x}) - \mathcal{C}_{0,i}) : \boldsymbol{\varepsilon}_i(\mathbf{x})$
3	Apply DFT	$\widehat{\boldsymbol{\tau}}_i(\mathbf{k}) = \mathcal{F}\{(\boldsymbol{\tau}_i(\mathbf{k})\}$
4	Compute $\widehat{\boldsymbol{\varepsilon}}_i(\mathbf{k})$	$\widehat{\boldsymbol{\varepsilon}}_i(\mathbf{0}) = \mathbf{E},$ $\widehat{\boldsymbol{\varepsilon}}_i(\mathbf{k}) = -\widehat{\boldsymbol{\Gamma}}_{0,i}(\mathbf{k}) : \widehat{\boldsymbol{\tau}}_i(\mathbf{k})$
5	Apply inverse DFT	$\boldsymbol{\varepsilon}_i(\mathbf{x}) = \mathcal{F}^{-1}\{\widehat{\boldsymbol{\varepsilon}}_i(\mathbf{k})\}$
6	Update the fields and check convergence	Back to step 2 with $i = i + 1$

2.2 Algorithmic Complexity

One of the main reasons why FFT-based homogenization is attractive for high-resolution periodic microstructures is that the use of the Fast Fourier Transform reduces the computational cost of the algorithm described in subsection 2.1.2.

When evaluating numerical algorithms, measuring only the execution time is insufficient. Runtime depends on the machine, compiler, memory usage, background processes, and hardware-specific effects. For meaningful comparisons, the computational cost of algorithms is therefore estimated by counting how the number of elementary operations grows with the input size [16]. The dominant scaling behaviour is more important than exact operation counts, which is expressed using asymptotic notation. If the number of operations is bounded by a constant multiple of a function $f(N)$ for sufficiently large input size N, the algorithm is said to have complexity $\mathcal{O}(f(N))$.

For instance, algorithms requiring $5N + 208$ or $23N + 4$ operations are both linear, $\mathcal{O}(N)$, since their cost grows proportionally to N for large problem sizes. Similarly a constant-time algorithm, $\mathcal{O}(1)$, requires a number of operations independent of N, and logarithmic factors, $\mathcal{O}(\log N)$, grow slowly compared with powers of N.

The FFT reduces the computational complexity of the DFT in the RVE solver to $\mathcal{O}(N \log N)$ for N discretization points [17]. The corresponding quantum operation is the Quantum Fourier Transform (QFT), introduced in subsection 3.1.2, which acts on an n-qubit register. It has complexity $\mathcal{O}(n^2)$, or $\mathcal{O}((\log N)^2)$ since the register represents $N = 2^n$ basis states [10]. Thus, the QFT based solver used in this work offers an exponential speed-up in the Fourier-transform steps [11].

2.3 Mathematical Framework of Quantum Computation

Quantum computation is described in terms of vectors and operators acting in complex Hilbert spaces [10]. For a single qubit, the state space is the Hilbert space $\mathcal{H} \cong \mathbb{C}^2$, where $\mathbb{C}^2$ represents the two dimensional complex vector space. Using Dirac notation, the computational basis states are written

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{2.13}$$

A general qubit state is then expressed as

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{2.14}$$

with $\alpha, \beta \in \mathbb{C}$ and normalization condition

$$|\alpha|^2 + |\beta|^2 = 1. \tag{2.15}$$

Measurement in the computational basis yields the outcomes 0 and 1 with probabilities $|\alpha|^2$ and $|\beta|^2$, respectively. Single-qubit states can also be represented geometrically on the Bloch sphere. In this representation the state

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle$$

corresponds to a point on the surface of a unit sphere parametrized by the angles θ and ϕ [10]. Single-qubit operations can therefore be interpreted as rotations of the quantum state on the Bloch sphere.

For a system of n qubits, the state space is given by the tensor product

$$\mathcal{H}_n = (\mathbb{C}^2)^{\otimes n}, \tag{2.16}$$

so that $\dim(\mathcal{H}_n) = 2^n$.

The computational basis of a multi-qubit system is formed by tensor products of the one-qubit basis states. For example, for two qubits one obtains

$$|00\rangle = |0\rangle\otimes|0\rangle, \qquad |01\rangle = |0\rangle\otimes|1\rangle, \qquad |10\rangle = |1\rangle\otimes|0\rangle, \qquad |11\rangle = |1\rangle\otimes|1\rangle. \tag{2.17}$$

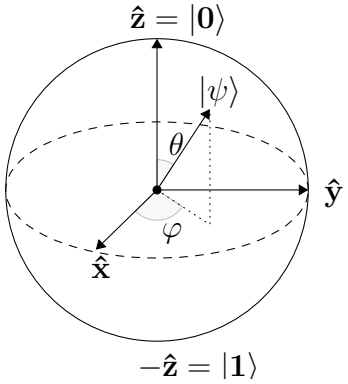


Figure 2.1: Illustration of the Bloch sphere: Single-qubit operations correspond to rotations of the quantum state.

More generally, an n-qubit state can be written as

$$|\Psi\rangle = \sum_{k=0}^{2^n-1} c_k |k\rangle, \tag{2.18}$$

where k is the number given by the binary representation formed by the qubits (e.g $|11\rangle = |3\rangle$). This tensor-product structure is central in quantum algorithms, since vectors defined on a numerical grid can be encoded in the amplitudes c_k of a multi-qubit state.

2.4 Quantum Gates and Unitary Operations

The evolution of a closed quantum system is described by unitary operators. A matrix U is unitary if

$$U^\dagger U = I. \tag{2.19}$$

Quantum algorithms are implemented as circuits built from such operations acting on a register of qubits (see Figure 2.2). If the register initially contains the state $|\psi_{\text{in}}\rangle$ and the circuit consists of gates $U_1, U_2, \ldots, U_m$, then the resulting state after the circuit is

$$|\psi_{\text{out}}\rangle = U_m \cdots U_2 U_1 |\psi_{\text{in}}\rangle. \tag{2.20}$$

A circuit therefore plays the same role for a quantum algorithm as a sequence of arithmetic operations does in a classical program.

2.4.1 Hadamard

A standard one-qubit unitary example is the Hadamard gate,

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \tag{2.21}$$

Applied to a basis state, it produces an equal superposition:

$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \qquad H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \tag{2.22}$$

Superposition plays a central role in many quantum algorithms by allowing information to be encoded across multiple computational paths.

2.4.2 Rotation

Another important one-qubit gate is the rotation gate, which rotates the state of a qubit about one of the axes of the Bloch sphere.

$$R_{\hat{\xi}}(\theta) \equiv e^{-i\frac{\theta}{2}\hat{\xi}} = \cos\left(\frac{\theta}{2}\right) I - i \sin\left(\frac{\theta}{2}\right) \xi \tag{2.23}$$

where θ is the rotation angle, I is the identity matrix, $\hat{\xi}$ is the rotation axis and ξ is the corresponding Pauli matrix:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{2.24}$$

.

This gate, when applied to the state $|0\rangle$, gives

$$R_x(\theta)|0\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle - i \sin\left(\frac{\theta}{2}\right)|1\rangle. \tag{2.25}$$

Thus, the rotation angle directly controls the amplitudes of the resulting state. Rotation gates therefore provide a convenient way to encode numerical quantities into quantum amplitudes.

2.4.3 Controlled-NOT

For multi-qubit systems, entangling operations can be used to generate correlations between qubits. A common example is the two-qubit gate *controlled-NOT*,

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{2.26}$$

When applied to a superposed control qubit, this operation can create entangled states. For example,

$$\text{CNOT}(H|0\rangle \otimes |0\rangle) = \frac{|00\rangle + |11\rangle}{\sqrt{2}}.$$

2.4.4 Qubit Reordering (SWAP)

For reversing the order of the qubits, *SWAP gates* are usually used. These gates exchange the states of two qubits,

$$|\psi\rangle|\phi\rangle \rightarrow |\phi\rangle|\psi\rangle.$$

Swap gates require several two-qubit operations and their implementation differs depending on the structure of the circuit.

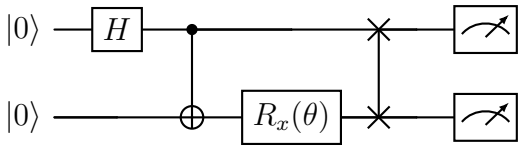


Figure 2.2: Illustration of a quantum circuit: The circuit is composed from left to right of the Hadamard-, controlled-NOT-, rotation- and Swap gates. The final measurement yields the states $|00\rangle$ and $|11\rangle$ with probability $\frac{\cos^2(\theta/2)}{2}$ each, and $|01\rangle$ and $|10\rangle$ with probability $\frac{\sin^2(\theta/2)}{2}$ each.

2.4.5 Universality

Sets of quantum gates capable of approximating any unitary operation are referred to as universal gate sets (e.g. an arbitrary single-qubit rotations gate with an entangling two-qubit gate such as the CNOT gate). These gates provide the basic building blocks for implementing arithmetic operations, Fourier transforms, and more general linear transformations on quantum states [10].

2.5 Quantum Representation of Numerical Algorithms

In this work, the relevance of quantum computation lies in the possibility of encoding numerical vectors into the amplitudes of a quantum state and evolving that state through a sequence of unitary operations to get an equivalent solution as a classical solver.

The quantum analogue of the discrete Fourier transform is the Quantum Fourier Transform (QFT) presented in subsection 3.1.2. As mentionned in section 2.2, it can be implemented with complexity $\mathcal{O}((\log N)^2)$, which is the complexity estimate commonly compared with the classical FFT cost $\mathcal{O}(N \log N)$ [10].

The main operations of the classical FFT-based homogenization procedure can thereby be interpreted in quantum-computational terms: field values are encoded in amplitudes, Fourier transforms are replaced by QFT-based circuits, and linear operations in Fourier space are implemented through gate sequences acting on the corresponding multi-qubit state [11].

2.6 Quantum Hardware Implementations

Several physical platforms have been developed for quantum computation, one example among many being superconducting circuits [18, 19]. In this case, the computational basis states are implemented physically and manipulated through externally applied control fields, but the platforms differ in their achievable coherence times, gate fidelities, connectivity, and scalability (see section A.2).

2.6.1 Transpiling

Quantum algorithms are described using abstract circuits built from a select gate set (e.g. Hadamard, phase, and CNOT gates). Actual hardware, however, only supports a limited set of native operations and a fixed qubit connectivity. Transpilation is the practical step that turns high-level language quantum algorithms into a hardware-specific implementation.

In practice, it means decomposing each abstract gate into the hardware's native basis gates and assigning logical qubits to physical qubits on the chip. If two qubits that must interact are not directly connected, additional routing operations -SWAP gates or equivalent decompositions- must be inserted (see Figure A.3). Finally, the resulting gate sequence is scheduled in time so that control pulses can be applied consistently with the device constraints.

2.6.2 NISQ Devices, Pulse Control, and Noise

Current processors operate in the Noisy Intermediate-Scale Quantum (NISQ) regime [20]. In this setting, circuit depth is limited by noise and by the absence of full fault-tolerant error correction.

Two characteristic time scales are commonly used to describe decoherence. T_1 is the relaxation time, corresponding to energy decay from the excited state $|1\rangle$ toward the ground state $|0\rangle$. T_2 is the dephasing time, which characterizes the loss of phase coherence between components of a superposition state. While relaxation affects state populations, dephasing destroys the phase information required for quantum interference.

In superconducting processors, gates are implemented through microwave pulse control. Pulse calibration, layout, and gate scheduling all affect the final circuit performance on present-day superconducting devices [19, 20].

This makes the transpilation step one of the main sources of performance loss on NISQ devices. Gate decompositions increase circuit depth, routing adds two-qubit gates (e.g. a SWAP is itself decomposed into entangling gates), and scheduling affects how long qubits remain idle. Since two-qubit gates are usually noisier than single-qubit gates, and idle times increase the impact of relaxation and dephasing, transpilation has a direct influence on the final error budget.

Crosstalk can also occur when control signals applied to one qubit affect neighboring qubits through residual couplings or control-line leakage. This can introduce correlated errors across the circuit and reduce its fidelity especially when working with poor scheduling.

2.6.2.1 Gate Errors

Gate quality is quantified through an average gate error obtained from calibration and benchmarking procedures. In randomized benchmarking, the qubit is initialized, a sequence of m random gates is applied, and a final inverting gate is added to return the qubit to its initial state. The circuit is then measured and the result is fitted to $A_0\alpha^m + B_0$, where A_0 and B_0 absorb state-preparation and measurement errors, while α is the decay parameter. This parameter is related to the average gate fidelity by

$$\mathcal{F} = \frac{(d-1)\alpha + 1}{d}, \tag{2.27}$$

where d is the Hilbert-space dimension [21]. The corresponding average gate error is then

$$p = 1 - \mathcal{F}. \tag{2.28}$$

This calibrated average gate error is used as the failure probability p_j of gate g_j, so that $p_j = P(\text{gate } g_j \text{ fails})$ and $1 - p_j = P(\text{gate } g_j \text{ succeeds})$.

Assuming independent gate errors and neglecting correlations and crosstalk, the probability that all gates m in a circuit succeed then simplifies to

$$P(\text{all gates succeed}) = \prod_{j=1}^{m}(1 - p_j). \tag{2.29}$$

This gives the probability of at least one gate error as

$$\mathcal{P} = P_{\exists\text{err,gate}} = 1 - \prod_{j=1}^{m}(1 - p_j). \tag{2.30}$$

For instance, on the device used in this work, circuits are expressed in the basis $\{SX, X, RZ, CZ\}$. X and RZ are the Pauli-X and z-axis rotation gates introduced in subsection 2.4.2. The single-qubit SX gate is the square root of X, with $SX^2 = X$, and the two-qubit CZ gate

$$\mathrm{CZ} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \tag{2.31}$$

introduces a conditional phase on the $|11\rangle$ state, providing an entangling operation.

For this basis,

$$\mathcal{P} = 1 - \prod_{\text{SX gates}} (1 - p_{\text{SX},q}) \prod_{\text{X gates}} (1 - p_{\text{X},q}) \prod_{\text{RZ gates}} (1 - p_{\text{RZ},q}) \prod_{\text{CZ gates}} (1 - p_{\text{CZ},(q_i,q_j)}). \tag{2.32}$$

A lower and upper estimate of $\mathcal{P}$ can be obtained by using the best and worst calibrated implementation of each gate type. This gives an interval $[\mathcal{P}_{\text{best}},\ \mathcal{P}_{\text{worst}}]$, computed from

$$\mathcal{P} = 1 - (1 - p_{\text{SX},q})^{\#\text{SX}} (1 - p_{\text{X},q})^{\#\text{X}} (1 - p_{\text{RZ},q})^{\#\text{RZ}} (1 - p_{\text{CZ},(q_i,q_j)})^{\#\text{CZ}}, \tag{2.33}$$

where $\#\zeta$ is the number of gates of type ζ in the transpiled circuit. For $\mathcal{P}_{\text{best}}$, the minimum calibrated error of each gate type is used, while for $\mathcal{P}_{\text{worst}}$, the maximum calibrated error is used.

3

Methods

3.1 Implementation Framework

All quantum circuits in this work were implemented using the *Qiskit* software framework. Qiskit provides tools to construct quantum circuits, simulate them on classical hardware, and execute them on IBM QPUs.

In Qiskit, circuits are written at an abstract level and then compiled for a specific backend using the *transpiler* as described in subsection 2.6.1. Once compiled, circuits can be executed through two interfaces known as *primitives*. The *Sampler* primitive executes the circuit repeatedly and returns measurement statistics, while the *Estimator* primitive evaluates expectation values of operators for the state produced by the circuit. When these primitives are used locally, the computation is carried out through classical simulation. When used through IBM runtime, the same interface submits the circuit to a physical QPU.

3.1.1 Polynomial Encoding

Arbitrary functions cannot be applied directly as quantum gates and must therefore be approximated. The functions used in homogenization are generally not polynomial and are instead represented by piecewise polynomials in the quantum formulation[11].

The domain of the function in k is partitioned into n_ω subintervals. On each subinterval, the target function is approximated by a polynomial $p_{n_w}(k)$.

The encoding of a polynomial $p(k)$ is achieved through a rotation of an ancillary qubit about the y-axis of the Bloch sphere. A rotation by an angle θ transforms the state $|0\rangle$ as

$$R_y(\theta)|0\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + \sin\left(\frac{\theta}{2}\right)|1\rangle. \tag{3.1}$$

To introduce a dependence on the register value k, the rotation angle is chosen as $\theta(k) = \delta\, p(k)$, where δ is a small scaling parameter. The operation acts on an

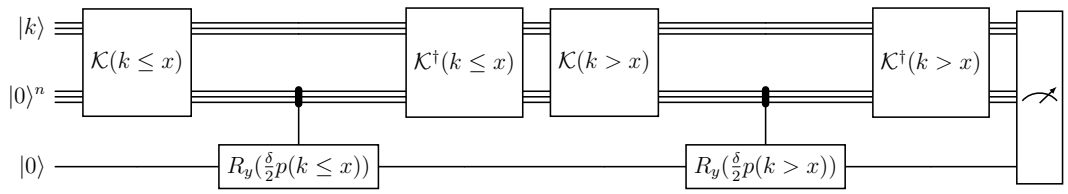


Figure 3.1: Encoding of a polynomial function: The comparator unitary $\mathcal{K}$ selects the states (e.g. $k > x$) which in turn control the y-rotations that encode the function. The ancillary qubit register $|0\rangle^n$ used for the selection process is reset between and after the computation.

ancilla qubit appended to the register. For a general state of the form given by Equation 2.18, this gives

$$U_P\left(\sum_k c_k|k\rangle|0\rangle\right) = \sum_k c_k|k\rangle\left(\cos\left(\frac{\delta p(k)}{2}\right)|0\rangle + \sin\left(\frac{\delta p(k)}{2}\right)|1\rangle\right)$$
$$\overset{\delta\ll 1}{\approx} \sum_k c_k|k\rangle|0\rangle + \sum_k \frac{\delta c_k}{2}p(k)|k\rangle|1\rangle. \tag{3.2}$$

To represent a general function, this construction is combined with comparator subcircuits which determine the interval to which k belongs. These circuits, composed of AND and OR operations, act on auxiliary qubits, which serve as control registers [22]. The corresponding polynomial unitary U_{P_i} is then applied conditionally on the active interval (see Figure 3.1). After the operation, the comparator ancillas are uncomputed and reset.

3.1.2 Quantum Fourier Transform

The classical Moulinec–Suquet homogenization algorithm relies on the Fast Fourier Transform (FFT) to move between spatial and spectral representations of the field variables. In the quantum formulation, this change of basis must be performed on the amplitudes of a quantum state. This operation is the *Quantum Fourier Transform* (QFT) [10].

For an n-qubit register, the computational basis states are labeled by integers

$$x \in \{0, \ldots, 2^n - 1\}.$$

The QFT is defined by its action on these basis states,

$$|x\rangle \mapsto \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i x k/2^n}|k\rangle. \tag{3.3}$$

This expression is identical to the kernel of the classical discrete Fourier transform. The circuit must therefore reproduce the phase factor $e^{2\pi i x k/2^n}$ using elementary quantum gates.

This is achieved by writing the integer k in binary form,

$$k = k_{n-1}2^{n-1} + k_{n-2}2^{n-2} + \cdots + k_0. \tag{3.4}$$

Substituting this into the Fourier phase gives

$$\frac{xk}{2^n} = \frac{xk_{n-1}}{2} + \frac{xk_{n-2}}{2^2} + \cdots + \frac{xk_0}{2^n}. \tag{3.5}$$

The exponential therefore factorizes,

$$e^{2\pi i x k/2^n} = \prod_{j=0}^{n-1} e^{2\pi i x k_j/2^{n-j}}. \tag{3.6}$$

With $k_j \in \{0, 1\}$, Equation 3.3 becomes

$$|x\rangle \overset{QFT}{\mapsto} \frac{1}{\sqrt{2^n}} \prod_{j=0}^{2^n-1} \left(|0\rangle + e^{2\pi i x/2^{n-j}}|1\rangle\right). \tag{3.7}$$

This makes it possible to construct the Fourier transform using gates controlled by the qubits storing binary digits of k. The construction begins with a Hadamard gate acting on the most significant qubit as seen in Equation 2.22. This operation creates the equal superposition required for the Fourier basis. Controlled phase rotations are then used to introduce the additional phase factors that depend on the remaining qubits.

A controlled rotation gate R_m applies the phase

$$R_m = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^m} \end{pmatrix}. \tag{3.8}$$

By applying these rotations between the qubits storing the binary digits of k, the circuit accumulates the phase contributions required by the Fourier kernel (see Figure 3.2). The final layer of the QFT circuit consists of swap gates, which reorder the qubits because the circuit naturally produces the Fourier coefficients in reversed binary order.

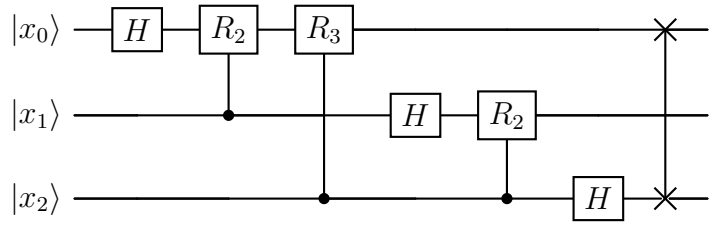


Figure 3.2: QFT: Three-qubit implementation of a QFT circuit.

3.1.3 Quantum Homogenization Algorithm

The quantum homogenization algorithm implemented in this work follows the construction proposed by Liu et al. [11].

3.1.3.1 Model problem

The model problem is a one-dimensional elastic composite rod with a periodic Young's modulus $E_Y(x)$, represented by a scalar modulus $\mu(x)$ on a one-dimensional RVE: $\Omega = (0, 1)$. A macroscopic strain E is applied, and the objective is to determine the corresponding microscopic stress field $\sigma(x)$ within the heterogeneous medium.

The tensorial formulation introduced in section 2 is therefore specialized to a scalar one-dimensional setting, while preserving the structure of the Moulinec-Suquet scheme (see section A.1). The stiffness tensor $\mathcal{C}(\mathbf{x})$ reduces to the scalar modulus $\mu(x)$, and the reference tensor $\mathcal{C}_0$ becomes the scalar reference modulus μ_0.

The polarization field introduced in Equation 2.10 then becomes

$$\tau(x) = \Big(\mu(x) - \mu_0\Big)\, \varepsilon(x), \tag{3.9}$$

and the convolution appearing in the Lippmann-Schwinger equations Equation 2.11 and Equation 2.12 simplifies in Fourier space to a mode-wise operation,

$$\hat{\varepsilon}(0) = E \quad \text{and} \quad \hat{\varepsilon}(k) = -\widehat{\Gamma}_0(k)\hat{\tau}(k) \quad \text{for} \quad k \neq 0, \tag{3.10}$$

with $\widehat{\Gamma}_0(k) = \frac{1}{\mu_0}$.

Using the computed strain field $\varepsilon(x)$ and the specialized forms of Equation 2.2 and Equation 2.4, the material response is obtained as

$$\sigma(x) = \mu(x)\varepsilon(x) \tag{3.11}$$

$$\mu^{\text{eff}} = \frac{\langle \sigma(x) \rangle}{E} \tag{3.12}$$

3.1.3.2 Quantum algorithm

The quantum algorithm retains the spectral structure of the classical Moulinec-Suquet solver but expresses each step as a quantum circuit on a system with three qubit registers. $|0\rangle^S$ for the applied strain, $|0\rangle^{2S}$ for the polynomial encodings, $|0\rangle^n$ which represents the field on the spatial grid. An additional register of $|0\rangle^n$ qubits is needed to delimit the domains of each field function but it is reset after every function encoding and doesn't impact the overall state of the system, it is therefore ignored in the derivations below. Here S is the number of iterations and $N = 2^n$ the number of field points.

The state preparation is performed in two steps. First, an auxiliary superposition is prepared over the ancillary register $|0\rangle^S$ (e.g. $|0\rangle^2 = |00\rangle$), whose basis states index the iteration and distinguish the applied strain branch from the stored mean strain branches,

$$\sqrt{1 - NSE^2}\, |0\rangle^n |0\rangle^S |0\rangle^{2S} + \sum_{s=0}^{S-1} \sqrt{N} E |0\rangle^n |s\rangle |0\rangle^{2S}, \tag{3.13}$$

where the states $|s\rangle$ (e.g. $|100\ldots\rangle$, $|010\ldots\rangle$, $\ldots$) form mutually orthogonal branches associated with each iteration. This allows the applied average strain to be stored separately for each step and later reinserted into the zero Fourier mode through a swap operation. This construction is required since the average strain cannot be duplicated across iterations due to the no-cloning theorem [10].

The initial state is then encoded with Hadamard gates over the physical register (encoding the spatial grid) in the applied strain branch, yielding a uniform strain field as the initial guess of the solver,

$$\sqrt{1-NSE^2}\sum_{\upsilon}\varepsilon_{\upsilon}|\upsilon\rangle|0\rangle^{S}|0\rangle^{2S}+\sum_{s=0}^{S-1}\sqrt{N}E|0\rangle^{n}|s\rangle|0\rangle^{2S}, \tag{3.14}$$

where $\varepsilon_{\upsilon}\equiv\varepsilon(x_{\upsilon})$ is the discretized strain field.

In the second step, the material contrast is introduced through the polynomial unitary $U_{p_{\upsilon}}$, using

$$p_{\upsilon}=\frac{\mu_{\upsilon}-\mu_0}{D},\qquad D=1-(\sqrt{N}\,E)^2, \tag{3.15}$$

so that the encoded field becomes the polarization field, consistent with Equation 3.9. Here, D is a normalization factor introduced to compensate for the amplitude encoding of the strain field, ensuring that the resulting amplitudes match the polarization field.

The Quantum Fourier Transform (QFT) is then applied on the physical register, mapping the state to its Fourier representation,

$$\sum_{\upsilon}\tau_{\upsilon}|\upsilon\rangle|0\rangle^{S}|1\rangle|0\rangle^{2S-1}\xrightarrow{\mathcal{F}(\tau_{\upsilon})}\sum_{k}\hat{\tau}_k|k\rangle|0\rangle^{S}|1\rangle|0\rangle^{2S-1}, \tag{3.16}$$

where $\tau_{\upsilon}\equiv\tau(x_{\upsilon})$ denotes the discretized polarization field.

The Fourier-space update is subsequently implemented with the polynomial unitary U_{P_i}, with $p_i=-\frac{1}{\mu_0}$, giving

$$\sum_{k}\hat{\tau}_k|k\rangle|0\rangle^{S}|1\rangle|0\rangle^{2S-1}\xrightarrow{U_{P_i}}\sum_{k}-\frac{1}{\mu_0}\hat{\tau}_k|k\rangle|0\rangle^{S}|11\rangle|0\rangle^{2S-2}. \tag{3.17}$$

A SWAP gate is then used to exchange the amplitude of the $k=0$ register with that of the auxiliary branch $|s\rangle$ corresponding to iteration s, thereby enforcing the prescribed macroscopic strain in the zero Fourier mode,

$$-\frac{\hat{\tau}_0}{\mu_0}|0\rangle|0\rangle^{S}|11\rangle|0\rangle^{2S-2}\xleftrightarrow{U_{SWAP}}\sqrt{N}E|0\rangle^{n}|s\rangle|0\rangle^{2S}. \tag{3.18}$$

The inverse QFT is finally applied to return to the spatial representation,

$$\sum_{k}\hat{\varepsilon}_k|k\rangle|0\rangle^{S}|11\rangle|0\rangle^{2S-2}\xrightarrow{\mathcal{F}^{\dagger}(\hat{\varepsilon}_k)}\sum_{\upsilon}\varepsilon_{\upsilon}^{\mathrm{new}}|\upsilon\rangle|0\rangle^{S}|11\rangle|0\rangle^{2S-2}. \tag{3.19}$$

This sequence corresponds to one iteration of the classical solver. Repeated application yields successive updates of the homogenization procedure and the final result is obtained by measuring on the appropriate register, i.e $|\upsilon\rangle|0\rangle^S|1\rangle^{2S}$.

The macroscopic stress is computed classically from Equation A.2 by taking its volume average. Alternatively, it can be obtained directly from the QPU computation by appending a polynomial unitary U_{p_s}, with $p_s = \mu(x)$, on an additional ancillary qubit. This maps the state to

$$\sum_{\upsilon} \sigma_{\upsilon}|\upsilon\rangle|0\rangle^S|1\rangle^{2S}|1\rangle. \tag{3.20}$$

Additional Hadamard gates are then applied to the physical register before measuring the branch $|0\rangle|0\rangle^S|1\rangle^{2S}|1\rangle$. This gives the macroscopic stress because the zero-frequency DFT component is proportional to the volume average:

$$y_{k=0} = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} y_j = \sqrt{N}\langle y\rangle, \tag{3.21}$$

$$\langle y\rangle = \frac{y_{k=0}}{\sqrt{N}}, \tag{3.22}$$

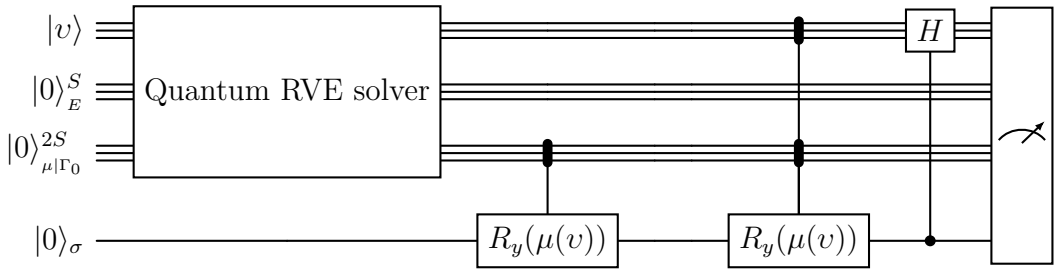


Figure 3.3: Stress computation: The modulus and QFT(for $k = 0$) are appended to the quantum RVE solver in order to get the macroscopic stress.

3.2 Simulation with Qiskit Aer

Before running the circuits on hardware, they were tested using the quantum simulator *Qiskit Aer*. A quantum simulator reproduces the action of a circuit on a classical computer by manipulating the mathematical representation of the quantum state with linear algebra operations (e.g. matrix-vector products and tensor-product structures).

In the Qiskit implementation, a backend is an object (in the programming sense) that executes the quantum circuit, either by running it on a physical device or by simulating its evolution on a classical computer, and returns the corresponding results. The *AerSimulator* backend provides several simulation methods, corresponding to different choices of state representation and numerical implementation [23].

For instance, the *matrix_product_state* method uses a compressed tensor representation and is efficient when entanglement remains limited [24]. The *density_matrix* method allows the inclusion of noise models by representing the state as a matrix ρ, where the state evolves as $\rho \to U\rho U^\dagger$. The *statevector* method can instead be selected for noiseless simulations, or for noisy simulations using stochastic sampling [23].

For the statevector method, the quantum state is represented with Equation 2.18 so that the action of the circuit is described by Equation 2.20. On a classical computer, this corresponds to successive matrix-vector operations acting on the 2^n-dimensional statevector, as implied by the unitary evolution of quantum states [10]. This method provides an exact simulation of the circuit in the absence of noise and was used in this work to validate the circuit implementation before running it on a QPU.

3.3 Execution on IBM Hardware

Hardware execution was performed through IBM's Qiskit runtime using the *SamplerV2* primitive, which submits circuits to IBM backends and retrieves statistics obtained from repeated executions of the circuit ("shots").

Because the experiments were carried out on NISQ hardware, three types of error-reduction strategies could in theory be considered: error correction, error mitigation, and error suppression. Quantum error correction was not used here because it would introduce a substantial overhead in both qubit and gate count, which would in turn increase the overall error on current hardware. Error mitigation was also not used in this work, since it involves heavy classical post-processing, which would defeat the purpose of this work by offsetting the speed-up obtained from the QFT [25, 26]. Thus, only error suppression was considered here.

After each circuit was transpiled for the selected processor, two error suppression techniques, *Pauli twirling* and *dynamical decoupling*, were used.

Pauli twirling is used to randomize *coherent gate errors*, i.e. systematic control errors such as repeated over- or under-rotations of the implemented gates relative to their ideal target operations. The noisy implementation of a gate is described by a quantum channel $\mathcal{E}$, i.e. a mapping from an input state ρ to the corresponding noisy output state. Mathematically, twirling replaces $\mathcal{E}$ by its Pauli-averaged form

$$\mathcal{E}_{\mathrm{t}}(\rho) = \frac{1}{|\mathcal{P}|} \sum_{P \in \mathcal{P}} P^\dagger \, \mathcal{E}(P \rho P^\dagger) \, P, \tag{3.23}$$

This is implemented by approximating Equation 3.23 with a randomly sampled subset of $\mathcal{P}$, i.e. by inserting randomly chosen Pauli gates before and after selected circuit operations, together with compensating Pauli updates, so that the ideal logical action of the circuit is unchanged.

The goal is to convert a systematic accumulation of coherent control errors into

an effective stochastic Pauli channel, which accumulates less destructively under repeated gate application [27].

Dynamical decoupling (DD) targets errors accumulated while a qubit is idle, i.e. during time intervals in which no gate is applied to it, but the qubit still undergoes unwanted evolution due to coupling to its environment. If the idle evolution during a time τ is generated by an effective noise Hamiltonian H_{n}, then

$$U_{\mathrm{idle}}(\tau) = e^{-iH_{\mathrm{n}}\tau}. \tag{3.24}$$

The principle of DD is to insert additional pulses whose net ideal action is the identity, but which reverse the sign of the error terms during the idle window [28]. For instance, under quasi-static dephasing, i.e. when the dephasing amplitude β can be treated as approximately constant over a single idle window,

$$H_{\mathrm{n}} = \frac{\beta}{2} Z, \tag{3.25}$$

so that the state evolves freely about the Z axis as

$$U_{\mathrm{idle}}(t) = e^{-i\beta Z t/2}. \tag{3.26}$$

DD can then be achieved by applying a pulse sequence, e.g. XX, which is the default sequence in IBM runtime for DD applications. It is defined by the schedule "wait $\tau/2$, apply X, wait τ, apply X, wait $\tau/2$", and it corresponds to the unitary

$$\begin{aligned} U_{\mathrm{DD}}(\tau) &= e^{-i\beta Z\tau/4}\, X\, e^{-i\beta Z\tau/2}\, X\, e^{-i\beta Z\tau/4} \\ &= e^{-i\beta Z\tau/4}\, e^{+i\beta Z\tau/2}\, e^{-i\beta Z\tau/4} \\ &= I. \end{aligned} \tag{3.27}$$

This means that the phase accumulated during the first half of the idle period is cancelled by the phase accumulated during the second half after the sign flip induced by the intermediate X pulse (see Figure 3.4).

In IBM's Qiskit runtime, DD is achieved by inserting scheduled gate sequences into idle windows. For this work, the built-in Sampler DD options were used with the *XX*, *XpXm*, and *XY4* sequences for different trials.

The *XX* sequence is a two-pulse echo-type sequence corresponding to

$$\xrightarrow{\tau/2} \quad X \quad \xrightarrow{\tau} \quad X \quad \xrightarrow{\tau/2}$$

The *XpXm* sequence similarly corresponds to

$$\xrightarrow{\tau/2} \quad X \quad \xrightarrow{\tau} \quad -X \quad \xrightarrow{\tau/2}$$

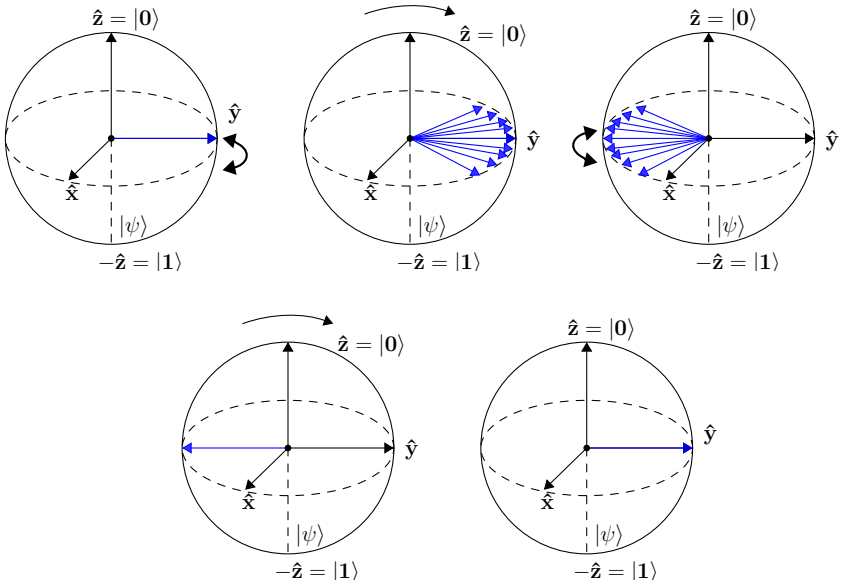


Figure 3.4: DD (Hahn spin echo) illustration: DD of a qubit in the $|y\rangle$ state. From left to right and top to bottom: the qubit first accumulates a phase during an idle time τ. The π rotation reverses the sign of the phase evolution, allowing the second idle period to cancel the first, while a final π rotation restores the original state.

and *XY4* to

$$\xrightarrow{\tau/2} \quad X \quad \xrightarrow{\tau} \quad Y \quad \xrightarrow{\tau} \quad -X \quad \xrightarrow{\tau} \quad -Y \quad \xrightarrow{\tau/2}$$

where τ denotes a waiting interval between pulses [29].

The final circuit was then run on the `ibm_aachen` backend (*Heron r3* processor). Different circuit models, shot counts, twirling parameters, DD pulse sequences and different types of scheduling were tested in order to reduce the observed error.

3.4 Evaluation Metrics

To evaluate the quality of the solutions obtained from the proposed algorithm, two measures were used. The relative L_2 error was used as a quantitative measure of the difference between the computed and reference strain fields, while a shape-similarity metric was used to compare the profile of the reconstructed field.

3.4.1 Relative L_2 Error

The relative L_2 error is sensitive to large local discrepancies. This makes it relevant for the strain-field reconstruction since QPU results can contain noisy entries, which should not be hidden by the correctly recovered points. The reference solution was obtained with the classical FFT solver.

For a discretized field with N grid points, the error is defined as

$$e_{L_2} = \sqrt{\frac{\sum_{i=1}^{N} |\varepsilon_i^{\mathrm{q}} - \varepsilon_i^{\mathrm{t}}|^2}{\sum_{i=1}^{N} |\varepsilon_i^{\mathrm{t}}|^2}}, \tag{3.28}$$

where $\varepsilon_i^{\mathrm{q}}$ is the value reconstructed from the quantum circuit and $\varepsilon_i^{\mathrm{t}}$ is the true value.

The resulting quantity is dimensionless and can be expressed as a percentage. It gives a measure of the overall error across the RVE and is used to compare solutions obtained from different implementations of the algorithm and hardware configurations.

3.4.2 Shape Similarity

The relative L_2 error does not separate amplitude errors from profile errors. Consequently, a shifted reconstructed field with the correct profile can still give a large L_2 error. It was therefore complemented by a shape-similarity metric for the QPU results.

Since the expected solution has a two-plateau structure, the metric compares the reconstructed field y to the reference field r through the jump direction, plateau smoothness, jump localization, oscillatory behaviour, and relative amplitude.

Numerically, the discontinuity location is determined from the reference field using zero-based indexing,

$$s = 1 + \arg \max_{0 \le i < N-1} |r_{i+1} - r_i| \,. \tag{3.29}$$

The same index is then used to split both fields into a left L and right R plateau. For a generic field z, the jump is computed as

$$J_z = \mathrm{mean}(R_z) - \mathrm{mean}(L_z). \tag{3.30}$$

Its direction and amplitude are given by $D_z = \mathrm{sign}(J_z)$ and $A_z = |J_z|$.

Plateau roughness is measured by

$$Q_z = \frac{\sigma(L_z) + \sigma(R_z)}{A_z}, \tag{3.31}$$

where σ denotes the standard deviation. The division by A_z makes the roughness relative to the step height.

To penalise staircase-like behaviour, the metric compares the total variation of the field T_z with the variation B_z located at the expected discontinuity. The result

extra-step score is

$$\begin{aligned} T_z &= \sum_{i=0}^{N-2} |z_{i+1} - z_i| \\ B_z &= |z_s - z_{s-1}| \\ E_z &= 1 - \frac{B_z}{T_z}. \end{aligned} \tag{3.32}$$

For a clean single step, most of the variation occurs at the expected discontinuity, so $B_z \simeq T_z$ and E_z is close to zero. If the field contains several jumps, E_z increases.

Oscillatory behaviour is measured from the local increments

$$\Delta z_i = z_{i+1} - z_i, \tag{3.33}$$

with the reverse-movement score defined as

$$\nu_z = \frac{1}{T_z} \sum_{\text{sign}(\Delta z_i) = -D_z} |\Delta z_i| \tag{3.34}$$

This penalises local variations that move opposite to the dominant step direction.

The total difference between the reference and reconstructed fields can then be computed as

$$\delta = |D_r - D_y| + |Q_r - Q_y| + w_E|E_r - E_y| + w_\nu|\nu_r - \nu_y| + w_A \left| \log\left(\frac{A_r}{A_y}\right) \right|. \tag{3.35}$$

The weights w_E, w_ν, and w_A control the contribution of extra-step structure, reverse movement, and amplitude mismatch. A logarithmic scaling is used for the amplitude term to measure relative, rather than absolute, amplitude mismatch.

Finally, the total difference score is mapped between 0 and 1:

$$\text{similarity} = \frac{1}{1 + \frac{\delta}{h}}. \tag{3.36}$$

Here, the calibration parameter h sets the scale at which fields are considered dissimilar.

4

Results and Discussion

In this section, the implementation of the algorithm is validated and the results obtained from the different circuit models and hardware configurations are presented. As discussed in subsection 3.1.3, the circuits solve a one-dimensional RVE problem on the domain $\Omega = (0, 1)$. The modulus was defined by the step function

$$\mu(x) = \begin{cases} 150, & x \leq 0.5 \\ 65, & x > 0.5 \end{cases} \tag{4.1}$$

which is shown in Figure 4.1. The reference modulus $\mu_0 = \langle \mu(x) \rangle$ and the macroscopic strain $E = 0.12$. The stress field $\sigma(x)$ is obtained from the computed local strain field using Equation 3.11, and its volume average gives the macroscopic stress $\langle \sigma(x) \rangle$.

The labels *2-qubit* and *3-qubit circuits* refer to the number of qubits used to represent the physical grid. A grid with $N = 2^n$ points requires n qubits for the spatial register. Therefore, the 2-qubit circuits correspond to $N = 4$ grid points, while the 3-qubit circuit corresponds to $N = 8$ grid points.

4.1 The 3-qubit Circuit

The implemented solver for a grid with $N = 8$ produced a solution that converged towards the reference field when run on the noiseless Aer simulator, as shown in Figure 4.2. The reference solution was obtained using a classical FFT solver with four iterations.

Subsequent executions on the QPUs resulted in a quasi-uniform random distribution, which can be seen in the same figure. This indicates that the computational-basis information encoded in the qubits was completely lost during execution. The result is consistent with strong effective noise accumulation over the circuit depth. While gate errors contribute to this behaviour, the results obtained from the different circuit models and hardware configurations suggest that decoherence and possible depolarization effects are important factors as well.

This discretization was therefore considered too fine for the available hardware.

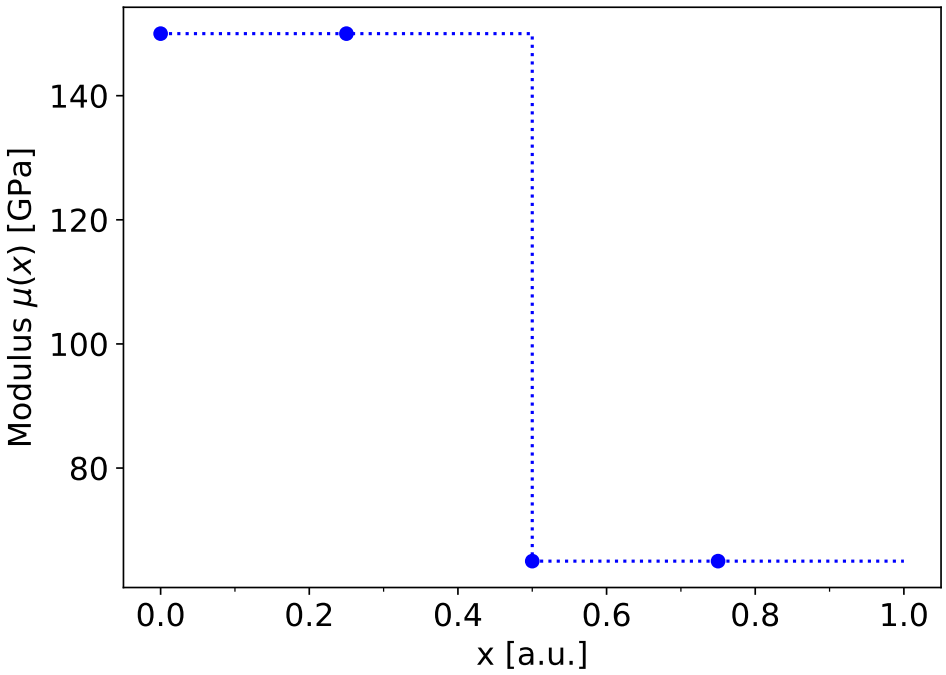


Figure 4.1: The RVE: Periodic homogenization of a one-dimensional RVE with a domain $\Omega = (0, 1)$. The shear modulus $\mu(x)$ was described by a piecewise constant function.

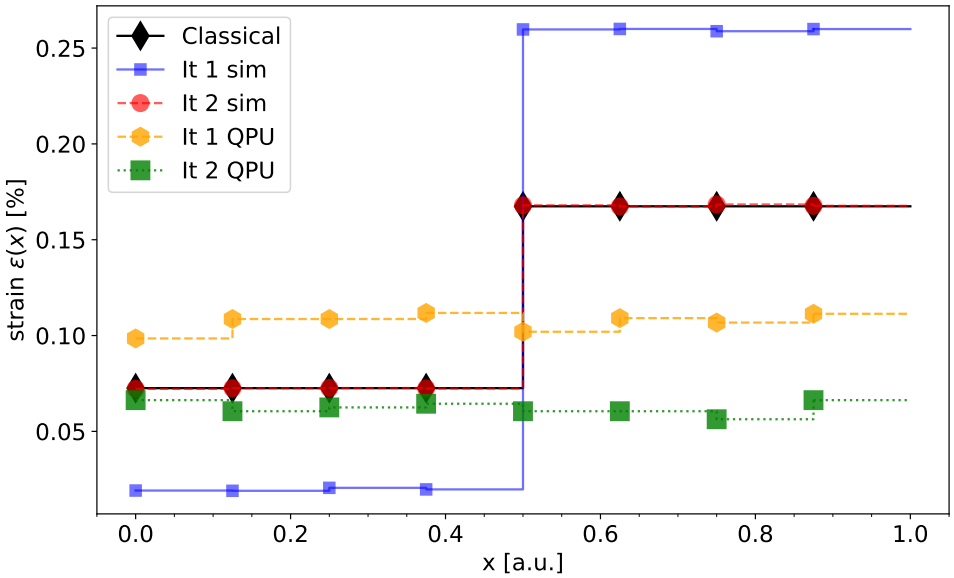


Figure 4.2: Three qubits solution: The solved local strain field $\varepsilon(x)$ of the one-dimensional RVE with a domain $\Omega = (0, 1)$. The grid was discretized to $N = 8$ points (3 qubits). The solutions were obtained with the classical FFT algorithm, one and two iterations circuits from the Aer simulator and QPU runs on `ibm_aachen`.

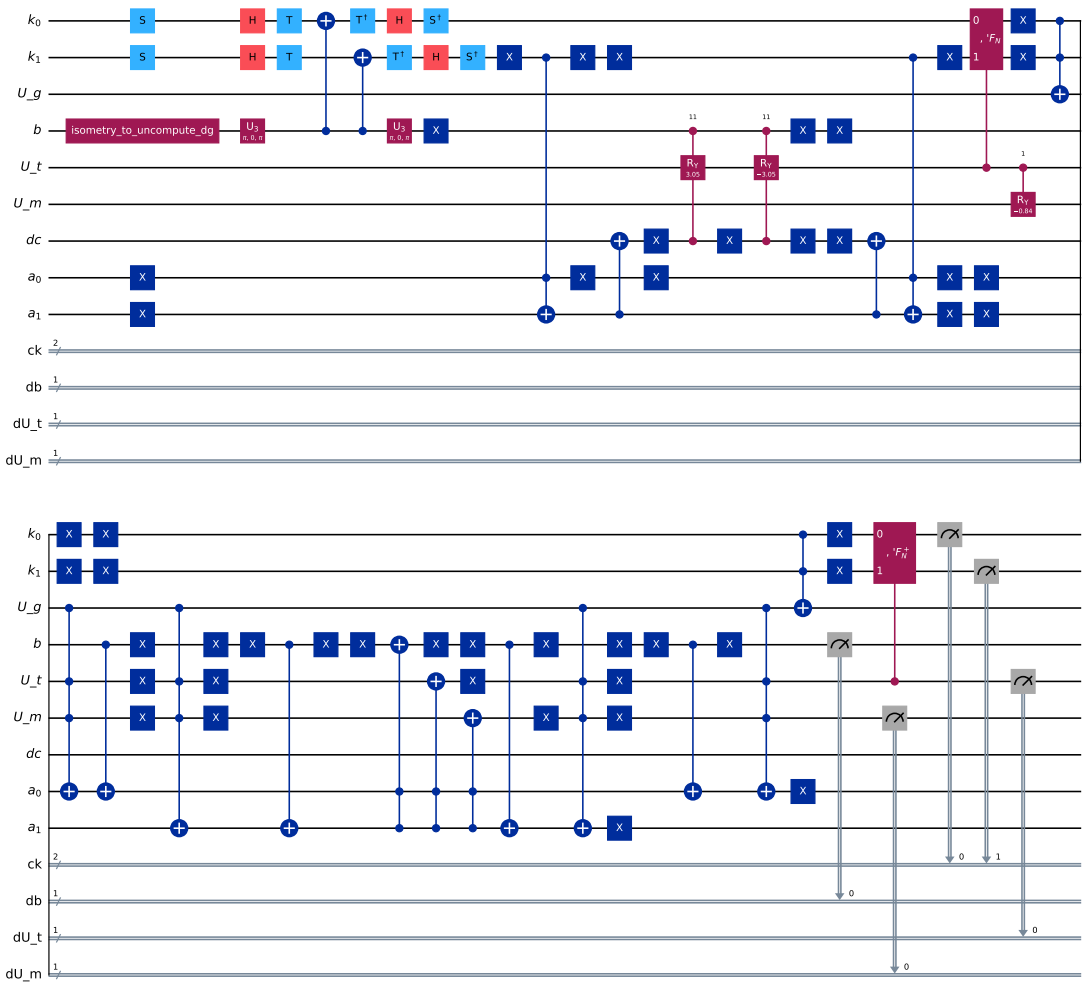


Figure 4.3: Two qubit circuit RVE solver (one iteration): From top to bottom and left to right, quantum circuit realization of the algorithm described in subsection 3.1.3, with two qubits describing the physical grid (discretization N=4). $|k_i\rangle$ represent the grid, $|a_i\rangle$, $|dc\rangle$, $|U_g\rangle$, the ancilla qubits, $|b\rangle$ the applied strain, $|U_t\rangle$ the material contrast and $|U_m\rangle$ the Fourier-space update.

Although the physical grid only requires three qubits, the corresponding circuit contains several multi-controlled operations. These operations increase the circuit depth directly and also lead to additional routing operations after transpilation and mapping to the physical qubits layout, as shown in Table 4.1.

4.2 2-qubits circuits

A coarser physical grid with $N = 4$ requires only two qubits to represent the spatial register. This reduces the number of controlled operations, especially in the zero-mode swap enforcing the macroscopic strain condition between the Fourier-transform steps, as shown in Figure 4.3. The resulting circuits are therefore significantly smaller than the 3-qubit circuit.

Three circuit models were studied: a *full* circuit, a *simplified* circuit, and a *sequential* circuit.

The *full* circuit implements the algorithm described in subsection 3.1.3, as in the previous section, but on the coarser grid (from $N = 8$ to $N = 4$).

The *simplified* circuit uses the geometry of the RVE to simplify the polynomial encoding. Instead of using comparator circuits and the corresponding Toffoli gates (CCX), the encoding is implemented with one R_y rotation and one-qubit controlled R_y rotation (see subsection A.3.1). The QFT block is also replaced by a custom implementation in which the final swap gates (see subsection 3.1.2) are omitted. This does not affect the resulting field, because the zero-mode operation acts only on $k = 0$, which is unaffected by the bit reversal, while the remaining ordering effects cancel between the QFT and its inverse. Finally, the zero-mode swap is simplified by using a modified algorithm with ancillary qubits and intermediate steps (see Figure A.2).

The *sequential* circuit implements a single iteration at a time. The output of one run is reconstructed classically and then re-encoded as the input state for the next run. Since the circuit contains only one iteration, there is no need to store several copies of the prescribed average strain in auxiliary branches, which simplifies the state preparation to a single R_y rotation as shown in Figure 4.10.

These changes led to significant differences in circuit depth, especially in the transpiled and mapped circuits, as shown in Table 4.1.

4.2.1 Validation

The circuits corresponding to the different configurations were first validated using the noiseless Aer simulator. This shows the behaviour of each circuit under ideal conditions, and verifies whether the implemented transformations produce the expected solution and convergence behaviour. This validation step is particularly important for the *simplified* and *sequential* circuits, since both modify the original quantum algorithm.

Figure 4.4 and Figure 4.5 show that the local strain field solution from all three circuits converges towards the true strain field. The *full* and *simplified* circuits reach relative errors below 1% after two iterations, while the *sequential* circuit requires three runs to reach the same level.

4.2.2 QPU runs

The hardware results are presented in this section. Figure 4.6 shows the results for the *full* circuit. The computations were performed using the different error-suppression strategies described in section 3.3. None of the tested configurations produced a clearly converging solution.

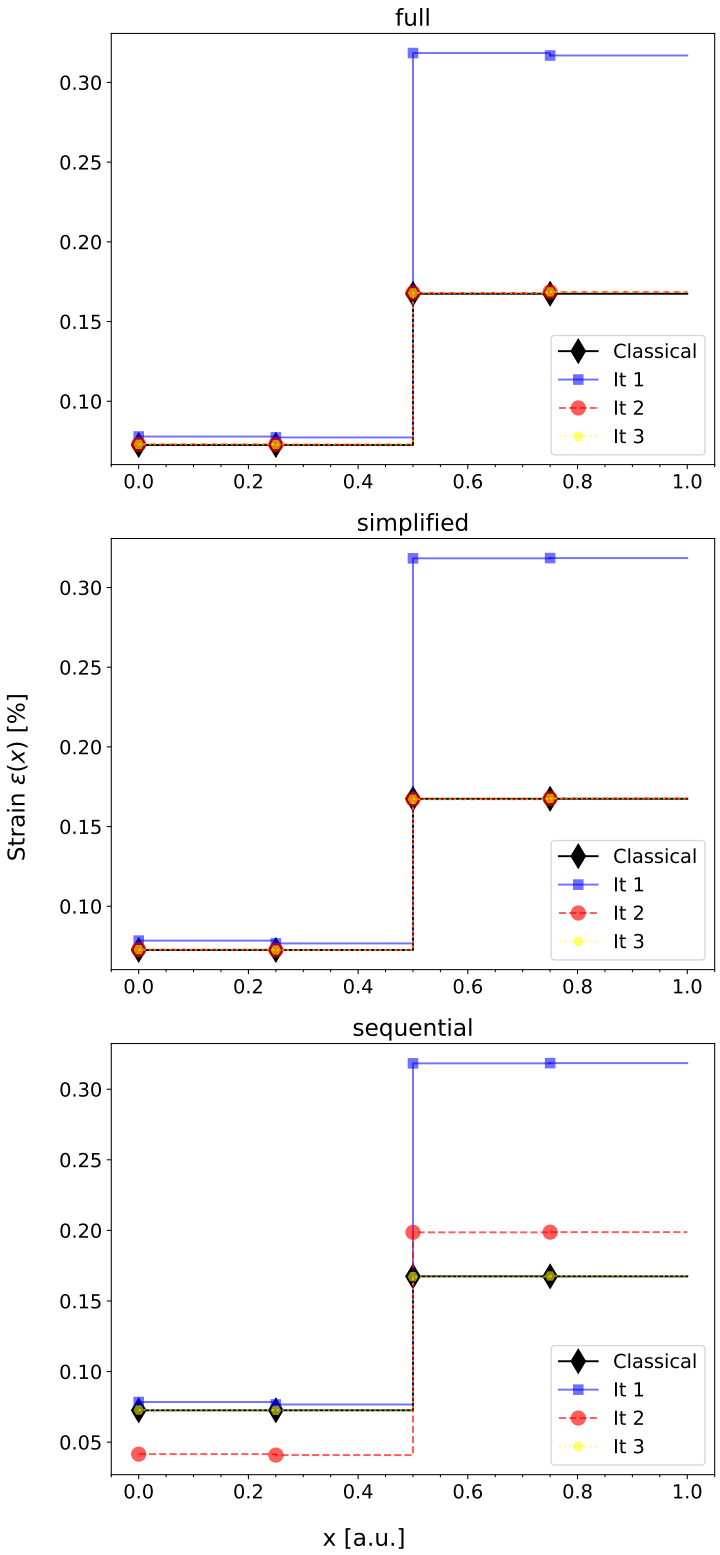


Figure 4.4: Aer simulator solutions: The solved local strain field $\varepsilon(x)$ of the one-dimensional RVE with a domain $\Omega = (0, 1)$. The grid was discretized to $N = 4$ points (2 qubits). The solutions were obtained with the classical FFT algorithm, the full and simplified one, two and three iterations circuits, as well three iterations runs of the sequential circuit on the Aer simulator.

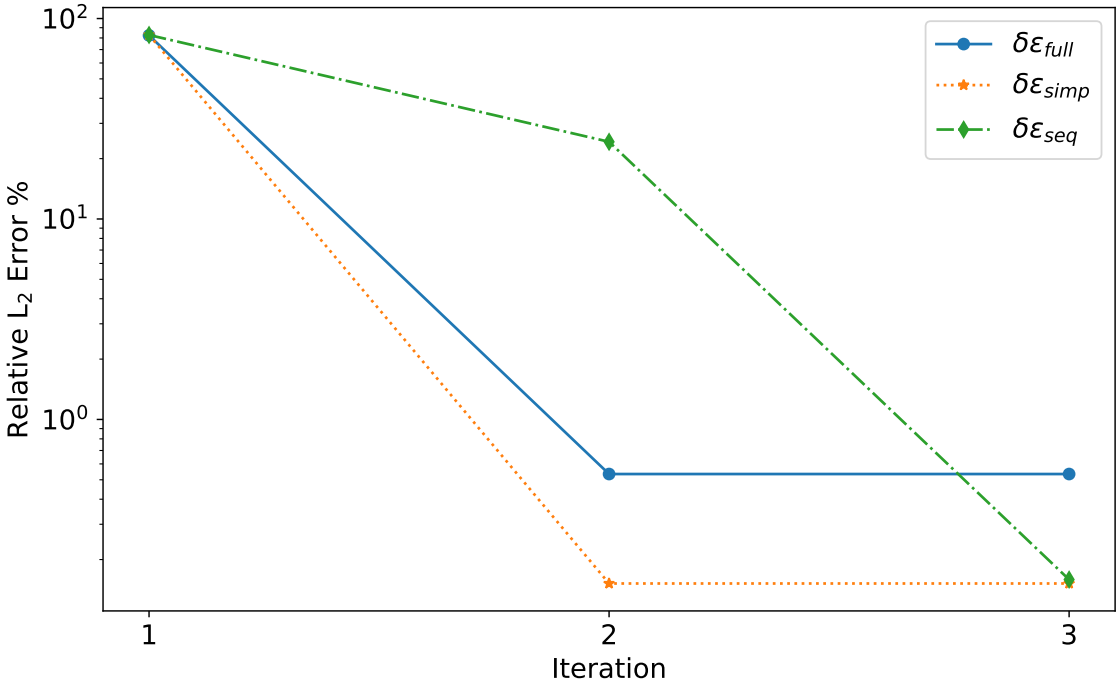


Figure 4.5: Ideal implementation relative L_2-norm error: Convergence of the relative L_2-norm error from running the full-, simplified and sequential circuits on the Aer simulator.

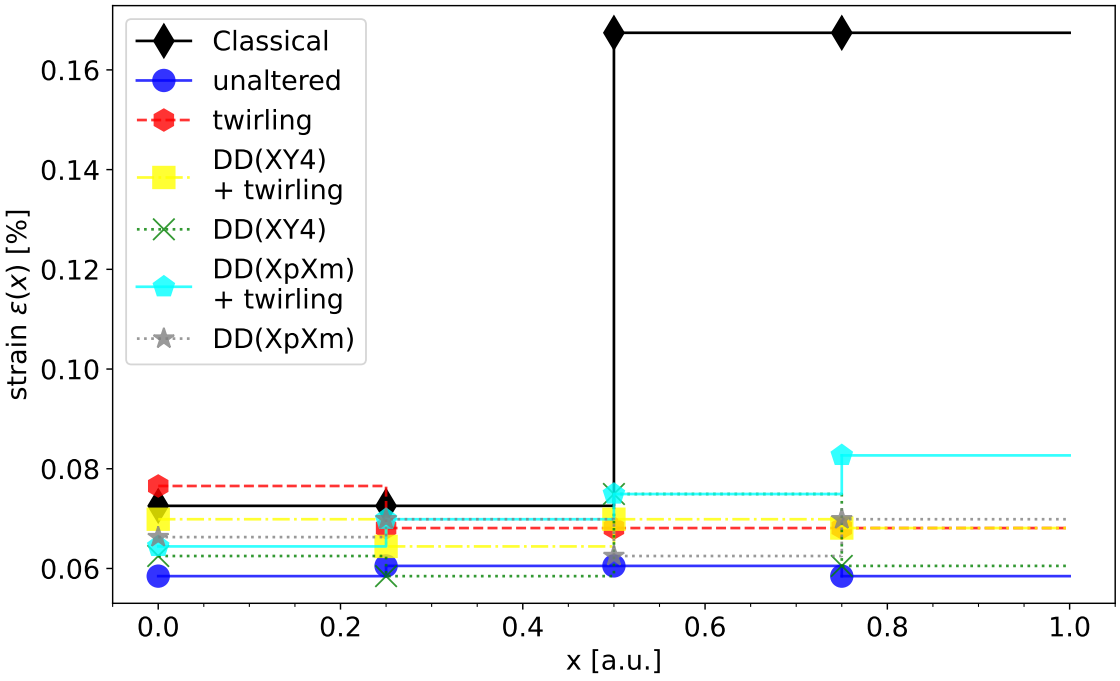


Figure 4.6: Two qubits full circuit solution: The solved local strain field $\varepsilon(x)$ of the one-dimensional RVE with a domain $\Omega = (0, 1)$. The grid was discretized to $N = 4$ points (2 qubits). The solutions were obtained with the classical FFT algorithm and two iterations full circuits runs on a QPU (`ibm_aachen`) with different error suppression techniques applied.

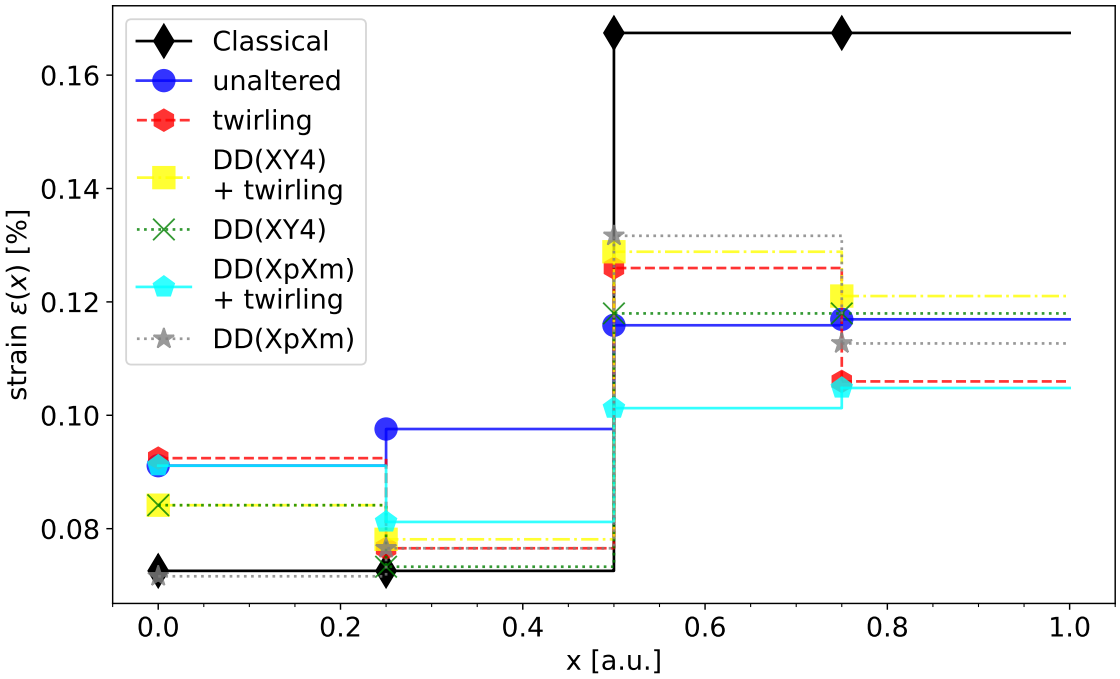


Figure 4.7: Two qubits simplified circuit solution: The solved local strain field $\varepsilon(x)$ of the one-dimensional RVE with a domain $\Omega = (0, 1)$. The grid was discretized to $N = 4$ points (2 qubits). The solutions were obtained with the classical FFT algorithm and runs of the simplified circuit with two iterations circuits on a QPU (`ibm_aachen`) with different error suppression techniques applied.

The results show a strongly noisy computation, with relative L_2 errors around 55% on average and a shape similarity of approximately 0.48, as shown in Figure 4.8. The error profile of the *full* circuit shows that both twirling and dynamical decoupling offer a small improvement, hinting that coherent-, relaxation- and dephasing errors might be impacting the results.

For the *simplified* circuit, the local strain field shown in Figure 4.7 is confined in a closer region to the reference solution and maintains a closer shape to it. An improvement is directly visible (*unaltered* step curve) indicating the impact of the decreased circuit depth. Further improvements can be observed when dynamical decoupling and twirling are used. This points to coherent errors from imperfect gates and dephasing effects when the qubits are idle. The circuit has an average relative L_2 error and shape similarity of 29% and 0.82 respectively, as shown in Figure 4.8. As *XpXm* doesn't suppress errors along the X-axis, while *XY4* suppresses all single-qubit Pauli error components to first order, the results suggests that the relevant noise is transverse to the X-axis. This is especially problematic for the polynomial encoding and the QFT, since both rely on preserving coherent phase relationships.

The macroscopic stress computed on the QPU was 74.19 MPa (−31.83%) for the *full* circuit and 99.54 MPa (−8.55%) for the *simplified* circuit, relative to the reference macroscopic stress of 108.84 MPa obtained classically using Equation 3.11. The

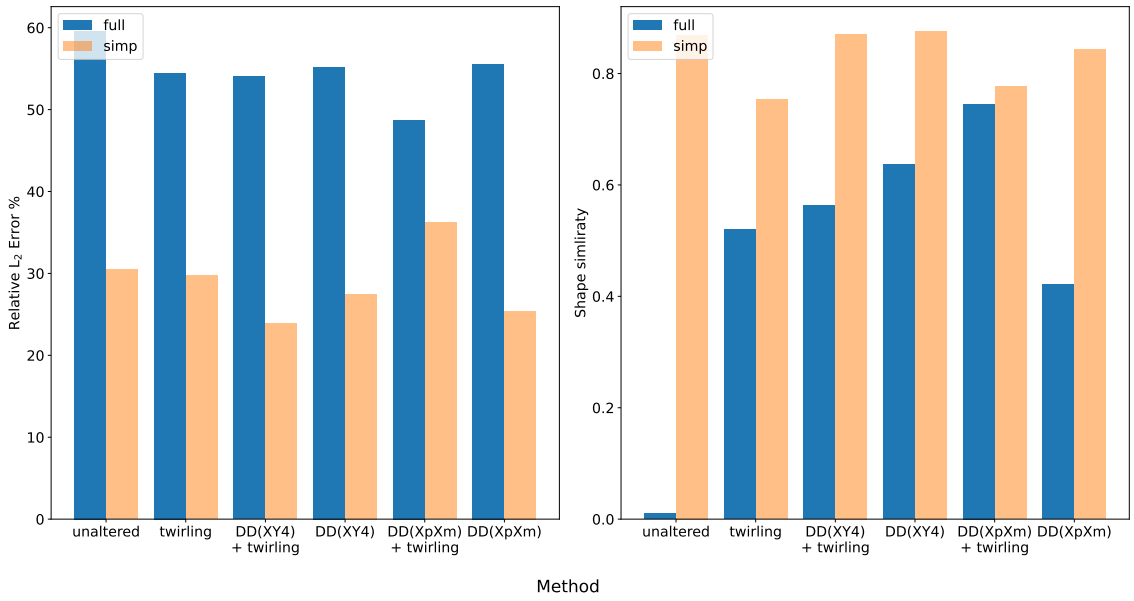


Figure 4.8: Two qubit circuits relative errors: Relative L_2-norm errors (left) and shape similarity with the true solution (right) from runs on the QPU (`ibm_aachen`) under configurations with and without error suppression techniques applied.

errors are smaller than for the local strain field since the volume average is obtained through a projection on the zero-frequency mode, it is therefore unaffected by the errors in orthogonal modes.

Table 4.1: Circuit depth: Depth and operation count of the four studied models before and after transpilation and mapping to the physical qubits.

Circuit	#-total qubits	circuit depth		Operation counts			
		pre-transpiling	post-transpiling	SX	RZ	CZ	X
Full (3-qubits)	14	88	3712	3496	1764	1742	61
Full (2-qubits)	12	81	2977	2503	1339	1258	50
Simp(2-qubits)	9	35	581	469	306	228	21
Seq (2-qubits)	6	19	212	165	106	80	6

Since the deeper circuits contain more two-qubit gates with finite fidelities, gate errors become increasingly important. Using the calibrated gate errors in Table 4.3, the probability of at least one gate error per circuit execution was estimated as described in subsubsection 2.6.2.1. The resulting bounds were $[0.835, 0.99996]$ for the 3-qubit circuit, $[0.754, 0.985]$ for the 2-qubit full circuit, $[0.246, 0.558]$ for the simplified circuit, and $[0.094, 0.164]$ for the sequential circuit. These estimates show that the accumulated gate-error exposure is very high for the full circuits, is significantly improved for the simplified circuit, and is further reduced for the sequential circuit.

Table 4.2: Basis gates' duration time (ns)**:** `ibm_aachen` basis gates and their execution time on the physical qubits presented in Table 4.3 and Figure 4.9. RZ is a virtual gate implemented via frame change and has therefore no duration [30, 31].

	I	RZ	SX	X	CZ
Duration time (ns)	32	0	32	32	68

During controlled operations such as the QFT and the zero-mode swap, scheduling constraints cause some qubits to remain inactive while gates are applied to others. During such periods, these qubits remain exposed to relaxation, dephasing, and unwanted interactions. The effect is especially relevant for two-qubit gates, which are both slower and less reliable than single-qubit gates, as shown in Table 4.2 and Table 4.3. They can therefore leave other qubits idle for a significant time relative to their T_1 and T_2 values.

The improvements obtained with dynamical decoupling suggests that these idle-time errors also contribute significantly to the degradation of the encoded information. This is consistent with Table 4.1, which shows the increase in circuit depth and operation count after transpilation and mapping to the physical qubits, and with Table 4.2 and Table 4.3, which provide the relevant QPU time scales.

Table 4.3: Characteristic time scales (μs) **and gate errors:** Relaxation time T_1 and dephasing time T_2 of the simplified two iteration circuit after transpiling and mapping to physical qubits. Gate errors of the circuit: RZ is virtual, so $RZ_{err} = 0$. $SX_{err} = X_{err}$, the subscript in CZ_q denotes the physical control qubit on the chip (note that $CZ_{err|1\to2} = CZ_{err|2\to1}$).

Qubit	k_0	k_1	b_0	b_1	U_{m_0}	U_{m_1}	U_{t_0}	U_{t_1}	U_g
$T1(\mu s)$	243.29	216.77	182.58	195.94	176.92	137.26	180.87	218.46	191.15
$T2(\mu s)$	251.33	253.64	220.29	183.13	227.20	92.53	194.03	110.95	240.65
$X_{err} \cdot 10^4$	1.50	2.13	1.36	1.41	2.14	1.43	2.14	2.38	1.68
$CZ_{err\|q} \cdot 10^4$	$12.99_{U_{m,0}}$	$14.24_{U_{t,0}}$	9.40_{k_0}	11.72_{b_0}	$30.47_{U_{t,1}}$	$13.49_{U_{t,1}}$	14.24_{k_1}	$15.80_{U_{m,1}}$	17.03_{b_1}

Figure 4.9 shows that the number and duration of idle intervals decrease from the *full* circuit to the *simplified* and *sequential* circuits. The *full* circuit contains 782 idle intervals with durations reaching approximately 61 μs. The *simplified* circuit reduces the number of intervals to 153 and caps their durations to 14 μs. The *sequential* circuit contains even fewer idle intervals (57), with durations up to 4 μs. This agrees with the observation that reduced circuit depth and the application of dynamical decoupling improve the results.

The *sequential* circuit results in Figure 4.11 show a significant improvement, with a strain field that converges towards a region closer to the true solution and its shape. The average relative L_2 error is 25.00%, 11.00% on the second run, as shown in Figure 4.12, which is a sharp improvement compared with the other circuits. The shape similarity remains stable over the iterations with an average of 0.88, thereby

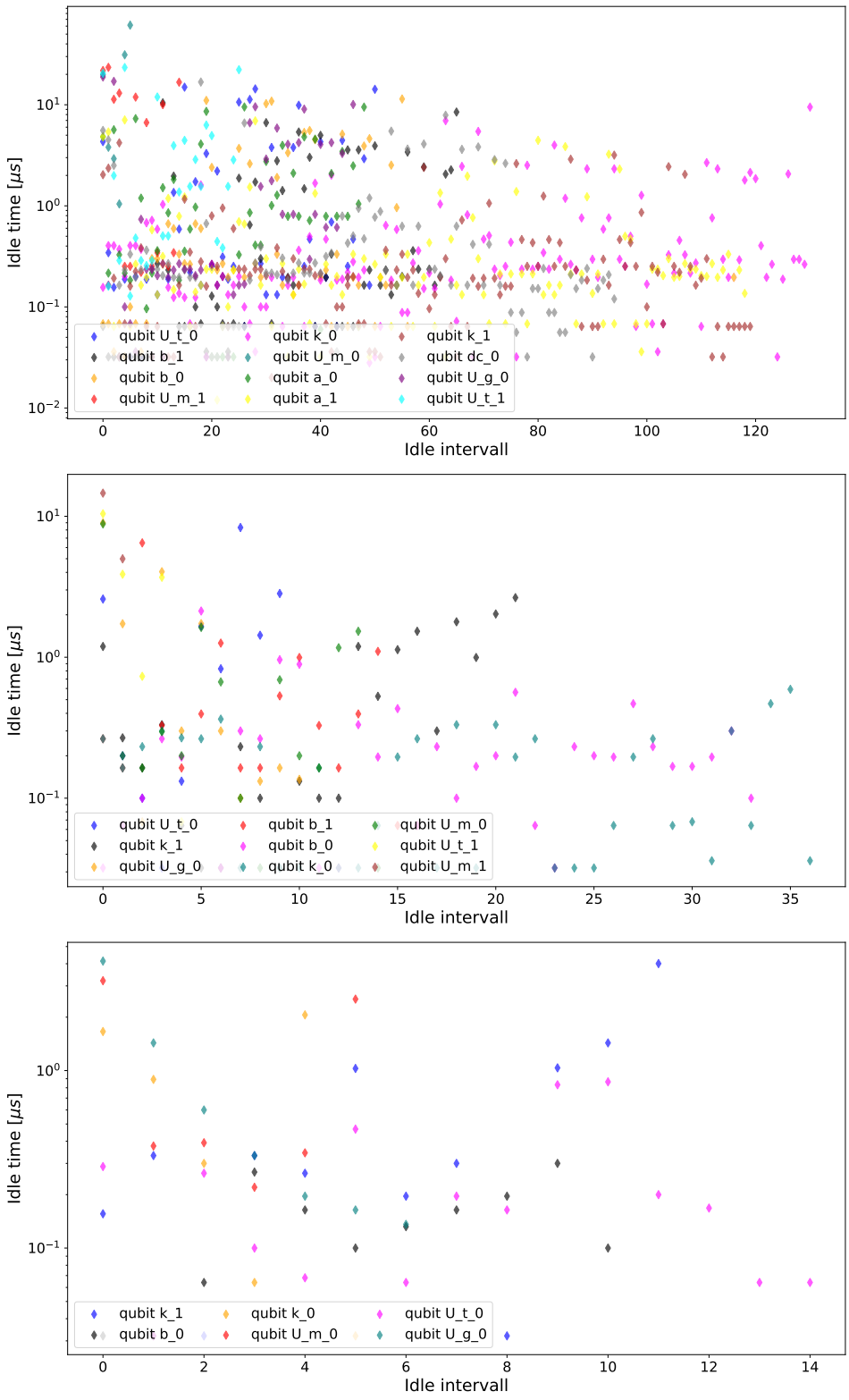


Figure 4.9: Two qubit circuits idle periods: Times intervals where each qubit is idle while waiting for operations on other qubits to be completed. From top to bottom: Two iterations full 2-qubit circuit, two iterations simplified 2-qubit and a sequential 2-qubit circuit. $|k_i\rangle$ represent the grid, $|a_i\rangle$, $|dc\rangle$, $|U_g\rangle$, the ancilla qubits, $|b_i\rangle$ the applied strain, $|U_t, i\rangle$ the material contrast and $|U_m, i\rangle$ the Fourier-space update.

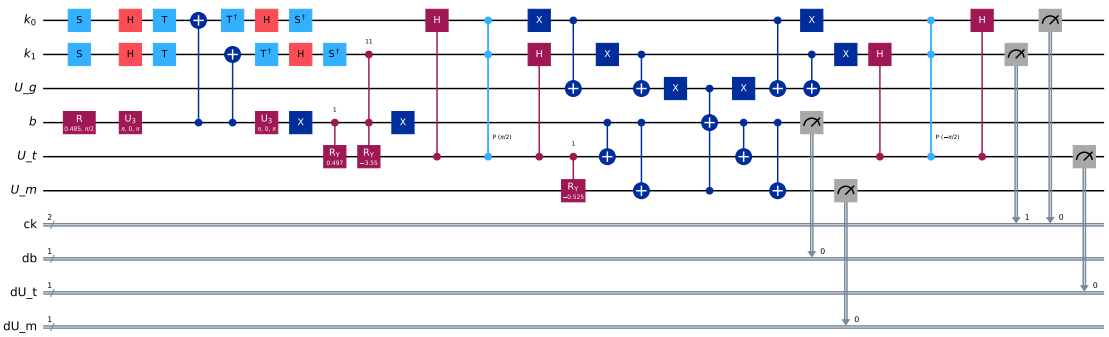


Figure 4.10: Sequential two qubit circuit RVE solver: Sequential version of the quantum solver circuit presented in Figure 4.3, as described in section 4.2. $|k_i\rangle$ represent the grid, $|b\rangle$ the applied strain, $|U_g\rangle$, the ancilla qubit, $|U_t\rangle$ the material contrast and $|U_m\rangle$ the Fourier-space update.

confirming that the sequential circuit preserves the true solution's overall shape more clearly than the other models.

4.3 Complexity

For a hardware-independent complexity analysis and comparison with the literature, all circuits were transpiled to the basis $\{\text{CNOT}, U_3\}$. $U_3 \sim R_z(\varphi)R_y(\theta)R_z(\lambda)$ gives all possible one-qubit rotations up to a global phase [10, 11]. It is expressed as

$$\mathrm{U}_3(\theta,\varphi,\lambda) = \begin{pmatrix} cos\frac{\theta}{2} & -e^{i\lambda}sin\frac{\theta}{2} \\ e^{i\varphi}sin\frac{\theta}{2} & e^{i(\varphi+\lambda)}cos\frac{\theta}{2} \end{pmatrix}. \tag{4.2}$$

A general multi-controlled operation with n control qubits, with $(N = 2^n)$, has complexity $\mathcal{O}(\log N)$, whereas the polynomial encoding has complexity $\mathcal{O}(\text{poly}(\log N))$. Furthermore, the QFT requires n Hadamard gates, $n(n+1)/2$ controlled Phase gates and $n/2$ swap gates. The complexity of the QFT circuit is therefore $\mathcal{O}((\log N)^2)$ [10, 11].

The complexity of the entire solver therefore scales as

$$\mathcal{O}(\text{poly}(\log N)). \tag{4.3}$$

It is consistent with Figure 4.13, which shows the scaling of the number of U_3 and CNOT gates for the three models as a function of the discretization.

For the *sequential* algorithm, additional classical post-processing is required between QPU executions. With C distinct measured bitstrings, and M bitstrings corresponding to the solution branch, this step first obtains the result probabilities through a division and sum over the measured outcomes, with cost $\mathcal{O}(C)$. It then filters the solution branch, also with cost $\mathcal{O}(C)$. The conversion of the M retained binary

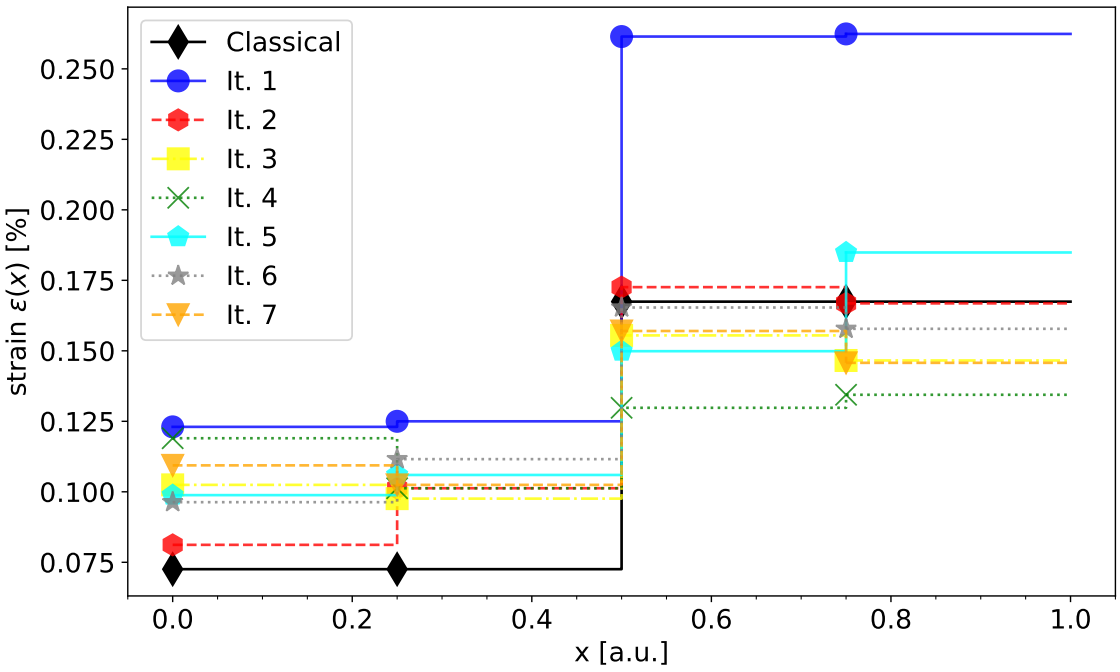


Figure 4.11: Two qubits sequential circuit solution: The solved local strain field $\varepsilon(x)$ of the one-dimensional RVE with a domain $\Omega = (0, 1)$. The grid was discretized to $N = 4$ points (2 qubits). The solutions were obtained with the classical FFT algorithm and seven consecutive runs of the sequential circuit on a QPU (`ibm_aachen`) with DD(XpXm) and *asap* scheduling applied.

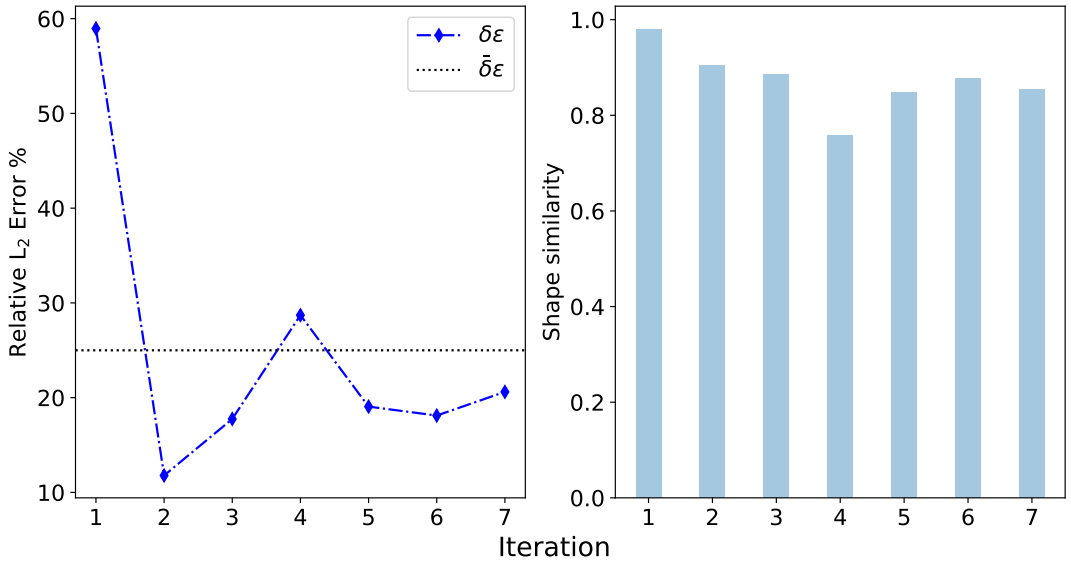


Figure 4.12: Two qubit sequential circuit relative errors: Convergence of the relative L_2-norm error and shape similarity with the true solution from runs on the QPU (`ibm_aachen`) with DD(XpXm) and *asap* scheduling applied.

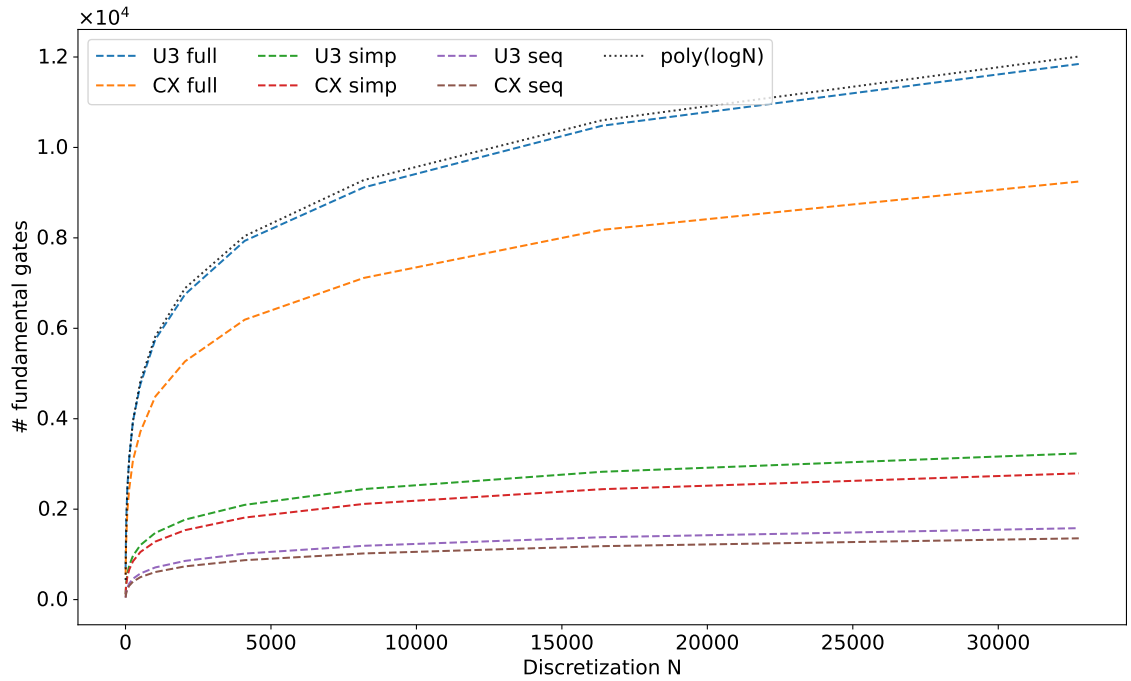


Figure 4.13: Complexity plot: Scaling of the total number of the general rotation $U_3(\theta, \varphi, \lambda)$ and $CNOT$ gates in the quantum circuits.

bitstrings to decimal grid coordinates requires $\mathcal{O}(M \log N)$, while the final sorting operation requires $\mathcal{O}(M \log M)$. The post-processing complexity is therefore

$$\mathcal{PP} = \mathcal{O}(C + M \log N + M \log M). \tag{4.4}$$

Since the full field over the N grid points must be reconstructed before it can be passed to the quantum circuit for the next iteration, M scales as $\mathcal{O}(N)$. With $n-$qubit for the grid and a constant number of ancilla qubits for the functions in the algorithm, the measured bitstrings C scales like $2^{n+\mathcal{O}(1)} = \mathcal{O}(N)$. The post-processing's complexity therefore becomes

$$\mathcal{PP} = \mathcal{O}(N + N \log N) = \mathcal{O}(N \log N), \tag{4.5}$$

and the total complexity of the sequential model

$$\mathcal{T} = \mathcal{O}(\text{poly}(\log N)) + \mathcal{O}(N \log N) = \mathcal{O}(N \log N). \tag{4.6}$$

This is comparable to the classical FFT-based RVE solver, it can therefore be concluded that the *sequential* algorithm loses its advantage over the FFT-based solver once the classical reconstruction between QPU runs is included.

5

Conclusion

In this work, a quantum solver for a simple one-dimensional RVE homogenization problem was studied in order to assess if the current state of the art in quantum computing can accelerate concurrent multiscale simulations in solid mechanics. The core of the solver is the quantum Fourier transform, which provides an exponential speed-up when compared to its classical counterpart. Three implementation models of the original algorithm were investigated on both a noiseless quantum circuit simulator and real IBM quantum hardware.

The studied algorithm showed good results on the circuit simulator, with relative errors below 1%. This indicates that the implemented circuits produce the expected solution and converge rapidly under ideal conditions. The executions on real quantum hardware were more challenging. For the finer grid discretization, the output became almost fully noisy. The *simplified* circuit showed better results with an average relative error between 23.91% and 36.19%. The best hardware result was obtained with the *sequential* circuit, reaching a relative error of 11.00% with two iterations and an average relative error of 25.00% when run up to seven iterations. The hardware results indicate that the computation is strongly limited by noise accumulation during execution. The structure of the algorithm leads to deep circuits and, as the circuit depth increases, gate errors become more significant. At the same time, the improvement obtained with dynamical decoupling and twirling suggests that small repeated gate imprecisions and idle-time decoherence and dephasing, also contribute substantially to the loss of encoded information.

The fully quantum versions of the proposed algorithms have a complexity of $\mathcal{O}(\text{poly}(\log N))$, which is consistent with the scaling reported by Liu et al. [11]. However, for the *sequential* model, the intermediate reconstruction step changes the overall scaling and makes it at least comparable to the classical FFT-based solver.

For future investigations, the principal challenge is the growing number of multi-controlled operations required as the number of iterations increases. Reducing this overhead would be necessary to avoid the need for sequential execution. Further hardware development should also help achieve this. In particular, processors with higher qubit connectivity (see subsection A.2.5), could offer better opportunities for approaching the desired $\mathcal{O}(\text{poly}(\log N))$ scaling.

Bibliography

[1] D. Hull and T. W. Clyne, *An Introduction to Composite Materials*, 2nd ed. Cambridge: Cambridge University Press, 1996.

[2] M. K. Hagnell, S. Kumaraswamy, T. Nyman, and M. Åkermo, "From aviation to automotive-a study on material selection and its implication on cost and weight efficient structural composite and sandwich designs," *Heliyon*, vol. 6, no. 3, p. e03716, 2020. [Online]. Available: http://dx.doi.org/10.1016/j.heliyon.2020.e03716

[3] B. D. Agarwal, L. J. Broutman, and K. Chandrashekhara, *Analysis and Performance of Fiber Composites*, 4th ed. Hoboken, NJ: Wiley, 2017.

[4] M. Zaidani, M. A. Omar, and S. Kumar, "Coupling of injection molding process to mechanical properties of short fiber composites: A through process modeling approach," *Journal of Reinforced Plastics and Composites*, vol. 34, no. 23, pp. 1963–1978, 2015. [Online]. Available: https://doi.org/10.1177/0731684415609138

[5] S. M. Mirkhalaf, E. H. Eggels, T. J. H. van Beurden, F. Larsson, and M. Fagerström, "A finite element based orientation averaging method for predicting elastic properties of short fiber reinforced composites," *Composites Part B: Engineering*, vol. 202, p. 108388, 2020. [Online]. Available: http://dx.doi.org/10.1016/j.compositesb.2020.108388

[6] M. G. D. Geers, V. G. Kouznetsova, and W. A. M. Brekelmans, "Multi-scale computational homogenization: Trends and challenges," *Journal of Computational and Applied Mathematics*, vol. 234, no. 7, pp. 2175–2182, 2010. [Online]. Available: http://dx.doi.org/10.1016/j.cam.2009.08.077

[7] B. Castricum, M. Fagerström, M. Ekh, F. Larsson, and S. Mirkhalaf, "A computationally efficient coupled multi-scale model for short fiber reinforced composites," *Composites Part A: Applied Science and Manufacturing*, vol. 163, p. 107233, 2022. [Online]. Available: https://doi.org/10.1016/j.compositesa.2022.107233

[8] H. Moulinec and P. Suquet, "A fast numerical method for computing the linear and nonlinear mechanical properties of composites," *Comptes Rendus de l'Académie des Sciences. Série II*, vol. 318, no. 11, pp. 1417–1423, 1994.

[9] H. Moulinec and P. Suquet, "A numerical method for computing the overall response of nonlinear composites with complex microstructure," *Computer Methods in Applied Mechanics and Engineering*, vol. 157, no. 1-2, pp. 69–94, 1998. [Online]. Available: http://dx.doi.org/10.1016/S0045-7825(97)00218-1

[10] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, 10th ed. Cambridge: Cambridge University Press, 2010.

[11] B. Liu, M. Ortiz, and F. Cirak, "Towards quantum computational mechanics," *Computer Methods in Applied Mechanics and Engineering*, vol. 432, p. 117403, 2024. [Online]. Available: http://dx.doi.org/10.1016/j.cma.2024.117403

[12] S. Torquato, *Random Heterogeneous Materials: Microstructure and Macroscopic Properties*. New York: Springer, 2002.

[13] T. Kanit, S. Forest, I. Galliet, V. Mounoury, and D. Jeulin, "Determination of the size of the representative volume element for random composites: Statistical and numerical approach," *International Journal of Solids and Structures*, vol. 40, no. 13-14, pp. 3647–3679, 2003. [Online]. Available: http://dx.doi.org/10.1016/S0020-7683(03)00143-4

[14] J. C. Michel, H. Moulinec, and P. Suquet, "A computational scheme for linear and non-linear composites with arbitrary phase contrast," *International Journal for Numerical Methods in Engineering*, vol. 52, no. 1-2, pp. 139–160, 2001. [Online]. Available: http://dx.doi.org/10.1002/nme.275

[15] M. Schneider, "FFT-based homogenization for microstructures discretized by linear hexahedral elements," *International Journal for Numerical Methods in Engineering*, vol. 109, no. 10, pp. 1461–1489, 2017. [Online]. Available: http://dx.doi.org/10.1002/nme.5336

[16] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. Cambridge, MA: MIT Press, 2009.

[17] P. Duhamel and M. Vetterli, "Fast fourier transforms: A tutorial review and a state of the art," *Signal Processing*, vol. 19, no. 4, pp. 259–299, 1990. [Online]. Available: http://dx.doi.org/10.1016/0165-1684(90)90158-U

[18] J. Clarke and F. K. Wilhelm, "Superconducting quantum bits," *Nature*, vol. 453, no. 7198, pp. 1031–1042, 2008. [Online]. Available: http://dx.doi.org/10.1038/nature07128

[19] P. Krantz, M. Kjaergaard, F. Yan, T. P. Orlando, S. Gustavsson, and W. D. Oliver, "A quantum engineer's guide to superconducting qubits," *Applied Physics Reviews*, vol. 6, no. 2, p. 021318, 2019. [Online]. Available: http://dx.doi.org/10.1063/1.5089550

[20] J. Preskill, "Quantum computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, 2018. [Online]. Available: http://dx.doi.org/10.22331/q-2018-08-06-79

[21] J. Helsen, I. Roth, E. Onorati, A. Werner, and J. Eisert, "General framework for randomized benchmarking," *PRX Quantum*, vol. 3, no. 2, Jun. 2022. [Online]. Available: http://dx.doi.org/10.1103/PRXQuantum.3.020357

[22] N. Stamatopoulos, D. J. Egger, Y. Sun, C. Zoufal, R. Iten, N. Shen, and S. Woerner, "Option pricing using quantum computers," *Quantum*, vol. 4, p. 291, 2020. [Online]. Available: http://dx.doi.org/10.22331/q-2020-07-06-291

[23] Qiskit Aer Development Team. (2026) AerSimulator documentation. [Online]. Available: https://qiskit.github.io/qiskit-aer/stubs/qiskit_aer.AerSimulator.html

[24] G. Vidal, "Efficient classical simulation of slightly entangled quantum computations," *Physical Review Letters*, vol. 91, no. 14, p. 147902, 2003. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.91.147902

[25] Z. Cai, R. Babbush, S. C. Benjamin, S. Endo, W. J. Huggins, Y. Li, J. R. McClean, and T. E. O'Brien, "Quantum error mitigation," *Rev. Mod. Phys.*, vol. 95, p. 045005, Dec 2023. [Online]. Available: https://link.aps.org/doi/10.1103/RevModPhys.95.045005

[26] IBM Quantum. (2026) Readout error mitigation for the sampler primitive using M3. [Online]. Available: https://quantum.cloud.ibm.com/docs/en/tutorials/readout-error-mitigation-sampler

[27] Z. Cai and S. C. Benjamin, "Constructing smaller pauli twirling sets for arbitrary error channels," *Scientific Reports*, vol. 9, p. 11281, 2019. [Online]. Available: http://dx.doi.org/10.1038/s41598-019-46722-7

[28] N. Ezzell, B. Pokharel, L. Tewala, G. Quiroz, and D. A. Lidar, "Dynamical decoupling for superconducting qubits: A performance survey," *Physical Review Applied*, vol. 20, no. 6, p. 064027, 2023. [Online]. Available: http://dx.doi.org/10.1103/PhysRevApplied.20.064027

[29] IBM Quantum. (2026) Dynamical decoupling options. [Online]. Available: https://quantum.cloud.ibm.com/docs/api/qiskit-ibm-runtime/options-dynamical-decoupling-options

[30] IBM Quantum. (2026) Rzgate. [Online]. Available: https://quantum.cloud.ibm.com/docs/en/api/qiskit/qiskit.circuit.library.RZGate

[31] D. C. McKay, C. J. Wood, S. Sheldon, J. M. Chow, and J. M. Gambetta, "Efficient z gates for quantum computing," *Physical Review A*, vol. 96, no. 2, 2017. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.96.022330

[32] J. Koch, T. M. Yu, J. Gambetta, A. A. Houck, D. I. Schuster, J. Majer, A. Blais, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf, "Charge-insensitive qubit design derived from the cooper pair box," *Physical Review A*, vol. 76, no. 4, p. 042319, 2007. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.76.042319

[33] M. Tinkham, *Introduction to Superconductivity*, 2nd ed. Mineola, NY, USA: Dover Publications, 2004.

[34] K. K. Likharev, *Dynamics of Josephson Junctions and Circuits*. New York, NY, USA: Gordon and Breach, 1986.

[35] B. D. Josephson, "The discovery of tunnelling supercurrents," *Reviews of Modern Physics*, vol. 46, no. 2, pp. 251–254, 1974. [Online]. Available: http://dx.doi.org/10.1103/RevModPhys.46.251

[36] V. V. Schmidt, *The Physics of Superconductors: Introduction to Fundamentals and Applications*. Berlin, Germany: Springer, 1997.

[37] C. Kittel, *Introduction to Solid State Physics*, 8th ed. Hoboken, NJ, USA: John Wiley & Sons, 2005.

[38] A. Blais, A. L. Grimsmo, S. M. Girvin, and A. Wallraff, "Circuit quantum electrodynamics," *Reviews of Modern Physics*, vol. 93, no. 2, p. 025005, 2021. [Online]. Available: http://dx.doi.org/10.1103/RevModPhys.93.025005

[39] J. Stehlik, D. M. Zajac, D. L. Underwood, T. Phung, J. Blair, S. Carnevale, D. Klaus, G. A. Keefe, A. Carniol, M. Kumph, M. Steffen, and O. E. Dial, "Tunable coupling architecture for fixed-frequency transmon superconducting qubits," *Physical Review Letters*, vol. 127, no. 8, p. 080505, 2021. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.127.080505

[40] IBM Quantum, "Processor types," Online, 2026, available: https://quantum.cloud.ibm.com/docs/guides/processor-types. Accessed: 2026-05-25.

[41] G. Ithier, E. Collin, P. Joyez, P. J. Meeson, D. Vion, D. Esteve, F. Chiarello, A. Shnirman, Y. Makhlin, J. Schriefl, and G. Schön, "Decoherence in a superconducting quantum bit circuit," *Physical Review B*, vol. 72, no. 13, p. 134519, 2005. [Online]. Available: http://dx.doi.org/10.1103/PhysRevB.72.

134519

[42] G. Bimana, P. Borg Karlsson, S. V. Strassburger, and W. Worawirat, "Controlling a quantum processor," 2024, unpublished lab report, Experimental methods in modern physics (TIF295), department of Physics.

[43] E. Magesan, J. M. Gambetta, and J. Emerson, "Scalable and robust randomized benchmarking of quantum processes," *Physical Review Letters*, vol. 106, no. 18, p. 180504, 2011. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.106.180504

[44] IBM, "Ibm delivers new quantum processors, software, and algorithm breakthroughs on path to advantage and fault tolerance," Online, 2025, available: https://newsroom.ibm.com/2025-11-12-ibm-delivers-new-quantum-processors,-software, -and-algorithm-breakthroughs-on-path-to-advantage-and-fault-tolerance. Accessed: 2026-05-25.

[45] IBM Quantum, "Quantum hardware," Online, 2026, available: https://www.ibm.com/quantum/hardware. Accessed: 2026-05-25.

A

Supplementary material

A.1 One-Dimensional FFT Formulation

For a one-dimensional problem such as an elastic rod on $\Omega = (0, 1)$, the homogenization formulation introduced in subsection 2.1.1 reduces to the a scalar problem.

In this case the displacement reduces to a scalar field $u = u(x)$ and the strain becomes

$$\varepsilon(x) = \frac{du}{dx}. \tag{A.1}$$

The constitutive law $\boldsymbol{\sigma} = \mathcal{C} : \boldsymbol{\varepsilon}$ then reduces to

$$\sigma(x) = \mu(x)\varepsilon(x), \tag{A.2}$$

where $\mu(x)$ is the spatially varying modulus. Equilibrium gives

$$\frac{d}{dx}\Big(\mu(x)\varepsilon(x)\Big) = 0. \tag{A.3}$$

Splitting the material into homo- and heterogeneous parts according to $\mu(x) = \mu_0 + (\mu(x) - \mu_0)$, and then defining $\tau(x) = \Big(\mu(x) - \mu_0\Big)\varepsilon(x)$ as the polarization field similarly to Equation 2.10 gives

$$\frac{d}{dx}\Big(\mu_0\varepsilon(x) + \tau(x)\Big) = 0. \tag{A.4}$$

Integration yields

$$\varepsilon(x) = \frac{\xi}{\mu_0} - \frac{1}{\mu_0}\tau(x), \tag{A.5}$$

where ξ is a constant stress. With $\langle\varepsilon\rangle = E$, taking the volume average gives

$$\xi = \mu_0 E + \langle\tau\rangle. \tag{A.6}$$

Substituting back into the strain relation yields

$$\varepsilon(x) = E - \frac{1}{\mu_0}\left(\tau(x) - \langle\tau\rangle\right), \tag{A.7}$$

which, in Fourier space becomes

$$\widehat{\varepsilon}_k = \widehat{E}_k - \frac{1}{\mu_0}\left(\widehat{\tau}_k - \widehat{\langle\tau\rangle}_k\right). \tag{A.8}$$

E and $\langle\tau\rangle$ are spatial average, therefore constant, hence

$$\begin{aligned} \hat{E}_0 &= E \quad \& \quad \widehat{\langle\tau\rangle}_0 = \langle\tau\rangle, \qquad \text{for } k = 0, \\ \widehat{E}_k &= 0 \quad \& \quad \widehat{\langle\tau\rangle}_k = 0, \qquad \text{for } k \neq 0. \end{aligned} \tag{A.9}$$

Furthermore, for the zero Fourier mode,

$$\widehat{\tau}_0 = \frac{1}{L}\int_0^L \tau(x)\,dx = \langle\tau\rangle. \tag{A.10}$$

Combining Equation A.9 and Equation A.10 with Equation A.8 gives,

$$\begin{aligned} \widehat{\varepsilon}_0 &= E, \qquad\qquad \text{for } k \neq 0 \\ \widehat{\varepsilon}_k &= -\frac{1}{\mu_0}\widehat{\tau}_k, \qquad \text{for } k \neq 0, \end{aligned} \tag{A.11}$$

which is the one-dimensional counterpart to Equation 2.12 with $\hat{\Gamma}_0(k) = -\frac{1}{\mu_0}$. The tensorial formulation is therefore specialized to a scalar one-dimensional problem, while preserving the structure of the Moulinec-Suquet scheme.

A.2 Superconducting Quantum Computers

Superconducting quantum computers are among the most advanced approaches to building practical quantum machines. They use μm-scale electrical circuits fabricated from superconducting materials and operated at mK temperatures, where dissipation is strongly suppressed and macroscopic quantum coherence quantum can persist in the electrical circuit [18, 19].

To function as qubits, these circuits must have non-equally spaced energy levels, which is achieved by introducing a Josephson junction (JJ). It provides the non-linearity needed to isolate two energy levels and use them as the computational states $|0\rangle$ and $|1\rangle$ [19].

The most common superconducting qubit is the transmon qubit. A transmon consists mainly of a Josephson junction shunted by a large capacitor. The Josephson junction provides the non-linearity, while the capacitor reduces sensitivity to charge noise. Together, these elements produce an anharmonic oscillator whose two lowest energy levels can be used as the qubit states $|0\rangle$ and $|1\rangle$, and selectively driven by microwave pulses [32].

A.2.1 The Josephson Junction

A Josephson junction is formed by separating two superconductors with a thin non-superconducting weak link, e.g. an insulating barrier, a normal metal, or a constriction [33, 34]. Cooper pairs can tunnel through this barrier, producing a supercurrent across the junction [35].

In the absence of an applied voltage, the junction supports a DC Josephson current

$$I_S = I_c \sin \phi, \tag{A.12}$$

where I_c is the critical current and ϕ is the superconducting phase difference across the junction.

The voltage across the junction is related to the phase evolution by

$$V = \frac{\Phi_0}{2\pi} \frac{d\phi}{dt} = \frac{\hbar}{2e} \frac{d\phi}{dt}, \tag{A.13}$$

where $\Phi_0 = h/2e$ is the flux quantum and e is the elementary charge. For a constant applied voltage, the phase evolves periodically and the current oscillates at the Josephson frequency

$$f_J = \frac{2eV}{h}. \tag{A.14}$$

Real junctions can be modelled by adding a resistive contribution to the ideal Josephson current. In the resistively shunted junction model, the total current is

$$I = I_c \sin \phi + \frac{\hbar}{2eR} \frac{d\phi}{dt}, \tag{A.15}$$

where R is the effective normal-state resistance of the junction [36, 37].

The Josephson junction behaves as a non-linear inductor. Its potential energy is

$$U(\phi) = -E_J \cos \phi, \tag{A.16}$$

with

$$E_J = \frac{\Phi_0 I_c}{2\pi}, \tag{A.17}$$

where E_J is the Josephson energy. The cosine potential is non-quadratic and therefore makes the oscillator anharmonic. This is essential for qubit operation, since the lowest transition can be addressed without simultaneously driving all higher transitions as in a harmonic LC oscillator [19, 32].

A.2.2 The Transmon Circuit

A transmon consists of a Josephson junction in parallel with a relatively large shunt capacitor. Its simplified Hamiltonian is

$$H = 4E_C(n - n_g)^2 - E_J \cos \phi, \tag{A.18}$$

where E_J is the Josephson energy, n is the number of excess Cooper pairs on the superconducting island, and n_g is the dimensionless offset charge induced by the surrounding electrostatic environment [32, 38]. The charging energy is

$$E_C = \frac{e^2}{2C_\Sigma}, \tag{A.19}$$

where C_Σ is the total effective capacitance of the island, including the shunt capacitance and smaller parasitic or coupling capacitances [32].

The transmon operates in the regime

$$\frac{E_J}{E_C} \gg 1. \tag{A.20}$$

Increasing E_J/E_C makes the energy levels much less sensitive to fluctuations in n_g, thereby reducing charge noise and improving dephasing times. The cost is reduced anharmonicity, although enough anharmonicity remains to selectively control the two lowest states with microwave pulses [32]. In practice, these two lowest eigenstates are used as $|0\rangle$ and $|1\rangle$.

A.2.3 Control, Coupling, and Readout

Single-qubit gates are applied using microwave pulses close to the qubit transition frequency. The pulse amplitude, duration, and phase determine the rotation applied to the qubit state.

Two-qubit gates require controllable interactions between neighboring qubits. In architectures such as IBM's `Heron r3` processor used in this work, tunable couplers are used to control these interactions. The coupling can be reduced while qubits are idle and increased during two-qubit gate operations, thereby suppressing unwanted residual interactions and reducing crosstalk-related errors [39, 40].

Readout is usually performed through a microwave resonator coupled to the qubit (see cover picture). The resonator frequency shifts depending on the qubit state. A microwave readout pulse is sent to the resonator, and the reflected or transmitted signal is amplified and classified as a 0 or 1 result [19].

A.2.4 Characterisation: T_1, T_2, and Fidelity

For qubit characterisation, spectroscopy is used to find the transition frequency, while Rabi measurements calibrate the pulse amplitudes and durations required for rotations such as X, Y, and $\pi/2$ gates.

The relaxation time T_1 measures the timescale over which the excited state $|1\rangle$ decays toward the ground state $|0\rangle$. The dephasing time T_2 measures how long phase information survives in a superposition. Ramsey and echo experiments are

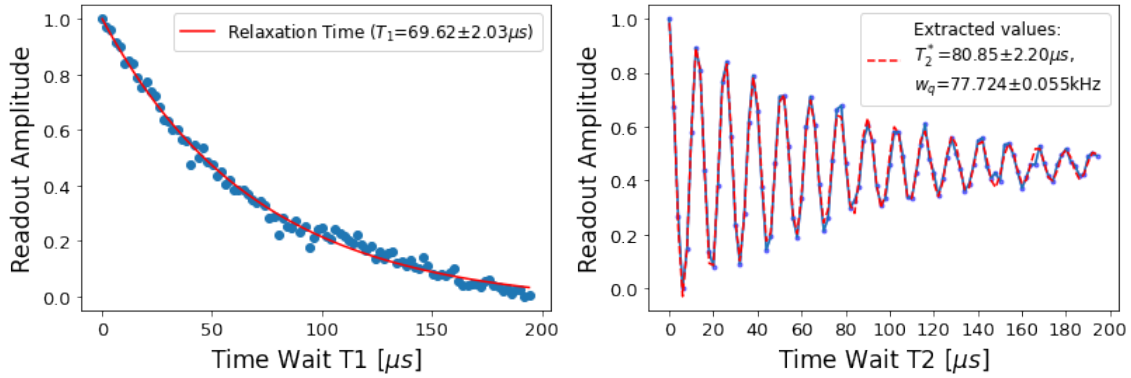


Figure A.1: Extraction of relaxation time (left) and pure dephasing time (right) for qubit one on *Pingu* at MC2, Chalmers [42].

commonly used to characterize dephasing and separate slow noise contributions from other decoherence mechanisms [41].

$$\frac{1}{T_2} = \frac{1}{2T_1} + \frac{1}{T_2^*}, \tag{A.21}$$

where T_2^* is the pure dephasing time as can be seen in Figure A.1.

Gate fidelity measures how closely an implemented hardware operation matches the ideal operation. It is commonly estimated using benchmarking methods such as randomized benchmarking [43]. Readout fidelity measures how often the measurement correctly reports the prepared state.

A.2.5 IBM Processors

The IBM `Heron` family, used in this work, combines fixed-frequency qubits with tunable couplers. Heron r3 uses 156 qubits arranged in a heavy-hexagonal lattice [40].

The newer IBM `Nighthawk` architecture follows the same superconducting-qubit approach but changes the connectivity. It uses 120 qubits connected by 218 next-generation tunable couplers in a square lattice with four-neighbor connectivity [40, 44, 45]. Better connectivity reduces routing overhead during transpilation.

A.3 Circuits

A.3.1 Polynomial encoding

The simplification of the polynomial encoding was done by taking advantage of the geometry of the RVE and the binary representation of the physical grid. The modulus $\mu(x)$ has a discontinuity at $x = 0.5$, which can be expressed as $\frac{k_0}{2^2} + \frac{k_1}{2^1}$ for $k_0 = 0$. Letting k_i represent states of the qubits in the computation, it follows

that a gate rotation controlled by qubit k_1 would change the encoded value μ_k for the corresponding states ($|01\rangle = |0.5\rangle$ and $|11\rangle = |0.75\rangle$ in Qiskit's little-endian notation).

Additionally, applying the R_y gate twice on a qubit gives

$$|\psi'\rangle = R_y(p_2(x))\left(R_y(p_1(x))|\psi\rangle\right), \tag{A.22}$$

which, expressed in its exponential (Equation 2.23) becomes

$$\begin{aligned} |\psi'\rangle &= e^{-\frac{i}{2}p_2(x)Y}e^{-\frac{i}{2}p_1(x)Y}|\psi\rangle \\ &= e^{-\frac{i}{2}(p_2(x)+p_1(x))Y}|\psi\rangle \\ &= R_y(p_2(x)+p_1(x))|\psi\rangle. \end{aligned} \tag{A.23}$$

The modulus is therefore encoded by applying one R_y rotation first to encode $|0\rangle$ and $|0.25\rangle$, then using a second a controlled R_y the remaining value is added on the controlled branch (see Figure A.2).

A.3.2 SWAP

To swap the states $|\psi'\rangle = -\frac{\hat{\tau}_0}{\mu_0}|0\rangle^n|0\rangle^S|11\rangle|0\rangle^{2S-2}$ and $|\psi''\rangle = \sqrt{N}E|0\rangle^n|0..10..\rangle|0\rangle^{2S}$, the workflow is as follow:

Flip the physical register with X gates to $|1\rangle^n$ and use CNOTs and X to make a parity check and see if $k = 0$. Store the result in an extra ancillary qubit $|U_g\rangle$. The parity check works because of the state of the system after the QFT and avoids using extra Toffoli gates.

Apply CNOTs to all the polynomial qubits up to current iteration, controlled by the relevant auxiliary mean strain branch (current iteration).

$$\begin{aligned} |\psi'\rangle &= -\frac{\hat{\tau}_0}{\mu_0}|1\rangle^n|0\rangle^S|11\rangle|0\rangle^{2S-2} \\ |\psi''\rangle &= \sqrt{N}E|1\rangle^n|0..10..\rangle|11\rangle|0\rangle^{2S-2} \end{aligned}$$

Use a Toffoli gate, controlled by the ancillary parity qubit and the current iteration polynomial qubit, to flip the current iteration auxiliary mean strain qubit,

$$\begin{aligned} |\psi'\rangle &= -\frac{\hat{\tau}_0}{\mu_0}|1\rangle^n|0..10..\rangle|11\rangle|0\rangle^{2S-2} \\ |\psi''\rangle &= \sqrt{N}E|1\rangle^n|0\rangle^S|11\rangle|0\rangle^{2S-2} \end{aligned}$$

Apply a CNOT back to all the polynomial qubits up to current iteration, controlled

by the relevant auxiliary mean strain branch (current iteration).

$$|\psi'\rangle = -\frac{\hat{\tau}_0}{\mu_0}|1\rangle^n|0..10..\rangle|0\rangle^{2S}$$
$$|\psi'\rangle = \sqrt{N}E|1\rangle^n|0\rangle^S|11\rangle|0\rangle^{2S-2}$$

Uncompute the parity qubit and flip back the physical grid register.

$$|\psi'\rangle = -\frac{\hat{\tau}_0}{\mu_0}|0\rangle^n|0..10..\rangle|0\rangle^{2S}$$
$$|\psi'\rangle = \sqrt{N}E|0\rangle^n|0\rangle^S|11\rangle|0\rangle^{2S-2}$$

The resulting states correspond to the original states swapped: $|\psi'\rangle \leftrightarrow |\psi''\rangle$.

A.3.3 2-qubit circuit

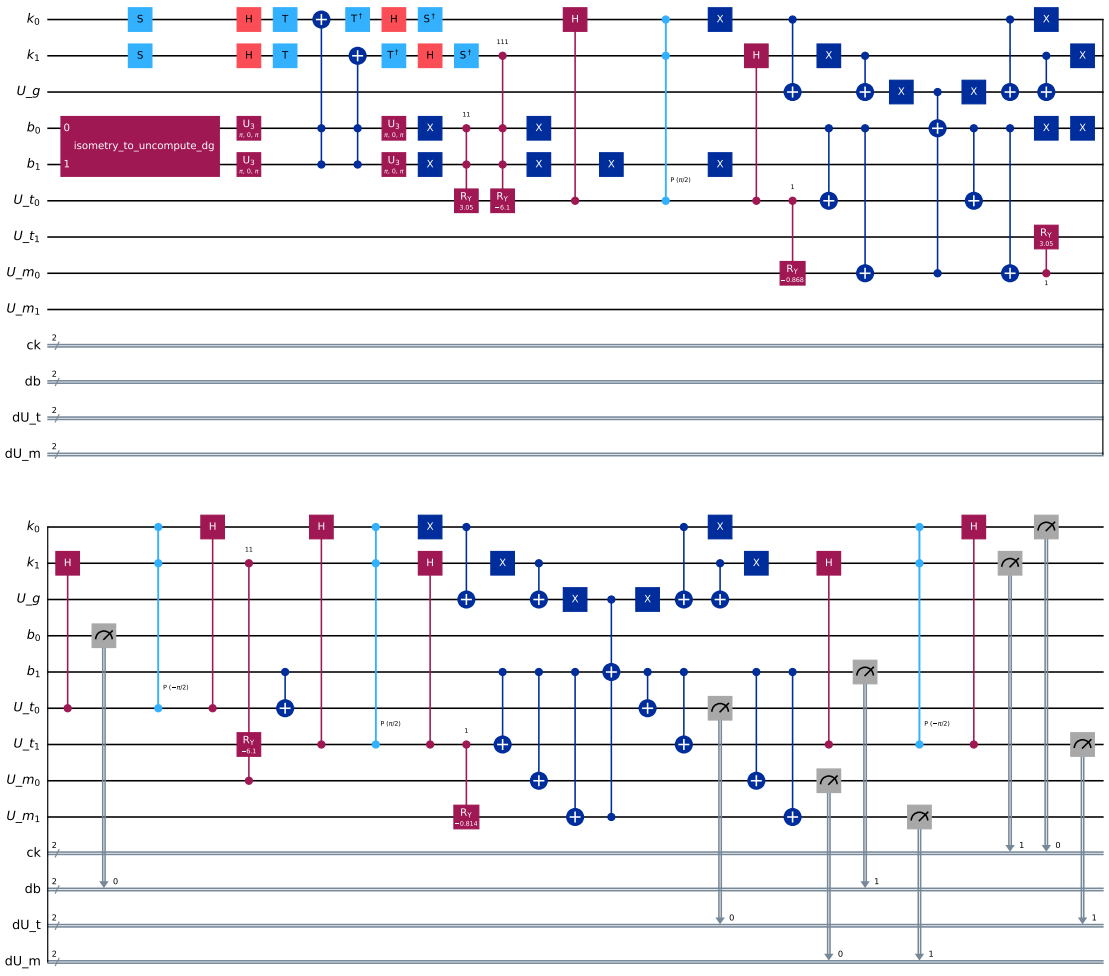


Figure A.2: Simplified two qubit circuit RVE solver: Two iterations simplified version of the quantum solver circuit presented in Figure 4.3, as described in section 4.2. $|k_i\rangle$ represent the grid, $|a_i\rangle$, $|d_c\rangle$, $|U_g\rangle$, the ancilla qubits, $|b\rangle$ the applied strain, $|U_{t,i}\rangle$ the material contrast and $|U_{m,i}\rangle$ the Fourier space update.

A.3.4 Transpiled and mapped circuit (sequential)

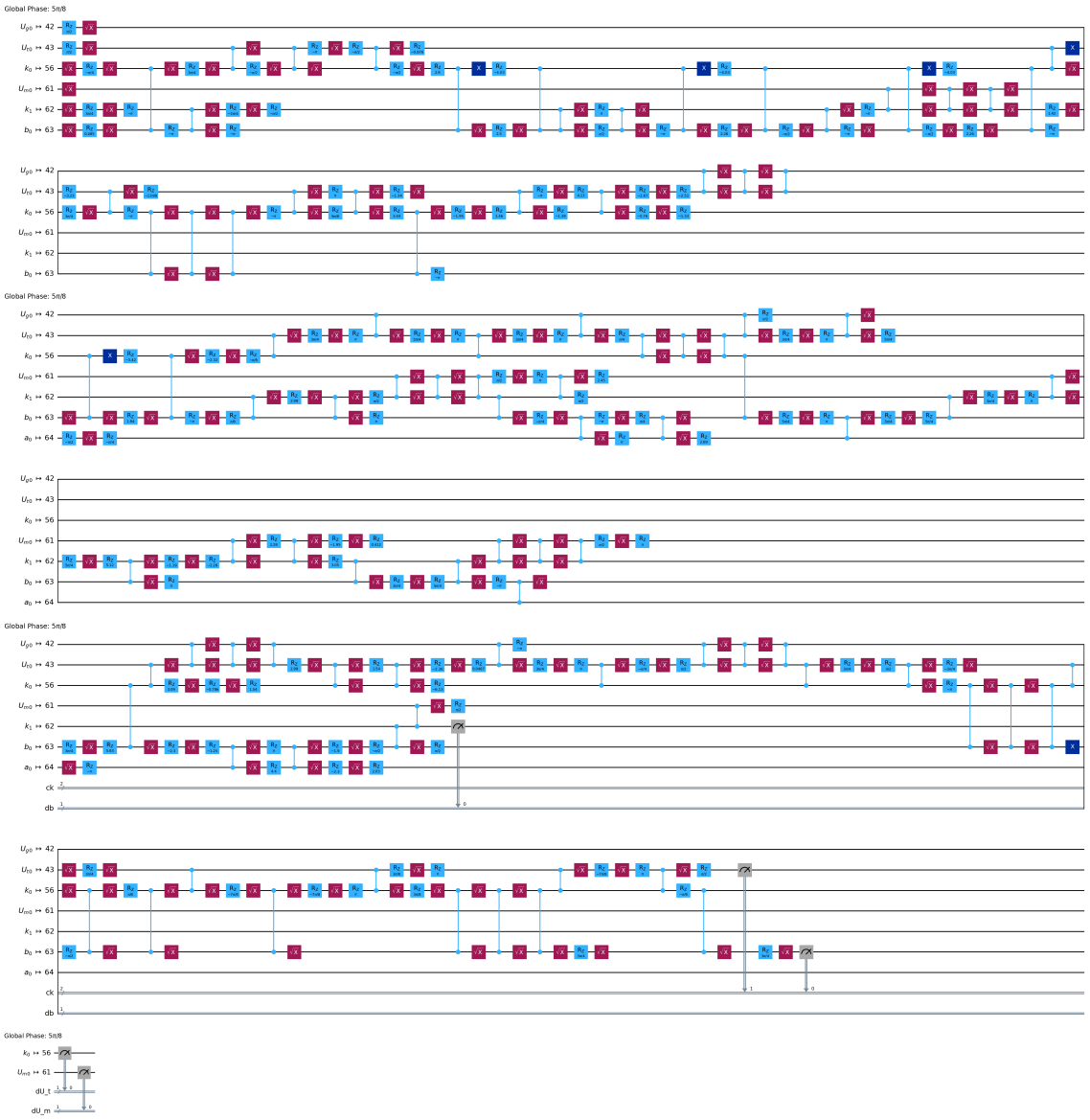


Figure A.3: Transpiled circuit: Circuit for the sequential model transpiled to the basis gates of `ibm_aachen` and mapped to its assigned physical qubits. Read from Left to right and top to bottom.

DEPARTMENT OF PHYSICS
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY