Computers and Fluids 295 (2025) 106619

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid

Application of the quantum Fourier transform in a harmonic balance solver for Burgers' equation

Loïc Dewitte [a,b,1], Jérémie Roland [c], Frank Eulitz [a,*]

[a] Turbomachinery and Propulsion Department, von Karman Institute for Fluid Dynamics, 1640 Sint-Genesius-Rode, Belgium
[b] Whittle Laboratory, Department of Engineering, University of Cambridge, Cambridge CB3 0DY, United Kingdom
[c] Centre for Quantum Information & Communication, École polytechnique de Bruxelles, Université libre de Bruxelles, 1050 Brussels, Belgium

ARTICLE INFO

Dataset link: http://dx.doi.org/10.17632/m4wxpzhdz5.1

Keywords:
Quantum computing
Quantum algorithms
Quantum Fourier transform
Computational fluid dynamics
Fourier methods
Harmonic balance method

ABSTRACT

This paper explores the potential of quantum computing for computational fluid dynamics (CFD) applications, with a focus on turbomachinery CFD. A harmonic balance solver is developed for Burgers' equation, in which discrete Fourier transforms are approximated using a hybrid quantum-classical algorithm based on the quantum Fourier transform (QFT). Three novel algorithms are presented to estimate Fourier coefficients, providing complete knowledge of their amplitudes and phases, which is not possible with the standard QFT. Their behaviour is studied theoretically and numerically, under noiseless and noisy conditions. The algorithms, whose performance is limited by the extensive sampling required to achieve a small error on the Fourier coefficients, tolerate low levels of depolarising noise. The behaviour of the hybrid solver is investigated for a baseline case with a Reynolds number of 1000, 7 harmonics and 100 grid cells, using the best performing algorithm in a noiseless setting with up to 10^8 samples per QFT. Residuals decrease until errors introduced by statistical uncertainty dominate the total error on the solution. Nevertheless, the hybrid solutions match the classical one closely, with RMS residuals as low as 10^{-4}. The impact of several solver parameters on convergence and solution quality is also assessed, including the effect of noise. Although the latter rapidly degrades the solution, the solver achieves satisfactory approximations to the classical solution with 0.001% of depolarising noise. Without seeking to demonstrate a quantum advantage, this work offers valuable insights into the opportunities and challenges of quantum computing, helping readers understand how to design, implement and study quantum algorithms, as well as evaluate their impact in CFD applications.

1. Introduction

Computational fluid dynamics (CFD) continually strives to enhance simulation and optimisation capabilities, requiring ever-increasing computational and storage resources. This is especially true for turbomachinery CFD, where scale-resolved simulations of multi-passage flows are increasingly used in design. As classical computing hardware development nears physical limits, the emerging field of quantum computing (QC) might prove a valuable addition to the engineer's toolbox, with the potential to address the challenges of classical computing and push CFD towards new frontiers through increased computing power and storage capacity. This paper ventures to the intersection of these disciplines and explore the opportunities and challenges that quantum algorithms present for future CFD applications.

Quantum computing leverages properties of quantum mechanics such as superposition and entanglement to perform calculations using quantum bits, or qubits. Superposition allows qubits to exist in multiple states simultaneously, while entanglement links qubits together, making them interdependent even if physically separated. These properties enable operations to be carried out on all data simultaneously, providing access to an exponentially larger computational and storage space than classical computing.

However, quantum computing has several drawbacks, due primarily to the principles of quantum mechanics. Quantum computers are inherently probabilistic because the action of measuring a quantum system collapses its state, which means that information can only be recovered by estimating the probability distribution of the system state. This involves sampling states that are reproduced many times, which can be resource-intensive as a considerable number of samples may be needed for accurate approximations. Additionally, the linear nature of quantum mechanics complicates the introduction of non-linearities. For instance, the no-cloning theorem prohibits the identical

* Corresponding author.
E-mail addresses: loic.dewitte@eng.ox.ac.uk (L. Dewitte), frank.eulitz@vki.ac.be (F. Eulitz).
[1] Present address: Oxford Thermofluids Institute, Department of Engineering Science, University of Oxford, Oxford OX2 0ES, United Kingdom.

https://doi.org/10.1016/j.compfluid.2025.106619
Received 18 July 2024; Received in revised form 14 March 2025; Accepted 23 March 2025
Available online 9 April 2025

and independent copying of information. Moreover, quantum state evolution is governed by unitary and reversible operations, which makes non-unitary processes such as dissipation difficult to model. Hardware implementation poses another major challenge. Current devices are built with too few qubits for most practical applications. For example, one of the largest quantum computer today, IBM Condor, has only 1121 physical qubits [1]. These are highly sensitive to their environment, and factors like internal noise and undesired interactions with the surroundings can introduce significant errors, as can operational issues. As a result, present-day devices are often referred to as noisy intermediate-scale quantum (NISQ) computers [2]. In spite of these difficulties, there are promising developments and some argue that NISQ devices could already start outperforming classical computers [3]. However, to build reliable and scalable quantum computers for the long run, hardware limitations must be addressed through advances in qubit technologies, fault-tolerant designs and error correction methods.

Despite these challenges, the field has garnered widespread interest as an advantage in computational complexity was demonstrated theoretically for several algorithms. Quantum advantage is assessed by comparing the complexities of a quantum algorithm with its best-known classical counterpart, often asymptotically and in terms of number of operations or logic gates. However, because the sampling process can negate the advantage, it is often restricted to specific tasks that only require information about a few output states. Furthermore, data input and output processes can also cancel out the advantage, with classical data pre- and post-processing, quantum state preparation and measurement steps often becoming bottlenecks. Hybrid quantum-classical algorithms, in which expensive parts of a classical algorithm are replaced by quantum subroutines, suffer particularly from these problems. Quantum algorithms are nevertheless worth studying, often using simulators, as future scalable and fault-tolerant quantum computers with efficient data handling might eventually translate theoretical advantages into tangible speed-ups.

For this reason, there is growing interest in using quantum algorithms to solve partial differential equations in fluid dynamics. Comprehensive reviews of the current research trends and detailed introductions were given at the 2022 and 2024 'Introduction to Quantum Computing in Fluid Dynamics' VKI Lecture Series [4]. Recent studies range from using quantum algorithms to solve linear or non-linear systems of equations within general methods, such as the finite element [5,6] or the finite volume methods [7], to developing more specialised approaches like the quantum lattice Boltzmann method [8]. Creative ideas are being investigated as well, including differentiable quantum circuits [9], hybrid reservoir computing [10], and Carleman linearisation [11]. The special issue in which this paper is published also highlights interesting areas of research.

Many quantum algorithms use the quantum Fourier transform (QFT), the quantum equivalent of the discrete Fourier transform (DFT), which enjoys an exponential advantage over the fast Fourier transform (FFT) algorithm. This advantage is however offset by state preparation and sampling. The QFT is rarely applied directly as a result, despite providing a great case study for exploring quantum computing. A single study [12] used the QFT as a subroutine in a CFD solver. In this case, it was a Poisson solver based on a vortex-in-cell method in which the velocity field was evaluated from spatial derivatives calculated in the spectral space. The transitions between spatial and spectral domains were carried out by QFTs and the information about the 3D meshes was encoded on up to 24 qubits. The article does not provide details about the QFT algorithm, but states that its implementation scaled like the FFT. The solver tolerated 0.1% of simulated noise while producing meaningful solutions, but the number of samples used is not mentioned.

While the literature on quantum computing for CFD grows rapidly, to the authors' knowledge, no study has yet been published on methods of interest for turbomachinery CFD. Yet this field presents unique opportunities for the application of quantum algorithms, for example in Fourier methods, which make intensive use of Fourier transforms.

Sometimes called harmonic methods, Fourier methods have gained popularity in turbomachinery for solving unsteady time-periodic flow problems efficiently. While low-fidelity Reynolds-averaged Navier-Stokes (RANS) simulations remain the industry standard for turbomachinery design, the growing need to account for unsteady effects and component interactions makes unsteady simulations such as unsteady RANS (URANS) and large eddy simulation (LES) essential. However, these remain computationally expensive for many applications. Fourier methods address this issue by trading some fidelity for a substantial reduction in computation time. Gains of one to two orders of magnitude compared to conventional time-marching methods are typically reported while maintaining engineering accuracy.

This efficiency is achieved by exploiting periodicities in time (e.g., blade row interactions) and space (e.g., axisymmetry) that characterise turbomachinery flows. The flow solution is decomposed into a truncated Fourier series of its main harmonics in time, which is injected into the governing equations. Harmonics are then balanced, i.e., terms of the same frequency are solved together. This transforms the unsteady problem into a set of coupled steady-state equations, which can be solved using computationally efficient steady-state methods. The main requirement of Fourier methods is to have knowledge of the base frequencies of the dominant phenomena affecting the flow, and few harmonics are usually sufficient to model unsteady flows accurately. Several methods exist and differ in their capabilities and their treatments of the harmonic coefficients.

In the harmonic balance (HB) method [13], for instance, Fourier coefficients are defined as the Fourier transform of the flow solution sampled uniformly over the fundamental time period. The method thus consists in solving the governing equations for snapshots of the flow at equally spaced time instants. Reviews of the development and applications of Fourier methods (and of the HB method in particular) up to the 2010s can be found in Refs. [14,15], while Refs. [16,17] provide examples of more recent developments.

This work addresses the gap in research on quantum algorithms for turbomachinery CFD by studying an application of the QFT in the harmonic balance method. Its purpose is to gain a preliminary understanding of the opportunities and challenges presented by quantum computing, and to develop skills in the design, implementation and study of quantum algorithms.

A one-dimensional, time-periodic pulsating flow governed by Burgers' equation serves as a model problem. It is solved using a numerical solver that implements the HB method with time-periodic boundary conditions, where DFTs are approximated by a hybrid algorithm based on the QFT. The solver is implemented in Python and quantum circuits are simulated with Qiskit [18]. Three novel algorithms are presented to estimate Fourier coefficients, providing complete knowledge of their amplitudes and phases, which is not possible with the standard QFT. Their behaviour is studied both theoretically and numerically in the absence of noise, and numerically with a general noise model (depolarising noise). The solver's behaviour and its solutions are studied for a baseline case using the best-performing algorithm in a noiseless setting, before the impact of various solver parameters on convergence and solution quality is assessed. In particular, the number of samples used in the algorithm is varied, and solver performance in a noisy environment is evaluated.

The authors emphasise that no complexity advantage is sought from the proposed algorithms due to the extensive sampling required and the short signals used. Instead, this case study allows to explore the development of a hybrid CFD solver and to investigate its performance with meaningful parameters using limited computational power, while highlighting the main effects of quantum computing on the solution. Moreover, the results of this study should inform future research on the design of more complex quantum algorithms for Fourier methods, on methods better suited to leveraging the QFT and, more broadly, on the development of hybrid CFD solvers.

The remainder of this paper is structured as follows. Section 2 briefly defines the model problem, then the mathematical framework of the HB method is described in Section 3 and its numerical implementation in Section 4. Section 5 provides a brief overview of quantum computing before Section 6 discusses the QFT, which is followed by the presentation of the designed QFT algorithms in Section 7. The numerical configuration used in this work is described in Section 8. The performance of the algorithms is reviewed in Section 9, based on deeper theoretical and numerical studies detailed in Appendices A and B. The hybrid solver behaviour is finally studied in Section 10.

2. Problem definition

The model problem considered in this study is a one-dimensional, time-periodic pulsating flow governed by Burgers' equation. Because it is non-linear and allows for discontinuities in the inviscid limit, Burgers' equation is regarded as an elementary model for more elaborate conservation equations governing fluids, such as the Navier-Stokes equations. In non-dimensional conservation form, it reads

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(\frac{u^2}{2}\right) = \frac{1}{Re}\frac{\partial^2 u}{\partial x^2},$$

where x, t and $u(x,t)$ are already non-dimensionalised by constants L, T and U, respectively, and Re is the Reynolds number.

The studied flow enters the domain with a sinusoidal variation between zero and an arbitrary value, and leaves freely. The case may serve, for instance, as a low-order model for the oscillating flow through a single inter-blade passage of a surging compressor close to flow reversal. The variable u may be interpreted as the flow velocity, non-dimensionalised by its peak velocity U. For a domain of length L and an oscillation of period T, this situation is modelled by the boundary conditions

$$u(0,t) = \frac{1}{2} + \frac{1}{2}\cos(\omega t) \qquad \text{and} \qquad \frac{\partial u}{\partial x}(L,t) = 0, \tag{1}$$

where $\omega = 2\pi/T$.

3. Harmonic balance method

The mathematical formulation of the original time-spectral harmonic balance method, developed by Hall et al. [13], is adapted below for the study of a one-dimensional time-periodic scalar problem. For a more general formulation and additional details, refer to Hall et al. [15]. The method first expands the unknown problem solution as a Fourier series truncated to N_h harmonics of the flow's fundamental frequency, which is substituted into the governing equation. By balancing the harmonics, the unsteady equation is then converted into a set of $2N_h + 1$ coupled steady-state equations. The system is finally recast to the time domain. The Fourier coefficients are defined as the Fourier transform of the solution sampled uniformly over the fundamental time period, and the equations are coupled via the resulting spectral operator that approximates the time derivative.

Starting from the beginning, a one-dimensional scalar conservation law takes the general form

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0, \tag{2}$$

with f the non-linear flux. Consider a time-periodic problem of fundamental period T. Its solution u can be approximated as a Fourier series in time truncated to N_h harmonics,

$$u(x,t) \approx \mathrm{Re}\left(\sum_{n=0}^{N_h} \hat{u}_n(x)\, e^{i\omega n t}\right),$$

where $\hat{u}_n$ is the nth spatially-varying Fourier coefficient. The sum can be restricted to positive frequencies since the solution is real. Indeed, the Fourier transform of a real function is Hermitian, therefore $\hat{u}_{-n} = \hat{u}_n^\dagger$ (Hermitian symmetry), with $\hat{u}_n^\dagger$ the complex conjugate of $\hat{u}_n$. Injecting the above series into $f(u)$ and discarding the high-order harmonics (which originate from the non-linear coupling) gives

$$f(x,t) \approx \mathrm{Re}\left(\sum_{n=0}^{N_h} \hat{f}_n(x)\, e^{i\omega n t}\right).$$

Note that $\hat{f}_n \neq f(\hat{u}_n)$ as it depends on all $\hat{u}_n$. Substituting the previous expansions into Eq. (2) and balancing (solving for) each harmonic yields

$$i\omega n\, \hat{u}_n + \frac{\partial \hat{f}_n}{\partial x} = 0, \qquad n = 0, 1, \dots, N_h. \tag{3}$$

This system consists of $2N_h + 1$ steady-state equations, coupled through the non-linear fluxes, in $2N_h + 1$ variables (due to the aforementioned Hermitian symmetry). The variables are the complex Fourier coefficients, i.e., the mean value (zeroth harmonic) and $2N_h$ harmonic variables (real and imaginary parts, or amplitudes and phases).

Rather than solving for the coefficients $\hat{u}_n$ directly, these are defined as the DFT of the solution u sampled at $N_t = 2N_h + 1$ discrete time levels that are equally spaced over the fundamental period. Introducing the harmonic index vector $\boldsymbol{h} = (0, 1, \dots, N_h, -N_h, \dots, -1)$, the time level vector $\boldsymbol{t} = \boldsymbol{h}T/N_t$, the vectors of Fourier coefficients $\hat{\boldsymbol{u}} = (\hat{u}_n)_{n\in\boldsymbol{h}}$ and $\hat{\boldsymbol{f}} = (\hat{f}_n)_{n\in\boldsymbol{h}}$, the vector of sampled solutions $\boldsymbol{u}^*(x) = (u(x,t))_{t\in\boldsymbol{t}}$, and the DFT operator $\mathscr{F}$, this means that $\hat{\boldsymbol{u}} = \mathscr{F}\,\boldsymbol{u}^*$. The unknowns thus change from the frequency content of the conserved variable to its value at discrete time instants.

With these definitions, Eq. (3) can be rewritten as

$$\Omega\,\hat{\boldsymbol{u}} + \frac{\partial \hat{\boldsymbol{f}}}{\partial x} = 0,$$

where $\Omega = i\omega\,\mathrm{diag}(\boldsymbol{h})$. This system can be recast to the time domain by taking the inverse DFT of both sides, which results in

$$\mathscr{D}\,\boldsymbol{u}^* + \frac{\partial \boldsymbol{f}^*}{\partial x} = 0,$$

where

$$\mathscr{D}\cdot = \mathscr{F}^{-1}\Omega\,\mathscr{F}\cdot \tag{4}$$

is the spectral time derivative operator, with $\mathscr{F}^{-1}$ the inverse DFT operator, and $\boldsymbol{f}^*$ is the vector of sampled fluxes, here defined as $\boldsymbol{f}^* = f(\boldsymbol{u}^*)$. The equations are now coupled via the Fourier transforms rather than the fluxes. This system can then be solved in a pseudo-transient approach using efficient steady-state time-marching techniques. To do so, a pseudo-time derivative is introduced, giving

$$\frac{\partial \boldsymbol{u}^*}{\partial \tau} + \mathscr{D}\,\boldsymbol{u}^* + \frac{\partial \boldsymbol{f}^*}{\partial x} = 0, \tag{5}$$

with τ the pseudo-time. This term will be driven to zero as $\boldsymbol{u}^*$ converges to its steady-state solution.

4. Numerical solver

Eq. (5) is discretised in space using the finite volume method as

$$\frac{\partial \boldsymbol{U}}{\partial \tau} + \mathscr{D}\,\boldsymbol{U} + \boldsymbol{R}(\boldsymbol{U}) = 0,$$

where $\boldsymbol{U}$ and $\boldsymbol{R}(\boldsymbol{U})$ store the values of $\boldsymbol{u}^*$ and of the numerical flux residuals, respectively, at every cell centre. The domain is made of N_x cells of size $\Delta x = L/N_x$. The DFTs in operator $\mathscr{D}$ are calculated using the FFT algorithm in the classical setting or one of the QFT algorithms (developed in Section 7) in the hybrid setting. The numerical flux residual at cell i is

$$\boldsymbol{r}_i(\boldsymbol{U}) = \underbrace{\frac{1}{\Delta x}\left(\boldsymbol{f}^*_{i+\frac{1}{2}} - \boldsymbol{f}^*_{i-\frac{1}{2}}\right)}_{\boldsymbol{r}_{a,i}} - \underbrace{\frac{1}{\Delta x Re}\left(\left.\frac{\partial \boldsymbol{u}^*}{\partial x}\right|_{i+\frac{1}{2}} - \left.\frac{\partial \boldsymbol{u}^*}{\partial x}\right|_{i-\frac{1}{2}}\right)}_{\boldsymbol{r}_{d,i}},$$

where in the advective term $\boldsymbol{r}_{a,i}$ the Riemann problems at cell interfaces $(i \pm 1/2)$ are solved using Godunov's method [19], and in the diffusive

term $\boldsymbol{r}_{d,i}$ the gradients are discretised using first-order central differences. The boundary conditions defined by Eq. (1) are extrapolated in ghost cells using first-order finite differences.

The system of equations is marched to its steady-state solution using the backward Euler method. The solution at iteration $(k+1)$ is thus

$$\boldsymbol{U}^{k+1} = \boldsymbol{U}^k - \Delta\tau \left(\mathscr{D}\boldsymbol{U}^{k+1} + \boldsymbol{R}\left(\boldsymbol{U}^{k+1}\right)\right), \tag{6}$$

with $\Delta\tau$ the pseudo-time step. To solve this implicit system, $\mathscr{D}\boldsymbol{U}^{k+1}$ and $\boldsymbol{R}(\boldsymbol{U}^{k+1})$ are linearised around $\boldsymbol{U}^k$ as

$$\mathscr{D}\boldsymbol{U}^{k+1} = \mathscr{D}\boldsymbol{U}^k + \left.\frac{\partial(\mathscr{D}\boldsymbol{U})}{\partial \boldsymbol{U}}\right|_{U=U^k} \Delta\boldsymbol{U}^k + \mathscr{O}\left(\left(\Delta\boldsymbol{U}^k\right)^2\right),$$
$$\boldsymbol{R}\left(\boldsymbol{U}^{k+1}\right) = \boldsymbol{R}\left(\boldsymbol{U}^k\right) + \left.\frac{\partial \boldsymbol{R}(\boldsymbol{U})}{\partial \boldsymbol{U}}\right|_{U=U^k} \Delta\boldsymbol{U}^k + \mathscr{O}\left(\left(\Delta\boldsymbol{U}^k\right)^2\right).$$

The Jacobians

$$\left.\frac{\partial(\mathscr{D}\boldsymbol{U})}{\partial \boldsymbol{U}}\right|_{U=U^k} = \left(\mathscr{F}^{-1}\Omega\mathscr{F} I_{N_t}\right) \otimes I_{N_x}, \tag{7}$$

with I_ν the identity matrix of size $\nu \times \nu$, and $\partial\boldsymbol{R}(\boldsymbol{U})/\partial\boldsymbol{U}|_{U=U^k}$ are calculated analytically. Note that in the latter, only the advective term is a function of $\boldsymbol{U}^k$. Substituting the truncated linearisations into Eq. (6) finally yields

$$\Delta\boldsymbol{U}^k = \boldsymbol{U}^{k+1} - \boldsymbol{U}^k = A^{-1}\left(\mathscr{D}\boldsymbol{U}^k + \boldsymbol{R}\left(\boldsymbol{U}^k\right)\right), \tag{8a}$$

$$A = \frac{I_{N_t N_x}}{\Delta\tau} + \left.\frac{\partial(\mathscr{D}\boldsymbol{U})}{\partial \boldsymbol{U}} + \frac{\partial \boldsymbol{R}(\boldsymbol{U})}{\partial \boldsymbol{U}}\right|_{U=U^k}. \tag{8b}$$

5. Overview of quantum computing

A brief overview of quantum computing is given below for the unfamiliar reader. A more comprehensive introduction can be found in Nielsen and Chuang [20].

Quantum computing exploits the quantum mechanical properties of quantum systems to process information. Unlike classical systems, a quantum system is inherently probabilistic. This is because its physical properties are not definite but exist in superposition, i.e., in multiple configurations simultaneously. A system is described by its quantum state, which encodes information about its possible outcomes and defines the probability distribution of observing each of them. An important characteristic of quantum systems made of multiple entities is that they can exhibit entanglement, a phenomenon where the entities become interdependent and cannot be described independently, regardless of the distance between them. A state evolves through both controlled and undesired physical interactions with its environment. The information contained in a state remains hidden to an observer until one of its properties is measured, which collapses the superposition to a single outcome that yields a numerical value but destroys the remaining information.

The algebraic notation used in quantum computing is the bra-ket notation commonly used in quantum mechanics. Mathematically, a state ψ is denoted by the ket $|\psi\rangle$ which represents a vector (called state vector) in a complex N-dimensional Hilbert space $\mathscr{H}^N$. Its conjugate transpose $\psi^\dagger$ is denoted by the bra $\langle\psi|$. The inner product between states ψ and ϕ is the bra-ket $\langle\phi|\psi\rangle$ and their tensor product the ket-bra $|\phi\rangle\langle\psi|$. The notation $\cdot^{\otimes n}$ means object $\cdot$ is tensored with itself n times.

The unit of information is the qubit. In contrast to classical bits that represent either 0 or 1, a qubit exists as a superposition of 0 and 1. Each qubit is represented as a linear combination $|\psi\rangle = a_0|0\rangle + a_1|1\rangle$ that spans a space $\mathscr{H}^2$ of basis $\{|0\rangle, |1\rangle\}$, called computational basis. An n-qubit state $|\psi\rangle = \sum_{j=0}^{N-1} a_j|j\rangle$ is defined on a space $\mathscr{H}^N = (\mathscr{H}^2)^{\otimes n}$ of dimension $N = 2^n$ and basis $\{|0\rangle, |1\rangle\}^{\otimes n}$. The latter is often represented in decimal notation, $\{|j\rangle\}_{j=0,\dots,N-1}$. The coefficients a_j are the complex probability amplitudes of the basis states $|j\rangle$, and obey $\sum_j |a_j| = 1$ because quantum states are normalised by definition.[2]

[2] Strictly speaking, qubits are therefore defined on projected Hilbert spaces.

The quantum circuit (or gate-based) model is used in this work. Qubits are first arbitrarily initialised before their states are evolved by applying a sequence of quantum logic gates. These are unitary transformations, i.e., linear operators that preserve the inner product without loss of information, and are thus reversible. Qubits are finally measured. The qubits used in the circuit are grouped into quantum registers, while the measurement results are stored in classical registers.

Gates can be divided into two categories: basis gates, which perform the basic operations available on a given quantum computer, and composite gates, whose operations are approximated from basis gates. Basis gates act on single (or sometimes two) qubits, while composite gates can act on any number of qubits. The circuits presented in this paper make use of three common single-qubit gates. The NOT gate X inverts the qubit state, $X|0\rangle = |1\rangle$ and vice versa. The phase shift gate $P(\theta)$ rotates state $|1\rangle$, $P(\theta)|1\rangle = e^{i\theta}|1\rangle$, and leaves state $|0\rangle$ unchanged. The Hadamard gate H puts the qubit in equal superposition, $H|0\rangle = |+\rangle$ and $H|1\rangle = |-\rangle$, where $|\pm\rangle = (|0\rangle \pm |1\rangle)/\sqrt{2}$. Several multi-qubit gates are also used in the circuits and will be introduced later. A control qubit can be added to a gate to conditionally apply its operation on a target qubit based on the state of the control qubit. This may entangle the control and target qubit states. For instance, adding a single control qubit to X gives the two-qubit controlled-X gate $CX = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X$, with I the identity matrix. The control qubit is often just called 'control'.

The first stage of a quantum circuit is generally state preparation, which is the process of preparing a quantum state whose probability amplitudes encode the values of a normalised complex vector, such as $\boldsymbol{a}^* = \boldsymbol{u}^*/\|\boldsymbol{u}^*\|_2$. This is achieved by using an initialisation algorithm that builds a circuit tailored to prepare the state.

Measurements can be made in any basis spanned by orthonormal states, and amount to projecting the measured state onto the basis states. The three usual measurement basis are the computational basis, the Hadamard basis $\{|+\rangle, |-\rangle\}^{\otimes n}$, and the circular basis $\{|+i\rangle, |-i\rangle\}^{\otimes n}$, where $|\pm i\rangle = (|0\rangle \pm i|1\rangle)/\sqrt{2} = P(\pi/2)|\pm\rangle$. If $\{|b_j\rangle\}_{j=0,\dots,N-1}$ is a measurement basis of $\mathscr{H}^N$, the probability of observing state $|b_j\rangle$ when measuring $|\psi\rangle$ is given by $\mathrm{P}_\psi(|b_j\rangle) = \langle\psi|P_{b_j}|\psi\rangle = |\langle b_j|\psi\rangle|^2$, where $P_{b_j} = |b_j\rangle\langle b_j|$ is the projector onto $|b_j\rangle$. The probability of observing $|j\rangle$ in the computational basis is thus $|a_j|^2$. Any uniform phase information encoded in $|\psi\rangle$, called global phase, is discarded in the measurement process as global phases are unobservable in quantum mechanics. Only relative phases between basis states of $|\psi\rangle$ can be observed.

Circuits are developed to be run on a backend, i.e., a physical or virtual device that executes quantum circuits. It can either be a quantum computer, which directly manipulates quantum states, or a simulator, which emulates quantum behaviour on a classical computer. A quantum computer is made of qubits, but not all of them are connected together nor display the same performance. Indeed, the biggest challenge in building quantum computers for practical use is that errors can be introduced in the circuits in many ways. They can be caused by operational issues when applying gates or measuring qubits, by noise originating from internal components, by interference with the external environment, or by state decoherence. The latter is a loss of information caused by the state collapsing due to environmental interaction. Qubit performance can be poor because of these errors, and is moreover inhomogeneous as different qubits have different error rates, gate fidelities and decoherence times – so-called qubit variability.

For practical reasons, backends only implement a finite set of gates, preferably universal. A universal gate set is a collection of gates from which any other gate can be approximated to arbitrary accuracy. For example, IBM's current open systems implement the set $\{I, R_z, \sqrt{X}, X, CX\}$, with CX sometimes replaced [21].

In most cases, circuits are unsuited to the specific backend and cannot be readily executed. They must first be transpiled. Transpilation is the process of mapping and optimising a circuit from a high-level representation to the constraints of the provided backend. It involves approximating and reordering gates to match the implemented set and the device topology, dealing with qubit variability, and mitigating

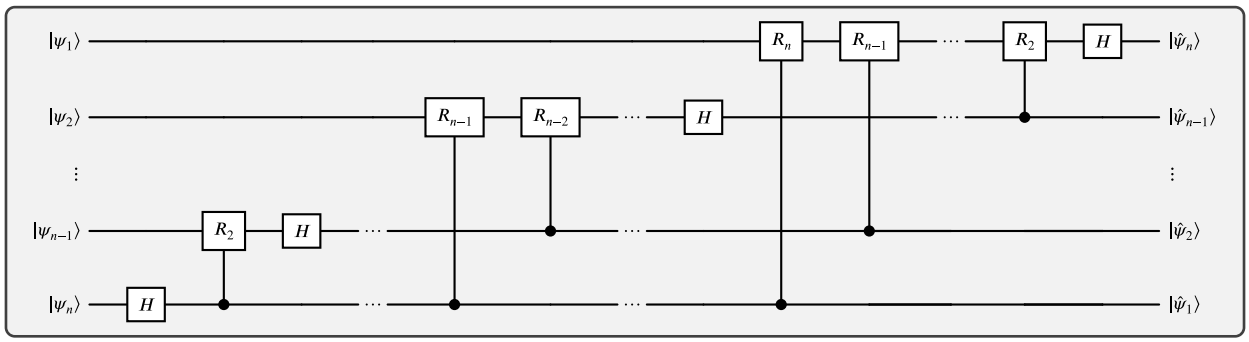


Fig. 1. QFT circuit, where the gate $R_l = P(-2\pi i/2^l)$ rotates its target qubit depending on the state of its control qubit. This circuit is the inverse of the standard QFT circuit [20].

errors preventively to improve circuit performance. Once transpiled the circuit can be run, and each successful execution will provide a single numerical value. It must be stressed that the performance of quantum circuits is highly backend-dependent.

Sampling is the process of executing a circuit many times to derive a statistical estimate of the probability distribution of the measured state. The number of times a circuit is run is usually called the number of shots or of circuit repetitions. The number of samples n_s is the number of successful executions. The estimated probability of observing state $|b_j\rangle$ is given by $\tilde{\mathrm{P}}_\psi(|b_j\rangle) = n_{b_j}/n_s \to \mathrm{P}_\psi(|b_j\rangle)$ as $n_s \to \infty$, with n_{b_j} the number of observations of $|b_j\rangle$. Because n_s is finite, sampling also introduces errors in the statistical approximation. The number of samples required for a sufficiently good approximation depends on the application and can be considerable, scaling as $\mathcal{O}(N)$ when complete information about the output state is sought.

Quantum states and operators can be represented in matrix notation, with kets and bras represented as column and row vectors, and gates as matrices. For instance, the states of the n-qubit computational basis are $|j\rangle = (\delta_{0j}\ \delta_{1j}\ \cdots\ \delta_{(N-1)j})^T$, where δ_{ij} is the Kronecker delta, and the matrices associated with the gates mentioned above are

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sqrt{X} = \frac{1}{2}\begin{pmatrix} 1+i & 1-i \\ 1-i & 1+i \end{pmatrix}, \quad CX = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix},$$

$$P(\theta) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}, \quad H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad R_z(\theta) = e^{-i\frac{\theta}{2}} P(\theta).$$

The discussion so far has focused on systems whose states are completely known, called pure states. But knowledge about a system may be incomplete because of classical probabilistic uncertainty, noise, or decoherence. Such systems can be described by mixed states, which are statistical ensembles of pure states. For instance, in the presence of noise, a system may be in a pure state $|\psi\rangle$ with probability q, while with the remaining probability $(1-q)$ it would be in a different, erroneous state. However, unlike pure states, mixed states cannot be represented by state vectors and a different formalism is required.

The density matrix formalism defines density matrices ρ that can represent both pure and mixed states and take the form

$$\rho = \sum_r p_r\, |\psi_r\rangle\langle\psi_r|,$$

where p_r is the probability of the system being in the pure state $|\psi_r\rangle$. Density matrices are used in this work to subject algorithms to noise.

The effects of various noise sources can be emulated by means of noise models defined as quantum operations ε, also known as channels. Ideally, models should be adapted to the specific architecture of quantum computers to best reflect the noise affecting them. Nevertheless, this study focuses on general trends and results, and as such aims to abstract from hardware considerations. A fairly standard model along these lines is depolarising noise, in which there is a slight probability of losing all information about the system. The depolarising noise channel acting on a mixed state ρ is defined as

$$\varepsilon(\rho) = p_d\, \frac{I}{2^n} + (1 - p_d)\, \rho, \tag{9}$$

where p_d is the probability of the state being depolarised, i.e., replaced by the maximally mixed state $I/2^n$. This channel is equivalent to applying a uniformly random gate on each qubit with probability p_d. The depolarising noise model can therefore be understood as introducing purely random noise into the system, an action independent of any particular quantum architecture.

6. Quantum Fourier transform

Just as the discrete Fourier transform decomposes a signal into its constituent discrete frequencies, the quantum Fourier transform decomposes a quantum state into its constituent basis states. If the elements of a complex signal $\boldsymbol{a} = (a_j)_{j=0,\ldots,N-1}$ are encoded in the amplitudes of the input state $|\psi\rangle$, the amplitudes of the output state $|\hat{\psi}\rangle$ will encode the corresponding Fourier coefficients $\hat{\boldsymbol{a}} = (\hat{a}_k)_{k=0,1,\ldots,N-1}$. Conventions on the definition of the QFT vary but, in general, the forward QFT is defined in the same way as the inverse DFT and vice versa. For the sake of clarity, in what follows, the forward QFT relates to the forward DFT and vice versa. The QFT is computed by the circuit shown in Fig. 1, which thus prepares the state

$$|\hat{\psi}\rangle = \sum_{k=0}^{N-1} \hat{a}_k\, |k\rangle,$$

where $\hat{a}_k$ is defined as

$$\hat{a}_k = |\hat{a}_k|\, e^{i\hat{\varphi}_k} = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} a_j\, e^{\frac{-2\pi i j k}{N}}.$$

For a signal of N elements, the QFT has a complexity of $\mathcal{O}(\log^2 N)$ compared with $\mathcal{O}(N \log N)$ for the FFT, which represents an exponential advantage. This, however, does not account for state preparation or sampling. If $\tilde{\boldsymbol{a}} = (\tilde{a}_k)_{k=0,1,\ldots,N-1}$ is the estimate of $\hat{\boldsymbol{a}}$ obtained by sampling, the L2 error on the estimate is defined as

$$\epsilon_{\mathrm{L2}}^2 = \|\tilde{\boldsymbol{a}} - \hat{\boldsymbol{a}}\|_2^2 = \sum_{k=0}^{N-1} |\tilde{a}_k - \hat{a}_k|^2. \tag{10}$$

Based on generic bounds on optimal state estimation [22], the number of samples n_s required to estimate $\hat{\boldsymbol{a}}$ with error ϵ_{L2} scales as $\mathcal{O}(N/\epsilon_{\mathrm{L2}}^2)$. Combined with the complexity of the QFT, this leads to a worse overall complexity than the FFT, without even factoring in state preparation.

The QFT circuit only allows to estimate the amplitudes $|\hat{a}_k|$ of the Fourier coefficients, because their phases $\hat{\varphi}_k$ are encoded as global phases which cannot be observed. The QFT is therefore typically used in applications where amplitude estimates are sufficient. However, the harmonic balance solver needs complete knowledge of the complex coefficients, i.e, of their amplitudes and phases, or, equivalently, of their real and imaginary parts. This requires more elaborate circuits that make use of relative phases.

7. QFT algorithms

This section presents three QFT-based algorithms to estimate the complete complex coefficients of forward and inverse discrete Fourier transforms. To the authors' knowledge, no algorithm was previously proposed for this purpose in the literature.

All three algorithms function similarly and share common circuits and steps. They follow two stages. First, they estimate the amplitudes of the Fourier coefficients and their projections onto the axes of the trigonometric circle. Depending on the algorithm, this translates to estimating either the absolute values of the real and imaginary parts of the coefficients, or their amplitudes and phases up to a constant π (modulo π). The projections therefore introduce phase ambiguities, which are unavoidable as they originate from the measurement process and the way relative phases are encoded between basis states. In the second stage, the algorithms lift these ambiguities. This means that they either estimate the signs of the real and imaginary parts, or evaluate whether the phases must be corrected.

Algorithm QFT1 consists of two double steps and exploits the symmetry properties of complex signals and of their Fourier transforms. By decomposing the signal into its even and odd components, whose Fourier transforms are purely real and imaginary, the algorithm estimates the real and imaginary parts of the coefficients separately. The absolute values of each part are first estimated in steps 1.1R and 1.1I, then their signs are estimated in steps 1.2R and 1.2I.

Algorithm QFT2 works similarly to QFT1, but without decomposing the signal. The absolute values of the real and imaginary parts are estimated together in step 2.1, then their signs in steps 2.2R and 2.2I.

The main difference between QFT1 and QFT2 can be understood in terms of amplitudes and phases. In QFT1, the states encoding the complex coefficients are projected onto the real and imaginary axes deterministically as a consequence of the signal decomposition, whereas in QFT2 this projection is probabilistic. The main advantage of QFT1 is that the importance given to the estimation of the real and imaginary parts can be adjusted. In particular, half of the algorithm can be skipped when the coefficients are known to be purely real or imaginary.

Algorithm QFT3 is slightly different and consists of two steps, the first of which is double. Steps 3.1C and 3.1S together estimate the amplitudes of the coefficients and their phases up to a constant π. This ambiguity is lifted in step 3.2 which evaluates whether the ambiguous phases are in phase or in opposition with the true phases, that is, whether the final phase estimates should be the phases calculated in step 3.1 or those phases plus π.

An algorithm uses a total of $n_s = \sum_m n_{s,m}$ samples, where the circuit used in step m is sampled with $n_{s,m}$ samples. Estimated quantities are denoted with a tilde in what follows, for example, $\tilde{a}_k \approx \hat{a}_k$ and $\tilde{\varphi}_k \approx \hat{\varphi}_k$.

Note that a signal $\boldsymbol{u}$ with $\|\boldsymbol{u}\|_2 \neq 1$ must be normalised as $\boldsymbol{a} = \boldsymbol{u}/\|\boldsymbol{u}\|_2$ before being encoded in a quantum register. Its Fourier transform is then denormalised as $\hat{\boldsymbol{u}} = \hat{\boldsymbol{a}}\,\|\boldsymbol{u}\|_2$. Also, a signal whose length is not a power of two must be padded with zeros, with the understanding that its Fourier transform will be different from that of the original signal.

7.1. Circuits

All circuits use an n-qubit working register to encode the $N = 2^n$ elements of a complex signal $\boldsymbol{a} = (a_j)_{j=0,\dots,N-1}$ and carry out computations. Circuit QFT-A contains only this register whereas circuits QFT-B, QFT-C and QFT-D use an additional control qubit. Qubits are initialised in the all-zero state $|0\rangle^{\otimes n}$ and are measured at the end of the circuits. The working register is consistently measured in the computational basis while the control qubit is measured in either the Hadamard or circular basis.

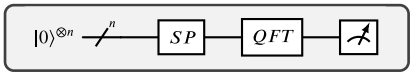


Fig. 2. Circuit QFT-A. The n-qubit working register encodes the Fourier coefficients.

7.1.1. Gate definitions

Recall that X and H are the NOT and Hadamard gates which respectively invert a qubit and put it in equal superposition. Applying Hadamard gates to multiple qubits placed in the all-zero state puts them all in equal superposition,

$$H^{\otimes n}|0\rangle^{\otimes n} = \sum_{k=0}^{N-1} \frac{1}{\sqrt{2^n}} |k\rangle\,.$$

The elements of signal $\boldsymbol{a}$ and of its complex conjugate $\boldsymbol{a}^\dagger$ can be encoded in the complex probability amplitudes of states $|\psi\rangle$ and $|\psi^\dagger\rangle$ by means of state preparation circuits, as mentioned in Section 5. These are implemented by the SP and $\overline{SP}$ gates. Applying them to the all-zero state results in

$$|\psi\rangle = SP\,|0\rangle^{\otimes n} = \sum_{j=0}^{N-1} a_j|j\rangle = \sum_{j=0}^{N-1} |a_j|\,e^{i\varphi_j}|j\rangle\,,$$

$$|\psi^\dagger\rangle = \overline{SP}\,|0\rangle^{\otimes n} = \sum_{j=0}^{N-1} a_j^\dagger|j\rangle = \sum_{j=0}^{N-1} |a_j|\,e^{-i\varphi_j}|j\rangle\,.$$

The QFT can be computed using the QFT circuit shown in Fig. 1, which is implemented by the QFT gate. Its conjugate version $\overline{QFT}$ is defined to implement the same circuit with rotations opposite to those of the QFT, i.e., where rotations R_l are replaced by $R_l^\dagger = P(2\pi i/2^l)$. Applying these gates to states $|\psi\rangle$ and $|\psi^\dagger\rangle$ yields

$$|\hat{\psi}\rangle = QFT\,|\psi\rangle = \sum_{k=0}^{N-1} \hat{a}_k|k\rangle = \sum_{k=0}^{N-1} |\hat{a}_k|\,e^{i\hat{\varphi}_k}|k\rangle\,,$$

$$|\hat{\psi}^\dagger\rangle = \overline{QFT}\,|\psi^\dagger\rangle = \sum_{k=0}^{N-1} \hat{a}_k^\dagger|k\rangle = \sum_{k=0}^{N-1} |\hat{a}_k|\,e^{-i\hat{\varphi}_k}|k\rangle\,.$$

An estimate $\tilde{\hat{a}}_k = |\tilde{a}_k|\exp(i\tilde{\hat{\phi}}_k)$ of the coefficients $\hat{a}_k$ can be encoded in state $|\tilde{\psi}\rangle$ by the $\widetilde{SP}$ gate. Applying it to the all-zero state gives

$$|\tilde{\psi}\rangle = \widetilde{SP}\,|0\rangle^{\otimes n} = \sum_{k=0}^{N-1} |\tilde{a}_k|\,e^{i\tilde{\hat{\phi}}_k}|k\rangle\,.$$

7.1.2. Circuit QFT-A

Shown in Fig. 2, circuit QFT-A encodes the Fourier coefficients in the working register, preparing the state

$$|\Psi_\mathrm{A}\rangle = |\hat{\psi}\rangle = QFT\,SP\,|0\rangle^{\otimes n} = \sum_{k=0}^{N-1} \hat{a}_k|k\rangle\,.$$

When sampling the circuit, states $|k\rangle$ are observed with probabilities

$$\mathrm{P}_{\Psi_\mathrm{A}}(|k\rangle) = \left|\langle k|\Psi_\mathrm{A}\rangle\right|^2 = |\hat{a}_k|^2\,. \tag{11}$$

7.1.3. Circuit QFT-B

In circuit QFT-B, shown in Fig. 3, the control qubit is put in equal superposition (state $|+\rangle = H|0\rangle$) and controls the $H^{\otimes n}$, SP and QFT gates. Depending on its state, the working qubits are either put in equal superposition or encode the Fourier coefficients. This is defined as

$$U_B = |0\rangle\langle 1| \otimes H^{\otimes n} + |1\rangle\langle 0| \otimes QFT\,SP\,,$$

where $|0\rangle\langle 1| = X|1\rangle\langle 1|$ and $|1\rangle\langle 0| = X|0\rangle\langle 0|$. The circuit thus prepares

$$\begin{aligned}|\Psi_\mathrm{B}\rangle &= U_B\left(|+\rangle \otimes |0\rangle^{\otimes n}\right)\\ &= \sum_{k=0}^{N-1} \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2^n}}|0\rangle + |\hat{a}_k|\,e^{i\hat{\varphi}_k}|1\rangle\right) \otimes |k\rangle\,.\end{aligned} \tag{12}$$

When sampled with the control qubit measured in the Hadamard basis, states $|\pm\rangle \otimes |k\rangle$ are observed with probabilities

$$\begin{aligned} \mathrm{P}_{\Psi_\mathrm{B}}(|\pm\rangle \otimes |k\rangle) &= \left| \left(\frac{\langle 0| \pm \langle 1|}{\sqrt{2}} \otimes \langle k| \right) \right. \\ &\qquad \left. \times \left(\sum_{k=0}^{N-1} \frac{1}{\sqrt{2}} \left(\frac{1}{\sqrt{2^n}} |0\rangle + |\hat{a}_k| \, e^{i\hat{\varphi}_k} |1\rangle \right) \otimes |k\rangle \right) \right|^2 \\ &= \left| \frac{1}{2} \left(\frac{1}{\sqrt{2^n}} \pm |\hat{a}_k| \, e^{i\hat{\varphi}_k} \right) \right|^2 \\ &= \frac{1}{4} \left(\frac{1}{2^n} + |\hat{a}_k|^2 \pm \frac{2}{\sqrt{2^n}} |\hat{a}_k| \cos \hat{\varphi}_k \right). \end{aligned} \tag{13}$$

Similarly, when sampled with the control measured in the circular basis, states $|\pm i\rangle \otimes |k\rangle$ are observed with probabilities

$$\mathrm{P}_{\Psi_\mathrm{B}}(|\pm i\rangle \otimes |k\rangle) = \frac{1}{4} \left(\frac{1}{2^n} + |\hat{a}_k|^2 \pm \frac{2}{\sqrt{2^n}} |\hat{a}_k| \sin \hat{\varphi}_k \right). \tag{14}$$

7.1.4. Circuit QFT-C

Circuit QFT-C is shown in Fig. 4. Depending on the state of the control qubit put in equal superposition, the working register either encodes the Fourier coefficients or their conjugates,[3]

$$U_C = |0\rangle\langle 1| \otimes \overline{QFT}\,\overline{SP} + |1\rangle\langle 0| \otimes QFT\,SP\,.$$

The circuit thus prepares

$$|\Psi_\mathrm{C}\rangle = \sum_{k=0}^{N-1} |\hat{a}_k| \, e^{-i\hat{\varphi}_k} \frac{|0\rangle + e^{2i\hat{\varphi}_k}|1\rangle}{\sqrt{2}} \otimes |k\rangle\,.$$

When sampled with the control measured in the Hadamard basis, states $|\pm\rangle \otimes |k\rangle$ are observed with probabilities

$$\mathrm{P}_{\Psi_\mathrm{C}}(|\pm\rangle \otimes |k\rangle) = |\hat{a}_k|^2 \frac{1 \pm \cos 2\hat{\varphi}_k}{2}\,. \tag{15}$$

When sampled with the control measured in the circular basis, states $|\pm i\rangle \otimes |k\rangle$ are observed with probabilities

$$\mathrm{P}_{\Psi_\mathrm{C}}(|\pm i\rangle \otimes |k\rangle) = |\hat{a}_k|^2 \frac{1 \pm \sin 2\hat{\varphi}_k}{2}\,. \tag{16}$$

7.1.5. Circuit QFT-D

Circuit QFT-D is shown in Fig. 5. Depending on the state of the control qubit put in equal superposition, the working register either encodes an ambiguous estimate $\tilde{\tilde{a}}_k = |\tilde{a}_k| \exp(i\tilde{\tilde{\varphi}}_k)$ of the Fourier coefficients, or their true values,

$$U_D = |0\rangle\langle 1| \otimes \widetilde{SP} + |1\rangle\langle 0| \otimes QFT\,SP\,.$$

The circuit thus prepares

$$|\Psi_\mathrm{D}\rangle = \sum_{k=0}^{N-1} \frac{1}{\sqrt{2}} \left(|\tilde{a}_k| \, e^{i\tilde{\tilde{\varphi}}_k} |0\rangle + |\hat{a}_k| \, e^{i\hat{\varphi}_k} |1\rangle \right) \otimes |k\rangle\,.$$

When sampled with the control measured in the Hadamard basis, states $|\pm\rangle \otimes |k\rangle$ are observed with probabilities

$$\begin{aligned} \mathrm{P}_{\Psi_\mathrm{D}}(|\pm\rangle \otimes |k\rangle) &= \frac{1}{4} \left(\left(|\tilde{a}_k| \cos \tilde{\tilde{\varphi}}_k \pm |\hat{a}_k| \cos \hat{\varphi}_k \right)^2 \right. \\ &\qquad \left. + \left(|\tilde{a}_k| \sin \tilde{\tilde{\varphi}}_k \pm |\hat{a}_k| \sin \hat{\varphi}_k \right)^2 \right) \\ &= \frac{1}{4} \left(|\tilde{a}_k|^2 + |\hat{a}_k|^2 \pm 2 \, |\tilde{a}_k| \, |\hat{a}_k| \cos\left(\tilde{\tilde{\varphi}}_k - \hat{\varphi}_k \right) \right), \end{aligned} \tag{17}$$

where the identity $\cos(\alpha - \beta) = \cos\alpha \cos\beta - \sin\alpha \sin\beta$ has been used.

[3] Note that for real-valued signals, gates SP and $\overline{SP}$ prepare the same state, so SP can be applied without control for greater efficiency.

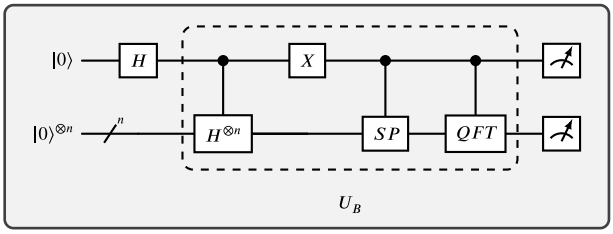


Fig. 3. Circuit QFT-B. Depending on the state of the control qubit put in equal superposition, the qubits of the n-qubit working register are either put in equal superposition or encode the Fourier coefficients.

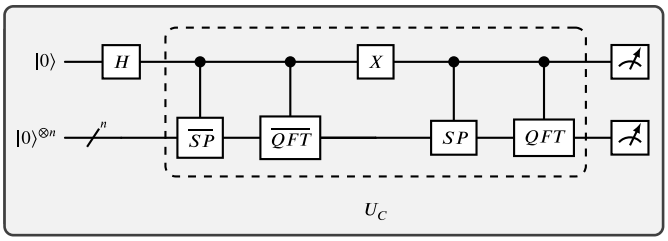


Fig. 4. Circuit QFT-C. Depending on the state of the control qubit put in equal superposition, the n-qubit working register either encodes the Fourier coefficients or their complex conjugates.

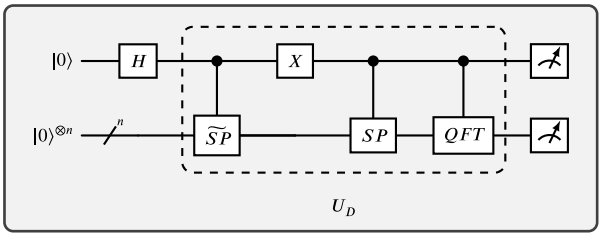


Fig. 5. Circuit QFT-D. Depending on the state of the control qubit put in equal superposition, the n-qubit working register either encodes an ambiguous estimate of the Fourier coefficients, or their true values.

7.2. Algorithm QFT1

The signal $\boldsymbol{a}$ is first decomposed and renormalised into its even and odd components $\boldsymbol{a}_e$ and $\boldsymbol{a}_o$ in order to estimate the real and imaginary parts of the Fourier coefficients separately. This calls for a few additional definitions.

7.2.1. Additional definitions

The Fourier coefficients of the even and odd components of the signal are defined as

$$\hat{a}_{e,k} = \frac{1}{R} |\hat{a}_k| \cos \hat{\varphi}_k \qquad \text{and} \qquad \hat{a}_{o,k} = \frac{1}{I} |\hat{a}_k| \sin \hat{\varphi}_k\,,$$

where

$$R = \sqrt{\sum_{k=0}^{N-1} |\hat{a}_k|^2 \cos^2 \hat{\varphi}_k} \qquad \text{and} \qquad I = \sqrt{\sum_{k=0}^{N-1} |\hat{a}_k|^2 \sin^2 \hat{\varphi}_k}\,.$$

These are also equal to the norms of $\boldsymbol{a}_e$ and $\boldsymbol{a}_o$, respectively. Note that $R^2 + I^2 = 1$. The coefficients of the original signal can be recovered as

$$\hat{a}_k = R\hat{a}_{e,k} + iI\hat{a}_{o,k}\,.$$

The quantum states associated with the even and odd components,

$$|\hat{\psi}_R\rangle = \sum_{k=0}^{N-1} \hat{a}_{e,k} |k\rangle \qquad \text{and} \qquad |\hat{\psi}_I\rangle = \sum_{k=0}^{N-1} \hat{a}_{o,k} |k\rangle, \tag{18}$$

can be prepared by circuit QFT-A with the SP gate encoding $\boldsymbol{a}_e$ and $\boldsymbol{a}_o$, respectively. This gives $|\Psi_{\mathrm{A,R}}\rangle = |\hat{\psi}_R\rangle$ and $|\Psi_{\mathrm{A,I}}\rangle = |\hat{\psi}_I\rangle$.

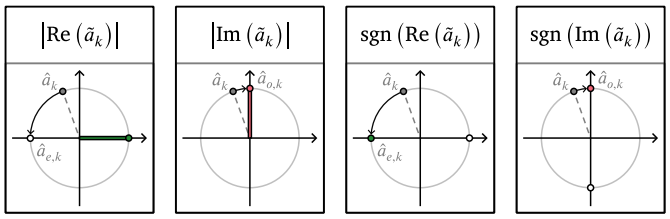


Fig. 6. Illustration of algorithm QFT1. Signal $\boldsymbol{a}$ is first decomposed and renormalised into its even and odd components $\boldsymbol{a}_e$ and $\boldsymbol{a}_o$. In steps (a) 1.1R and (b) 1.1I, estimates of the absolute values of the real and imaginary parts of the Fourier coefficients $\hat{a}_k$ of $\boldsymbol{a}$ are derived from estimates of the absolute values of the coefficients $\hat{a}_{e,k}$ of $\boldsymbol{a}_e$ and $\hat{a}_{o,k}$ of $\boldsymbol{a}_o$, respectively. In steps (c) 1.2R and (d) 1.2I, estimates of the signs of the real and imaginary parts of $\hat{a}_k$ are derived from estimates of the projections of the coefficients $\hat{a}_{e,k}$ and $\hat{a}_{o,k}$ onto the real and imaginary axes, respectively.

Reworking Eq. (12), the execution of circuit QFT-B with SP encoding $\boldsymbol{a}_e$ and $\boldsymbol{a}_o$ respectively prepares

$$|\Psi_{\mathrm{B},R}\rangle = \sum_{k=0}^{N-1} \frac{1}{\sqrt{2}} \left(\frac{1}{\sqrt{2^n}} |0\rangle + \hat{a}_{e,k}|1\rangle \right) \otimes |k\rangle ,$$

$$|\Psi_{\mathrm{B},I}\rangle = \sum_{k=0}^{N-1} \frac{1}{\sqrt{2}} \left(\frac{1}{\sqrt{2^n}} |0\rangle + i\hat{a}_{o,k}|1\rangle \right) \otimes |k\rangle .$$

When sampled with the control measured in the Hadamard and circular bases, states $|\pm\rangle \otimes |k\rangle$ and $|\pm i\rangle \otimes |k\rangle$ are observed with probabilities

$$\mathrm{P}_{\Psi_{\mathrm{B},R}} (|\pm\rangle \otimes |k\rangle) = \frac{1}{4} \left(\frac{1}{2^n} + \hat{a}_{e,k}^2 \pm \frac{2}{\sqrt{2^n}} \hat{a}_{e,k} \right), \tag{19}$$

$$\mathrm{P}_{\Psi_{\mathrm{B},I}} (|\pm i\rangle \otimes |k\rangle) = \frac{1}{4} \left(\frac{1}{2^n} + \hat{a}_{o,k}^2 \pm \frac{2}{\sqrt{2^n}} \hat{a}_{o,k} \right). \tag{20}$$

7.2.2. Algorithm

The algorithm is made of two double steps illustrated in Fig. 6. Steps 1.1R and 1.1I estimate the absolute values of the real and imaginary parts, then steps 1.2R and 1.2I estimate their signs.

Step 1.1R. Circuit QFT-A is first sampled with SP encoding $\boldsymbol{a}_e$. Adapting Eq. (11) for $|\Psi_{\mathrm{A},R}\rangle$ and rearranging it, the absolute values of the real parts are estimated as

$$|\tilde{a}_{e,k}| = \sqrt{\tilde{\mathrm{P}}_{\Psi_{\mathrm{A},R}} (|k\rangle)} ,$$

$$\left|\mathrm{Re}\left(\tilde{a}_k\right)\right| = R|\tilde{a}_{e,k}| .$$

Step 1.1I. By sampling circuit QFT-A with SP encoding $\boldsymbol{a}_o$, the absolute values of the imaginary parts are similarly estimated as

$$|\tilde{a}_{o,k}| = \sqrt{\tilde{\mathrm{P}}_{\Psi_{\mathrm{A},I}} (|k\rangle)} ,$$

$$\left|\mathrm{Im}\left(\tilde{a}_k\right)\right| = I|\tilde{a}_{o,k}| .$$

Step 1.2R. After that, circuit QFT-B is sampled with SP encoding $\boldsymbol{a}_e$ and the control measured in the Hadamard basis. Reworking Eq. (19) as

$$\tilde{a}_{e,k} = \sqrt{2^n} \left(\tilde{\mathrm{P}}_{\Psi_{\mathrm{B},R}} (|+\rangle \otimes |k\rangle) - \tilde{\mathrm{P}}_{\Psi_{\mathrm{B},R}} (|-\rangle \otimes |k\rangle) \right), \tag{21}$$

the signs of the real parts can be estimated as

$$\mathrm{sgn}\left(\mathrm{Re}\left(\tilde{a}_k\right)\right) = \mathrm{sgn}\left(\tilde{a}_{e,k}\right) .$$

Step 1.2I. Circuit QFT-B is finally sampled with SP encoding $\boldsymbol{a}_o$ and the control measured in the circular basis. Reworking Eq. (20) gives

$$\tilde{a}_{o,k} = \sqrt{2^n} \left(\tilde{\mathrm{P}}_{\Psi_{\mathrm{B},I}} (|+i\rangle \otimes |k\rangle) - \tilde{\mathrm{P}}_{\Psi_{\mathrm{B},I}} (|-i\rangle \otimes |k\rangle) \right), \tag{22}$$

from which the sign estimates of the imaginary parts are calculated,

$$\mathrm{sgn}\left(\mathrm{Im}\left(\tilde{a}_k\right)\right) = \mathrm{sgn}\left(\tilde{a}_{o,k}\right) .$$

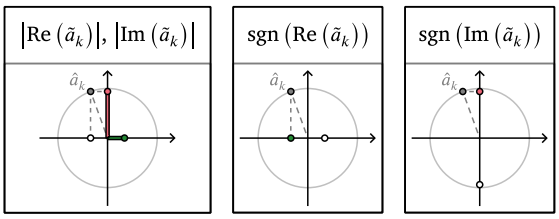


Fig. 7. Illustration of algorithm QFT2. (a) In step 2.1, estimates of the absolute values of the real and imaginary parts of the Fourier coefficients $\hat{a}_k$ of $\boldsymbol{a}$ are derived from estimates of their amplitudes and of the images of their phases in the first quadrant of the trigonometric circle. In steps (b) 2.2R and (c) 2.2I, estimates of the signs of the real and imaginary parts are derived from estimates of the projections of the coefficients onto the real and imaginary axes, respectively.

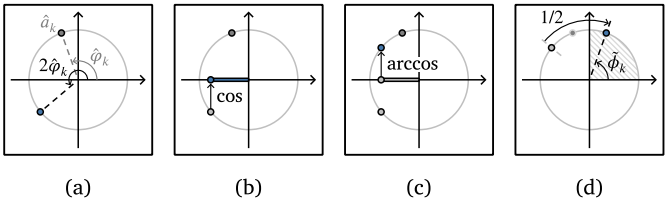


Fig. 8. Illustration of the 2-bit phase ambiguity in step 2.1. (a) Symmetry and post-processing considerations led to encoding $2\hat{\varphi}_k$ in relative phases of $|\Psi_{\mathrm{C}}\rangle$. (b) Sampling circuit QFT-C allows to estimate $\cos 2\hat{\varphi}_k$, from which information about $\hat{\varphi}_k$ can be recovered. (c) A first bit of ambiguity is introduced by the arccos function, which maps phases $2\hat{\varphi}_k \in [-\pi, \pi]$ to their absolute values $|2\hat{\varphi}_k| \in [0, \pi]$, i.e., maps phases in the third or fourth quadrant to their images in the first or second quadrant by symmetry about the real axis. (d) A second bit of ambiguity is then introduced by the $1/2$ factor which maps these angles to half their values, yielding the estimates $\tilde{\phi}_k \in [0, \pi/2]$ (indicated by the hatched area).

7.3. Algorithm QFT2

Illustrated in Fig. 7, the algorithm is made of two steps, the second of which is double. Step 2.1 estimates the absolute values of the real and imaginary parts of the coefficients together, then steps 2.2R and 2.2I estimate their signs.

Step 2.1. Circuit QFT-C is first sampled with the control measured in the Hadamard basis. Rearranging Eq. (15) gives

$$|\tilde{a}_k| = \sqrt{\tilde{\mathrm{P}}_{\Psi_{\mathrm{C}}} (|+\rangle \otimes |k\rangle) + \tilde{\mathrm{P}}_{\Psi_{\mathrm{C}}} (|-\rangle \otimes |k\rangle)} ,$$

$$\tilde{\phi}_k = \frac{1}{2} \arccos \left(\frac{\tilde{\mathrm{P}}_{\Psi_{\mathrm{C}}} (|+\rangle \otimes |k\rangle) - \tilde{\mathrm{P}}_{\Psi_{\mathrm{C}}} (|-\rangle \otimes |k\rangle)}{|\tilde{a}_k|^2} \right),$$

where $\tilde{\phi}_k$ is an estimate of the image of phase $\hat{\varphi}_k$ in the first quadrant of the trigonometric circle. This introduces a 2-bit phase ambiguity because arccos maps phases $2\hat{\varphi}_k \in [-\pi, \pi]$ to their absolute values in $[0, \pi]$ (first bit), which are then mapped to half their values by the $1/2$ factor (second bit), such that $\tilde{\phi}_k \in [0, \pi/2]$. This is illustrated in Fig. 8. As can be verified, the cosine and sine of $\tilde{\phi}_k$ correspond to the absolute values of the cosine and sine of $\hat{\varphi}_k$. Estimates of the absolute values of the real and imaginary parts can therefore be calculated as

$$\left|\mathrm{Re}\left(\tilde{a}_k\right)\right| = |\tilde{a}_k| \cos \tilde{\phi}_k \qquad \text{and} \qquad \left|\mathrm{Im}\left(\tilde{a}_k\right)\right| = |\tilde{a}_k| \sin \tilde{\phi}_k .$$

Step 2.2R. The signs of the real parts are then estimated by sampling circuit QFT-B with the control measured in the Hadamard basis, as in step 1.2R. The only difference is that the encoded signal is now complex. Rearranging Eq. (13), the signs are estimated as

$$|\hat{a}_k| \widetilde{\cos} \hat{\varphi}_k = \sqrt{2^n} \left(\tilde{\mathrm{P}}_{\Psi_{\mathrm{B}}} (|+\rangle \otimes |k\rangle) - \tilde{\mathrm{P}}_{\Psi_{\mathrm{B}}} (|-\rangle \otimes |k\rangle) \right), \tag{23}$$

$$\mathrm{sgn}\left(\mathrm{Re}\left(\tilde{a}_k\right)\right) = \mathrm{sgn}\left(|\hat{a}_k| \widetilde{\cos} \hat{\varphi}_k \right) .$$

Because of sampling uncertainty, the closer $|\hat{\varphi}_k|$ is to $\pi/2$, the greater the number of observations of $|k\rangle$ should be to estimate the kth sign confidently. Consequently, the greater the weight of the imaginary parts in the coefficients, the larger $n_{s,2\mathrm{R}}$ should be.

Step 2.2I. The signs of the imaginary parts are finally estimated by sampling circuit QFT-B with the control measured in the circular basis, as in step 1.2I. Rearranging Eq. (14), they are estimated as

$$\widetilde{|\hat{a}_k|\sin\hat{\varphi}_k} = \sqrt{2^n}\left(\tilde{\mathrm{P}}_{\Psi_\mathrm{B}}(|+i\rangle\otimes|k\rangle) - \tilde{\mathrm{P}}_{\Psi_\mathrm{B}}(|-i\rangle\otimes|k\rangle)\right), \tag{24}$$

$$\mathrm{sgn}\left(\mathrm{Im}\left(\tilde{a}_k\right)\right) = \mathrm{sgn}\left(\widetilde{|\hat{a}_k|\sin\hat{\varphi}_k}\right).$$

Similarly to step 2.2R, $n_{s,2\mathrm{I}}$ should be larger when the weight of the real parts in the coefficients is greater.

7.4. Comparison of algorithms QFT1 and QFT2

Note that algorithm QFT2 can be seen as a special case of QFT1, where the choice of using a sample to estimate the absolute value of the real or imaginary part of the kth coefficient in step 2.1 is random rather than deterministic. Indeed, state $|\Psi_\mathrm{C}\rangle$ can be reworked as

$$\begin{aligned}|\Psi_\mathrm{C}\rangle &= \sum_{k=0}^{N-1}|\hat{a}_k|\,\frac{e^{-i\hat{\varphi}_k}|0\rangle + e^{i\hat{\varphi}_k}|1\rangle}{\sqrt{2}}\otimes|k\rangle\\ &= \sum_{k=0}^{N-1}|\hat{a}_k|\left(\cos\hat{\varphi}_k\,|+\rangle - i\sin\hat{\varphi}_k\,|-\rangle\right)\otimes|k\rangle\\ &= R|+\rangle\otimes|\hat{\psi}_R\rangle - iI|-\rangle\otimes|\hat{\psi}_I\rangle.\end{aligned}$$

Measuring this state with the control measured in the Hadamard basis will thus project $|\Psi_\mathrm{C}\rangle$ onto $|\hat{\psi}_R\rangle$ with probability R^2 and onto $|\hat{\psi}_I\rangle$ with probability I^2. For a large number of samples $n_{s,1}$, the law of large numbers then says that $|\hat{\psi}_R\rangle$ and $|\hat{\psi}_I\rangle$ will be observed close to $R^2 n_{s,1}$ and $I^2 n_{s,1}$ times, respectively, which is similar to what happens in steps 1.1R and 1.1I when

$$n_{s,1\mathrm{R}} = R^2 n_{s,1} \qquad \text{and} \qquad n_{s,1\mathrm{I}} = I^2 n_{s,1}.$$

7.5. Algorithm QFT3

Illustrated in Fig. 9, the algorithm consists of two steps, the first of which is double. Steps 3.1C and 3.1S together estimate the amplitudes of the coefficients and their phases up to a constant π, then step 3.2 lifts this ambiguity by evaluating whether the ambiguous phases are in phase or in opposition with the true phases.

Step 3.1C. Circuit QFT-C is first sampled with the control measured in the Hadamard basis, as in step 2.1. Rearranging Eq. (15) gives

$$|\tilde{a}_{k,c}| = \sqrt{\tilde{\mathrm{P}}_{\Psi_\mathrm{C}}(|+\rangle\otimes|k\rangle) + \tilde{\mathrm{P}}_{\Psi_\mathrm{C}}(|-\rangle\otimes|k\rangle)},$$

$$\widetilde{\cos 2\hat{\varphi}_k} = \frac{\tilde{\mathrm{P}}_{\Psi_\mathrm{C}}(|+\rangle\otimes|k\rangle) - \tilde{\mathrm{P}}_{\Psi_\mathrm{C}}(|-\rangle\otimes|k\rangle)}{|\tilde{a}_{k,c}|^2}.$$

Step 3.1S. Circuit QFT-C is then sampled with the control measured in the circular basis. Rearranging Eq. (16) yields

$$|\tilde{a}_{k,s}| = \sqrt{\tilde{\mathrm{P}}_{\Psi_\mathrm{C}}(|+i\rangle\otimes|k\rangle) + \tilde{\mathrm{P}}_{\Psi_\mathrm{C}}(|-i\rangle\otimes|k\rangle)},$$

$$\widetilde{\sin 2\hat{\varphi}_k} = \frac{\tilde{\mathrm{P}}_{\Psi_\mathrm{C}}(|+i\rangle\otimes|k\rangle) - \tilde{\mathrm{P}}_{\Psi_\mathrm{C}}(|-i\rangle\otimes|k\rangle)}{|\tilde{a}_{k,s}|^2}.$$

Step 3.1 processing. After calculating the previous expressions, the amplitude estimates are refined as

$$|\tilde{a}_k| = \sqrt{\frac{n_{s,1\mathrm{C}}\,|\tilde{a}_{k,c}|^2 + n_{s,1\mathrm{S}}\,|\tilde{a}_{k,s}|^2}{n_{s,1\mathrm{C}} + n_{s,1\mathrm{S}}}},$$

while the phases are estimated with some ambiguity as

$$\tilde{\phi}_k = \frac{1}{2}\,\mathrm{atan2}\left(\widetilde{\sin 2\hat{\varphi}_k}, \widetilde{\cos 2\hat{\varphi}_k}\right).$$

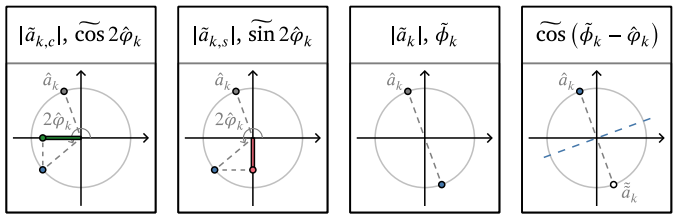


Fig. 9. Illustration of algorithm QFT3. Steps (a) 3.1C and (b) 3.1S estimate the amplitudes of the Fourier coefficients $\hat{a}_k$ of $\boldsymbol{a}$ and the cosines and sines of twice their phases. (c) These quantities are then processed together (step 3.1) to refine the amplitude estimates and derive estimates of the phases up to a constant π. (d) In step 3.2, the phase ambiguity is lifted by evaluating whether the ambiguous phases are in phase or in opposition with the true phases, which is deduced from estimates of the cosines of the relative angles between the ambiguous and true phases.

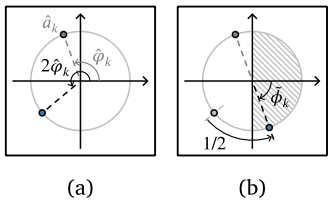


Fig. 10. Illustration of the 1-bit phase ambiguity in step 3.1. (a) Symmetry and post-processing considerations led to encoding $2\hat{\varphi}_k$ in relative phases of $|\Psi_\mathrm{C}\rangle$. Sampling circuit QFT-C with the control qubit measured in two different bases allows to estimate these double angles using the atan2 function, whose codomain is $[-\pi,\pi]$. (b) The ambiguity is introduced by the 1/2 factor which maps $2\hat{\varphi}_k \in [-\pi,\pi]$ to half their values, yielding the estimates $\tilde{\phi}_k \in [-\pi/2,\pi/2]$ (indicated by the hatched area).

These are either in phase or in opposition with the true phases, $\tilde{\phi}_k \approx \hat{\varphi}_k \bmod \pi$. As illustrated in Fig. 10, this 1-bit ambiguity appears because the 1/2 factor maps phases $2\hat{\varphi}_k \in [-\pi,\pi]$ (the codomain of atan2) to half their values, such that $\tilde{\phi}_k \in [-\pi/2,\pi/2]$. The above quantities then define the ambiguous coefficient estimate

$$\tilde{\tilde{a}}_k = |\tilde{a}_k|\,e^{i\tilde{\phi}_k}.$$

Note that in the absence of updated information on the cosines and sines, a good estimation of $\hat{\phi}_k$ can be obtained by setting $n_{s,1\mathrm{C}} = n_{s,1\mathrm{S}}$.

Step 3.2. Circuit QFT-D is finally sampled with $\widetilde{SP}$ encoding $\tilde{\tilde{a}}_k$ to evaluate whether the ambiguous phases $\tilde{\phi}_k$ are in phase or in opposition with the true phases $\hat{\varphi}_k$. Assuming that the error on $|\tilde{a}_k|$ is small, i.e., $|\tilde{a}_k| = |\hat{a}_k| + \epsilon$ with $\epsilon \to 0$, Eq. (17) becomes

$$\mathrm{P}_{\Psi_\mathrm{D}}(|\pm\rangle\otimes|k\rangle) = \frac{|\hat{a}_k|^2}{2}\left(1 \pm \cos\left(\tilde{\phi}_k - \hat{\varphi}_k\right)\right),$$

which can be rearranged to calculate

$$\sigma_k = \widetilde{\cos}\left(\tilde{\phi}_k - \hat{\varphi}_k\right) = \frac{\tilde{\mathrm{P}}_{\Psi_\mathrm{D}}(|+\rangle\otimes|k\rangle) - \tilde{\mathrm{P}}_{\Psi_\mathrm{D}}(|-\rangle\otimes|k\rangle)}{|\tilde{a}_k|^2}.$$

Since $\sigma_k \approx 1$ when $\tilde{\phi}_k \approx \hat{\varphi}_k$ and $\sigma_k \approx -1$ when $\tilde{\phi}_k \approx \hat{\varphi}_k + \pi$, the sign of σ_k indicates whether $\tilde{\phi}_k$ is likely in phase or in opposition with $\hat{\varphi}_k$. The phase ambiguity can therefore be lifted by

$$\bar{\varphi}_k = \tilde{\phi}_k + \begin{cases}\pi & \text{if } \mathrm{sgn}\left(\sigma_k\right) < 0,\\ 0 & \text{otherwise.}\end{cases}$$

The assumption $\epsilon \to 0$ holds for $n_{s,2} \to \infty$, so for large $n_{s,2}$ a few observations of $|k\rangle$ should lift the ambiguity with certainty.

7.6. Further comment

One might wonder why an algorithm based solely on circuit QFT-B was not designed. Indeed, by sampling QFT-B with the control qubit

measured in the Hadamard and circular bases, an estimate of the Fourier coefficients could be obtained in just two steps using Eqs. (23) and (24). However, this approach was discarded because of the inelegant post-processing it would have entailed. It was thought that the $\sqrt{2^n}$ factor would add noise to the signal, making the estimation of the coefficients less uniform and accurate than in the previous algorithms. This idea was therefore not explored.

8. Numerical configuration

The solver described in Section 4 was implemented in Python with NumPy and SciPy. The QFT algorithms were implemented in the VKI QCFD Framework, an in-house framework for running hybrid quantum-classical algorithms written in Python and based on the open-source Qiskit SDK [18] from IBM. The state preparation gates ran the recursive initialisation algorithm [23]. Quantum circuits were simulated on the AerSimulator backend from Qiskit Aer, parameterised as follows: density matrix simulations, noiseless or with the noise model described below, no target backend or connectivity constraint, no basis gate set (noiseless simulations) or the set given below (noisy simulations), and optimisation level 3. Other parameters were the defaults. The simulation environment used Python 3.11.8 with NumPy 1.26.4, SciPy 1.12.0, Qiskit 1.0.2 and Qiskit Aer 0.14.0. Simulations were run on a Dell PowerEdge R7525 server with two AMD EPYC 7H12 2.60 GHz 64-core processors and 128 GB of RAM.

Density matrix simulations are particularly expensive as they perform calculations based on the density matrix formalism presented in Section 5, and therefore scale as 2^{2n} ($2^n \times 2^n$ matrices). Nevertheless, they were chosen because they are able to simulate the evolution of both noisy and noiseless quantum systems and are representative of a real sampling process.

The basis gate set used for noisy simulations was the universal set $\{I, R_z, \sqrt{X}, X, CX\}$. The noise model applied the depolarising channel defined by Eq. (9) to all basis gates in the circuits with the same depolarising probability p_d. The latter will be referred to as the 'noise level' in the following. For reference, Bravyi et al. [24] stated that a physical error rate (defined as the probability that an operation on a quantum computer fails independently) of 0.1% seemed realistic in the short term. With effective error correction, the logical error rate could be reduced by several orders of magnitude. The backend, its parameterisation and the noise model were chosen to produce results as general as possible.[4]

9. Algorithmic behaviour

A theoretical study of the behaviour of the QFT algorithms under noiseless conditions is provided in Appendix A, the results of which are compared with a numerical study in Appendix B. The latter also investigates the impact of depolarising noise on the algorithms. The numerical study focused on the general case of computing the QFT of arbitrary complex signals and used the numerical configuration described in the previous section. This section summarises the main results obtained to characterise algorithmic behaviour.

The algorithms introduce errors with respect to the true coefficients calculated by the DFT. This error is composed of the errors introduced in each step of the algorithms, the magnitudes of which vary depending on the step. The most predictable errors are those introduced by the statistical uncertainty due to finite sampling, called 'sampling errors' in the following, which can be reduced by increasing the number of samples $n_{s,m}$ used in step m. Noise also introduces errors, but these are independent of $n_{s,m}$. In absolute terms, errors are introduced fairly uniformly among the coefficients, as the errors introduced in each step are independent of the coefficient amplitudes. The behaviour of the algorithms is independent of the input signal because the algorithms are invariant by rotation, that is, the phases of the coefficients do not influence the algorithmic behaviour.

The total sampling error introduced in the algorithms can be quantified by means of the L2 error ϵ_{L2}^2 defined by Eq. (10). While theoretical bounds allow one to estimate how the L2 errors scale with the number of samples, it was found that ϵ_{L2}^2 could be predicted fairly accurately by asymptotic fits of the errors introduced in each step. Recall that a total of n_s samples are used in the algorithms. The results recommend that they be distributed between the steps as

$$n_{s,1\mathrm{R}} = n_{s,1\mathrm{I}} = \frac{\lambda}{2} n_s \quad \text{and} \quad n_{s,2\mathrm{R}} = n_{s,2\mathrm{I}} = \frac{(1-\lambda)}{2} n_s \quad \text{for QFT1,} \tag{25a}$$

$$n_{s,1} = \lambda\, n_s \quad \text{and} \quad n_{s,2\mathrm{R}} = n_{s,2\mathrm{I}} = \frac{(1-\lambda)}{2} n_s \quad \text{for QFT2,} \tag{25b}$$

$$n_{s,1\mathrm{S}} = n_{s,1\mathrm{C}} = \frac{\lambda}{2} n_s \quad \text{and} \quad n_{s,2} = (1-\lambda)\, n_s \quad \text{for QFT3,} \tag{25c}$$

where $\lambda \in (0, 1)$ is a distribution parameter. With these distributions, the L2 errors are predicted to scale with n_s/N as

$$\epsilon_{\mathrm{L2}}^2 \approx \frac{0.526}{\lambda} \frac{N}{n_s} + \frac{0.817}{(1-\lambda)^{3/2}} \left(\frac{N}{n_s}\right)^{\frac{3}{2}} \quad \text{for QFT1,} \tag{26a}$$

$$\epsilon_{\mathrm{L2}}^2 \approx \frac{0.566}{\lambda} \frac{N}{n_s} + \frac{4.389}{(1-\lambda)^{3/2}} \left(\frac{N}{n_s}\right)^{\frac{3}{2}} \quad \text{for QFT2,} \tag{26b}$$

$$\epsilon_{\mathrm{L2}}^2 \approx \frac{0.739}{\lambda} \frac{N}{n_s} + \frac{2.720}{(1-\lambda)^{2}} \left(\frac{N}{n_s}\right)^{2} \quad \text{for QFT3.} \tag{26c}$$

However, caution is advised when extrapolating these results due to the limited range of n_s/N studied and the intrinsic uncertainty in the errors for which the fits were established.

For greater clarity in what follows, steps 1.1R, 1.1I, 2.1, 3.1C and 3.1S will be referred to as 'amplitude-estimating steps', while steps 1.2R, 1.2I, 2.2R, 2.2I and 3.2 will be called 'ambiguity-lifting steps'.

The first terms on the right-hand sides of the above equations are the errors introduced in the amplitude-estimating steps, and the second terms are those introduced in the ambiguity-lifting ones. The errors of the amplitude-estimating steps scale worse with n_s/N than the others. They are therefore dominant for most numbers of samples, except at very small values for which the errors of the ambiguity-lifting steps are of comparable magnitude. As a result, the L2 errors scale linearly with n_s/N for large values and exhibit a non-linear portion for small values. Algorithm QFT1 shows the lowest constants, followed by QFT2 and QFT3, although all algorithms perform similarly. Note that the main advantage of QFT1 is that half of its steps can be skipped when the Fourier transform is known to be purely real or imaginary, which roughly halves the number of samples needed to reach a given error.

The optimal sample distributions between the steps were estimated for each algorithm by differentiating the above equations with respect to λ and setting the derived equations equal to zero. The estimated optimal values of λ and $(1-\lambda)$ are plotted in Fig. B.20 as functions of the number of samples. Parameter λ increases rapidly with n_s/N, suggesting that for larger numbers of samples, a larger share of samples should be allocated to the amplitude-estimating steps. This is expected considering that the errors introduced in these steps are dominant. However, ϵ_{L2}^2 can at best be reduced by one order of magnitude if a poor value of λ is replaced by the optimum, resulting in a $\sqrt{10}$ reduction in ϵ_{L2}. The improvement over a naive value would likely be much smaller. Say one decides to distribute the samples evenly between the steps of the algorithms, knowing that step 2.1 should be compared with steps 1.1R and 1.1I, such that it makes sense to have $n_{s,1} = n_{s,1\mathrm{R}} + n_{s,1\mathrm{I}}$. This corresponds to setting $\lambda = 0.5$ for QFT1 and QFT2, and $\lambda = 2/3$ for

[4] As for computing time, preparation time (classical pre-processing, circuit transpilation, etc.) was negligible, so simulation time was proportional to the size of the matrices and the number of samples used. It ranged from under 1 s per QFT for $n = 2$ and $n_s = 10^2$ to nearly 5 min for $n = 7$ and $n_s = 10^8$. Noisy simulations took about twice as long because twice as many matrices were involved in the calculations (one extra matrix per gate for the noise channel).

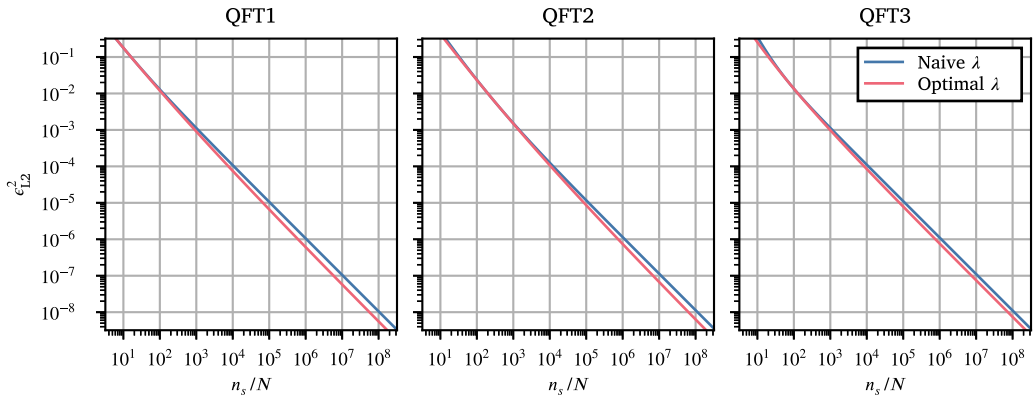


Fig. 11. Predicted L2 errors of the algorithms. The errors ϵ^2_{L2} defined by Eq. (26) are plotted as functions of n_s/N for naive and optimal sample distributions set by parameter λ, with n_s the total number of samples and N the signal length. In the naive case, $\lambda = 0.5$ for QFT1 and QFT2, and $\lambda = 2/3$ for QFT3. The optimal values of λ are shown in Fig. B.20.

QFT3. The predictions of the L2 errors in this case are plotted in Fig. 11 for a wide range of n_s/N, and are compared with the predictions for the optimal values of λ. The best improvement obtained by the optimum is a reduction of ϵ^2_{L2} by a factor 1.927, for QFT1 at large n_s/N.

The algorithms are very sensitive to noise because they use state preparation and QFT circuits made of many gates, each of which is subject to errors. Their behaviour in noisy conditions is shown in Fig. B.21. For small n_s/N and low noise levels, errors are initially dominated by statistical uncertainty. Increasing the number of samples reduces ϵ^2_{L2} until the errors due to noise become comparable in magnitude to the sampling errors. At this point, the L2 errors deviate from their noiseless scalings, eventually becoming independent of n_s/N as noise becomes dominant. The deviation occurs earlier for higher noise levels. Performance also degrades for longer signals because the number of gates in the state preparation and QFT circuits is proportional to the number of working qubits. The amplitude-estimating steps proved to be more sensitive to noise than the ambiguity-lifting ones, for all algorithms. Overall, algorithm QFT1 is the most robust to depolarising noise, followed by QFT3 and QFT2. Unfortunately, increasing the number of samples to reduce the error only works up to a point, as noise inevitably becomes dominant.

10. Hybrid solver study

Now that the algorithms have been thoroughly described and analysed, the focus can return to the hybrid solver. Its behaviour and the quality of its solutions were studied for a baseline case before the impact of several parameters was examined.

Recall that the solver implemented Eq. (8) with the boundary conditions defined by Eq. (1). In the baseline case, the Reynolds number was set to $Re = 10^3$, the domain length to $L = 1$, the fundamental time period to $T = 1$, and the number of harmonics and time levels to $N_h = 7$ and $N_t = 16$. The latter was rounded up from 15 to fully exploit the superposition of $n = 4$ working qubits used in the QFT algorithm. The spatial domain was divided into $N_x = 100$ cells, the initial condition was set to $\boldsymbol{U}^0 = 1/2$ and the CFL number was set to 10. The CFL number was defined as $CFL = u_{\max}\Delta\tau/\Delta x$, where $u_{\max} = \max_t(u(0,t)) = 1$ was the maximum speed at the left boundary.

In hybrid mode, the DFTs in the spectral time derivative operator $\mathscr{D}$, defined by Eq. (4), were estimated using algorithm QFT1. Those in the Jacobian, Eq. (7), were calculated by the FFT. The solver therefore calculated 100 forward and 100 inverse QFTs at each iteration. The total number of samples ranged from $n_s = 10^3$ to 10^8 and the backend was noiseless ($p_d = 0\%$). The samples were distributed evenly between the steps, as in Eq. (25a) with the naive value $\lambda = 0.5$. The L2 errors ϵ^2_{L2} predicted for this distribution are shown in Fig. 11. Note that the relevant metric in this section is ϵ_{L2}, not ϵ^2_{L2}. Given the limited error reduction achievable with the optimal λ, this sample distribution was chosen to ensure consistency between results at different n_s. QFT1 was selected because the inverse Fourier transform in operator $\mathscr{D}$ is known to return purely real coefficients (the conserved variable u is real), and because it is more robust to noise. The calculation of the imaginary parts was therefore skipped in the inverse QFTs, for which a total of $n_s/2$ samples were distributed between steps 1.1R and 1.2R. This approach allowed to introduce the same errors in both directions of the Fourier transforms while reducing computation time.

Several parameters were modified from the baseline, one at a time. The first two affect the quantum algorithm, the next two the solver pseudo-time marching, and the last two the physics of the problem. Firstly, the levelled-out residuals were compared for the different algorithms. The sample distributions defined in Eq. (25) were used with naive values $\lambda = 0.5$ for QFT2 and $\lambda = 2/3$ for QFT3. The effect of noise on the solution was then investigated for noise levels of $p_d = 0.001\%$, 0.01% and 0.1%. Next, the impact of the CFL number was studied, as it affects the stability of the solver. It was changed to $CFL = 1$ and 100. After this, the number of harmonics was varied to $N_h = 3$ and 15 to evaluate its influence on the propagation of sampling errors in the solver. Finally, the Reynolds number was set to $Re = 10^2$ (more viscous) and 10^4 (less viscous) to assess in which flow regime the hybrid solver best performs.

10.1. Baseline case

The hybrid solver behaviour is best investigated by first comparing its residuals for the baseline case with those of the classical solution. Fig. 12 shows the evolution of the RMS and maximum residuals, defined at iteration k as

$$\Delta\boldsymbol{U}^k_{\text{rms}} = \sqrt{\frac{1}{N_t N_x}\sum_{t,i}\left(\Delta u^k_{t,i}\right)^2} \qquad \text{and} \qquad \Delta\boldsymbol{U}^k_{\max} = \max_{t,i}\left(\Delta u^k_{t,i}\right),$$

where $\Delta u^k_{t,i}$ is the (t,i)th element of $\Delta\boldsymbol{U}^k$ in Eq. (8). While the residuals of the classical solution converge to machine accuracy, those of the hybrid ones converge up to a certain point, then level out.

Recall that marching the system of equations towards its steady-state solution amounts to reducing the total error in the current solution by removing the perturbations on the exact solution of the discrete equations. Long-wavelength perturbations are mostly advected out of the computational domain through its boundaries (when they allow it), while short-wavelength ones are dissipated mainly by physical and numerical diffusion. Short-wavelength perturbations generally decay more quickly than long-wavelength ones. In a classical setting, residuals level out when the remaining long-wavelength perturbations cannot leave the domain and bounce back between the boundaries.

But in a hybrid solver, quantum algorithms are a constant source of perturbations.

At every iteration, new perturbations ('quantum errors') are introduced in operator $\mathscr{D}$. Both sampling errors and depolarising noise (which is white noise) have a flat spectrum and are therefore introduced at all wavelengths. Residuals decrease normally at first, then level out when the error in the current solution becomes dominated by quantum errors. At this point, the solver removes perturbations that are reintroduced right away. Two factors therefore come into play. On the one hand, the quantum computing backend, the quantum algorithm and its parameterisation fix the amount of errors introduced. On the other hand, the discretised equations and the solver parameters determine the rate at which errors decrease. Levelled-out residuals are set by the balance between these two factors.

To better understand this process, it is interesting to examine where quantum errors are introduced and how they evolve within the spectral operator. The remainder of the discussion is limited to sampling errors. Recall that applying $\mathscr{D}$ to $\boldsymbol{u}_i^*$, the vector of the current solution at the N_t time levels at cell i, consists in calculating

$$\underbrace{\mathscr{D}\,\boldsymbol{u}_i^*}_{(d)} = \underbrace{\mathscr{F}^{-1}}_{(c)}\,\overbrace{\underbrace{\Omega}_{(a)}\,\underbrace{\mathscr{F}\,\boldsymbol{u}_i^*}_{(a)}}^{(b)}. \tag{27}$$

Errors are first introduced by the forward QFT (a) and amplified by matrix Ω (b). They are then fed to the inverse QFT, which flattens the spectrum of the errors present in its input signal and attenuates them. It also introduces additional errors itself (c). The total error introduced in operator $\mathscr{D}$ (d) therefore conceals four contributions, whose magnitude can be investigated in more details.

To do so, the following error metrics are defined:

$$\Delta^k_{h,i}\mathscr{F} = \left(\mathscr{F}_{QFT}\,\boldsymbol{u}_i^{*\,k} - \mathscr{F}_{FFT}\,\boldsymbol{u}_i^{*\,k}\right)_h, \tag{28a}$$

$$\Delta^k_{h,i}\Omega\mathscr{F} = \left(\Omega\,\mathscr{F}_{QFT}\,\boldsymbol{u}_i^{*\,k} - \Omega\,\mathscr{F}_{FFT}\,\boldsymbol{u}_i^{*\,k}\right)_h, \tag{28b}$$

$$\Delta^k_{t,i}\mathscr{F}^{-1} = \left(\mathscr{F}^{-1}_{QFT}\,\Omega\,\mathscr{F}_{FFT}\,\boldsymbol{u}_i^{*\,k} - \mathscr{F}^{-1}_{FFT}\,\Omega\,\mathscr{F}_{FFT}\,\boldsymbol{u}_i^{*\,k}\right)_t, \tag{28c}$$

$$\Delta^k_{t,i}\mathscr{D} = \left(\mathscr{F}^{-1}_{QFT}\,\Omega\,\mathscr{F}_{QFT}\,\boldsymbol{u}_i^{*\,k} - \mathscr{F}^{-1}_{FFT}\,\Omega\,\mathscr{F}_{FFT}\,\boldsymbol{u}_i^{*\,k}\right)_t, \tag{28d}$$

where the subscripts of $\mathscr{F}$ specify how the Fourier coefficients were calculated, and where subscripts h and t indicate the harmonic (in the frequency domain) and time level (in the time domain), respectively. Fig. 13 shows the evolution of these metrics for the baseline case with $n_s = 10^6$ samples for a selection of harmonics and time levels. The metrics were spatially averaged by taking their mean values over all the cells, for instance,

$$\Delta\mathscr{F} = \frac{1}{N_x}\sum_i \left|\Delta^k_{h,i}\mathscr{F}\right|. \tag{29}$$

With these definitions, consider what happens at iteration k.

(a) $\Delta\mathscr{F}$. The forward QFT first introduces sampling errors rather uniformly across all harmonics, as expected from its flat spectrum.[5] The average error introduced in the system can be calculated by taking the mean over all the cells and iterations of the RMS over all the harmonics of the error $\Delta^k_{h,i}\mathscr{F}$, i.e.,

$$\Delta_{\text{avg}}\mathscr{F} = \frac{1}{N_k N_x}\sum_{k,i}\sqrt{\frac{1}{N_t}\sum_h \left|\Delta^k_{h,i}\mathscr{F}\right|^2}. \tag{30}$$

In this case, $\Delta_{\text{avg}}\mathscr{F} = 2.386\cdot 10^{-3}$. Normalising it by the average of the solution $\boldsymbol{U}$ calculated in the same way gives $\epsilon_a = 3.946\cdot 10^{-3}$, which is close to the error $\epsilon_{\text{L2}} = 4.121\cdot 10^{-3}$ predicted by Eq. (26a).

(b) $\Delta\Omega\mathscr{F}$. The Fourier coefficients are then multiplied by matrix Ω, which amplifies the errors at harmonic h by a factor $i\omega h$. This results

[5] The error introduced in harmonic zero was actually about 5 times smaller, which has not yet been explained. Unpublished investigation suggests that this is a feature of algorithms QFT1 and QFT2, but not QFT3.

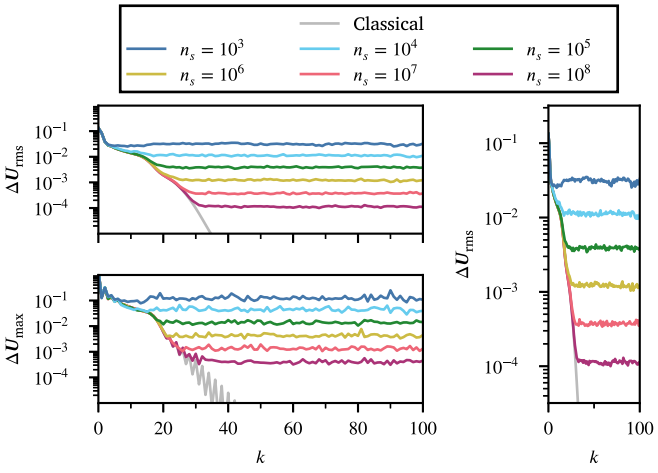


Fig. 12. Residuals of the hybrid and classical solvers for the baseline case. The RMS and maximum residuals ΔU_{rms} and ΔU_{max} are plotted as functions of the iteration k on the left-hand plots, with the RMS residuals rescaled on the right-hand plot. Classical residuals decrease to machine accuracy in about 100 iterations. The mean levelled-out RMS residuals are reported in Table 1.

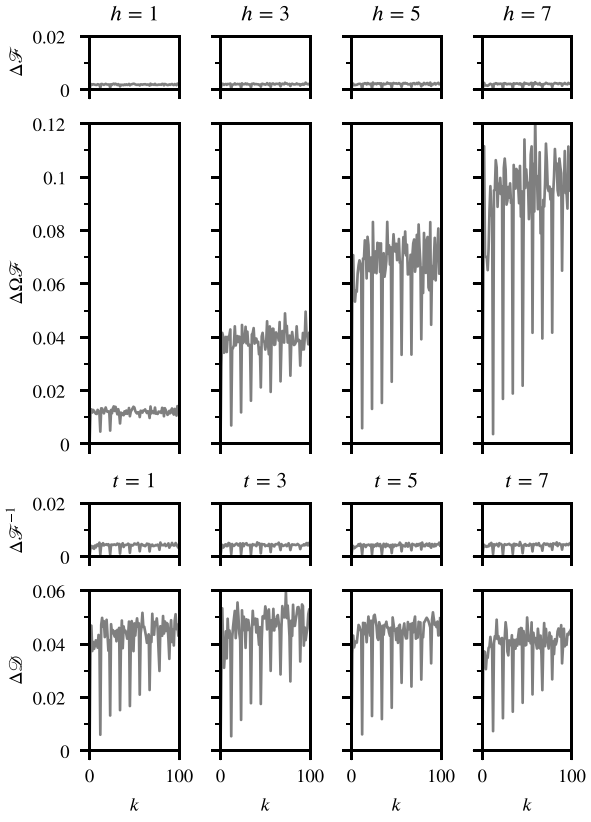


Fig. 13. Sampling errors introduced in the spectral time derivative operator $\mathscr{D}$ for the baseline case with a total of $n_s = 10^6$ samples per QFT. A selection of harmonics h (in the frequency domain) and time levels t (in the time domain) is shown. The errors are defined by Eq. (28) and averaged as in Eq. (29). (a) $\Delta\mathscr{F}$. At each iteration k, sampling errors are introduced uniformly on all Fourier coefficients by the forward QFT. (b) $\Delta\Omega\mathscr{F}$. As the coefficients are multiplied by matrix Ω, the error in each harmonic is amplified by $i\omega h$. (c) $\Delta\mathscr{F}^{-1}$. The inverse QFT introduces additional errors uniformly. (d) $\Delta\mathscr{D}$. The total error introduced in operator $\mathscr{D}$ is composed of the $\Delta\Omega\mathscr{F}$ errors fed to the inverse QFT, whose spectrum is flattened and which are attenuated by $\mathscr{F}^{-1}$, and of the additional errors introduced by $\mathscr{F}^{-1}$. The dips occurring every 10 iterations could be explained by signal processing.

Table 1
Mean levelled-out RMS residuals $\Delta U_{\mathrm{rms},\infty}$ for the baseline case for the different total numbers of samples n_s used per QFT.

n_s	10^3	10^4	10^5	10^6	10^7	10^8
$\Delta U_{\mathrm{rms},\infty}$	$3.078 \cdot 10^{-2}$	$1.119 \cdot 10^{-2}$	$3.918 \cdot 10^{-3}$	$1.212 \cdot 10^{-3}$	$3.749 \cdot 10^{-4}$	$1.127 \cdot 10^{-4}$

in larger values of $\Delta\Omega\mathscr{F}$ at higher frequencies. The average error is calculated similarly to Eq. (30), giving $\Delta_{\mathrm{avg}}\Omega\mathscr{F} = 71.328 \cdot 10^{-3}$.

(c) $\Delta\mathscr{F}^{-1}$. The inverse QFT then introduces additional errors uniformly, on average $\Delta_{\mathrm{avg}}\mathscr{F}^{-1} = 5.138 \cdot 10^{-3}$. Normalising this value by the average of $\Omega\mathscr{F}\boldsymbol{U}$ calculated in the same manner yields $\epsilon_c = 3.817 \cdot 10^{-3} \sim \epsilon_{\mathrm{L2}}$.

(d) $\Delta\mathscr{D}$. The total error introduced by operator $\mathscr{D}$ is composed of the error already present in the input signal of the inverse QFT (b) and of the error introduced by the inverse QFT itself (c). The average was $\Delta_{\mathrm{avg}}\mathscr{D} = 47.764 \cdot 10^{-3}$. The total error seems clearly dominated by the $\Delta\Omega\mathscr{F}$ errors, although the inverse QFT appears to attenuate these, in addition to flattening their spectrum.

Consequently, due to the amplification by matrix Ω, it may be more sensible to improve the accuracy of the forward QFT than that of the inverse. This suggests that, for a total number of samples n_s to be distributed between the forward and inverse QFTs, most samples should be allocated to the forward one.

Return now to the residuals in Fig. 12. As the number of samples used in the quantum algorithm increases, the magnitude of the sampling errors decreases accordingly, as does the value at which residuals level out. Small variations are observed because sampling errors are probabilistic. The mean levelled-out RMS residuals are reported in Table 1. As expected from the inverse linear asymptotic scaling of ϵ_{L2}^2 with n_s/N, increasing the number of samples by one order of magnitude results in a reduction of the sampling errors, and hence of the levelled-out residuals, by a factor of about $\sqrt{10}$ for $n_s \geq 10^5$. The reduction is slightly less for $n_s < 10^5$.

The reconstructed hybrid solutions at iteration $k = 100$ for the different n_s values are plotted as spacetime diagrams in Fig. 14(a) and are compared with the classical solution. Fig. 14(b) offers an alternative visualisation in the form of snapshots of the solutions at different times. The hybrid solutions match the overall trend of the classical solution, with high-frequency wobbles around it caused by the sampling errors. As expected, increasing n_s greatly improves the solution and reduces the wobbles at first, but the improvement becomes gradually less noticeable. It gets clearer in Fig. 14(c), which shows logarithmic spacetime diagrams of the error $\Delta u = u_H - u_C$ between the hybrid (H) and classical (C) solutions. The hybrid solutions oscillate at high frequency between overshooting the classical one (in red) and undershooting it (in blue), causing the wobbles in Fig. 14(b). Although attenuated in magnitude for larger n_s, this behaviour is consistent for all hybrid solutions. Nevertheless, these results indicate that hybrid solutions can provide good approximations to the classical solution, depending on the desired level of accuracy.

10.2. Parameter variations

Several parameters were changed from the baseline case in order to evaluate their impact on solution quality. The latter is assessed by means of the RMS residuals and always compared with the baseline.

First, the hybrid solver was run using algorithms QFT2 and QFT3. The residuals obtained are compared with those of the baseline case in Fig. 16(a). QFT1 and QFT2 perform slightly better than QFT3 for small n_s, but all algorithms reach similar levelled-out residuals for $n_s \geq 10^6$, with a minimal advantage for QFT2.

After that, the solver was run under noisy conditions with the backend implementing noise levels of $p_d = 0.001\%$, 0.01% and 0.1%. As Fig. 16(b) shows, noise ruins the solutions except for $p_d = 0.001\%$. Focusing on this case, the error is dominated by sampling errors for small numbers of samples, but as n_s increases, noise eventually becomes the dominant source. It takes over somewhere between 10^4 and 10^5 samples, explaining why the residuals coalesce from $n_s = 10^5$. Just how much noise affects the solution can be appreciated in Fig. 15, where spacetime diagrams and snapshots of the solutions are plotted alongside spacetime diagrams of the errors. While no improvement can be expected when noise dominates the error, satisfactory approximations to the classical solution can still be obtained, depending on the required accuracy. The solutions become progressively worse for $p_d > 0.001\%$, but some still capture the general trend.

Next, the solver was run with CFL numbers of $CFL = 1$ and 100, which had a significant impact on the residuals shown in Fig. 16(c). Decreasing and increasing the CFL number improves and worsens them, respectively. This could be explained by the fact that the solver is less stable at higher CFL (it is unstable above 265 in the classical setting).

The number of harmonics was then changed to $N_h = 3$ and 15 ($N_t = 8$ and 32 time levels). Fig. 16(d) shows that the residuals level out at larger values for higher numbers of harmonics. This is because the amplification caused by Ω in Eq. (27) is proportional to N_h, such that when the number of harmonics is doubled, $\Delta_{\mathrm{avg}}\Omega\mathscr{F}$ roughly doubles. As it dominates $\Delta\mathscr{D}$, the residuals level out at about twice their original value. This highlights a trade-off between the fidelity of the approximation to the unsteady solution (the number of harmonics retained) and the quality of the hybrid numerical solution (the residuals).

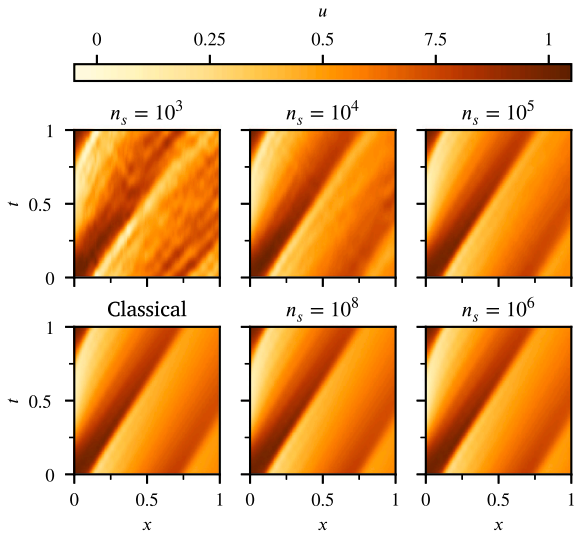
The problem was finally changed to be more and less viscous by setting the Reynolds number to $Re = 10^2$ and 10^4, yielding the residuals plotted in Fig. 16(e). Residuals decrease as Re decreases, because a higher physical viscosity dissipates the short-wavelengths errors more. Sampling errors therefore begin to dominate at lower residuals for more viscous cases. The baseline solution is close to that of the less viscous case, which is regarded as quasi-inviscid.

11. Conclusion

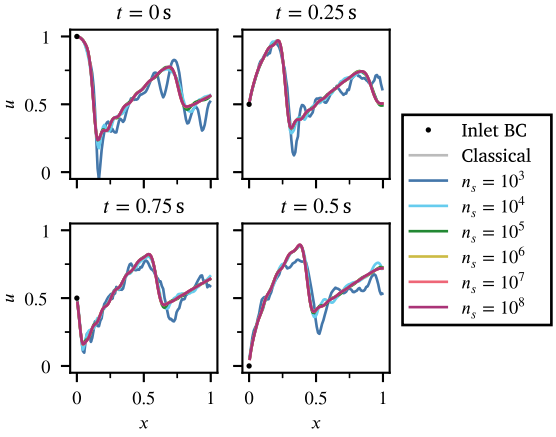
This manuscript detailed the design and study of quantum algorithms based on the quantum Fourier transform, as well as their implementation in a harmonic balance solver for Burgers' equation with time-periodic boundary conditions.

Given that the HB solver requires complete knowledge of the Fourier coefficients, which the standard QFT cannot provide directly, three novel QFT algorithms were developed. They first estimate the amplitudes of the coefficients and their projections onto the axes of the trigonometric circle (amplitude-estimating steps). The projections arise from the measurement process and the way in which relative phases are encoded between basis states, thereby introducing phase ambiguities. Relative phases between quantum states are needed to recover information about the phases of the Fourier coefficients. The algorithms then resolve these ambiguities (ambiguity-lifting steps).

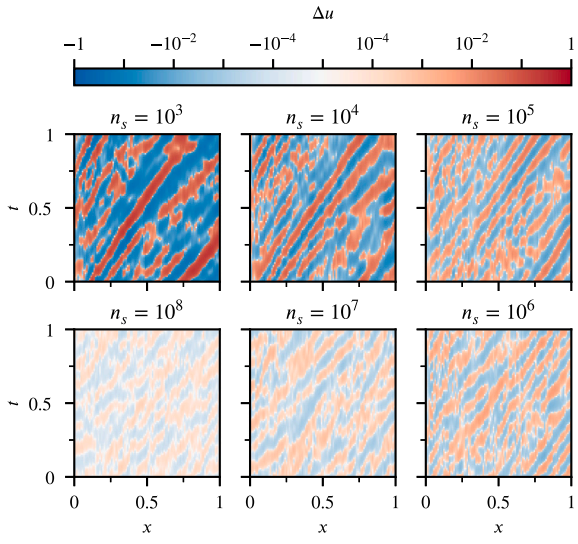
A theoretical study derived asymptotic bounds on the L2 errors of the algorithms under noiseless conditions. These errors are due to the statistical uncertainty associated with finite sampling. The bounds scale inversely linearly with the number of samples (more precisely, with n_s/N), and suggest that algorithm QFT3 should perform best, followed by QFT1 and QFT2. However, QFT1 performed slightly better in a numerical study where samples were evenly distributed between the steps. Note that QFT1 has the advantage that the weights given to estimating the real and imaginary parts of the Fourier coefficients can be adjusted, so that one part can be skipped when the output signal is known to be purely real or imaginary. Asymptotic fits of the errors introduced in each step of the algorithms were calculated, suggesting that all amplitude-estimating steps scale as $\mathscr{O}(N/n_{s,m})$ while ambiguity-lifting steps scale as $\mathscr{O}((N/n_{s,m})^{\alpha})$, with $\alpha = 3/2$ for QFT1 and QFT2, and $\alpha = 2$ for QFT3. The L2 error is therefore dominated by errors in the amplitude-estimating steps, except for very small numbers of samples. It was found that the fits of the errors introduced in each step predicted



(a) Spacetime diagrams of the solutions.

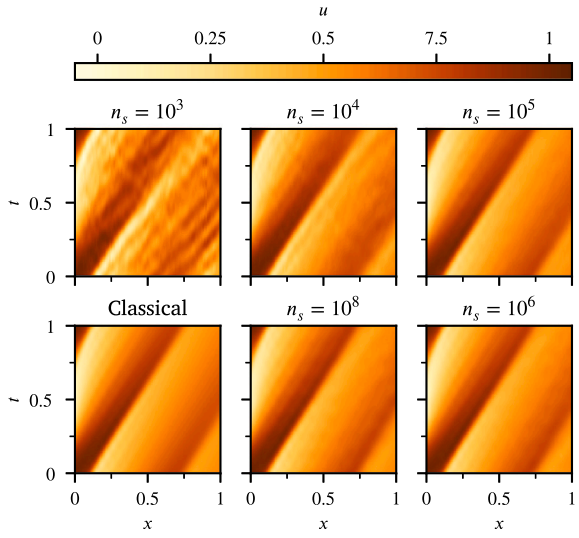


(b) Solution snapshots. Hybrid solutions for $n_s \geq 10^5$ are stacked onto the classical one.

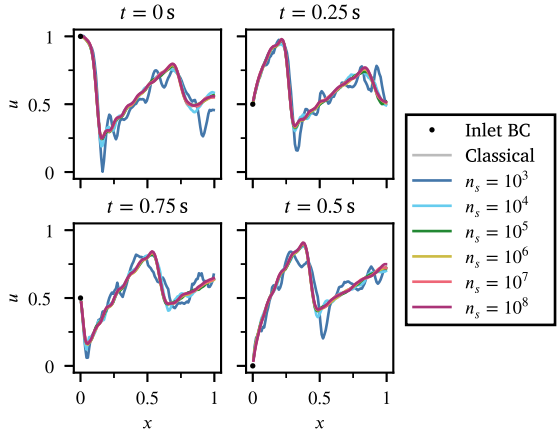


(c) Spacetime diagrams of the errors in the hybrid solutions. Colour scale is logarithmic.

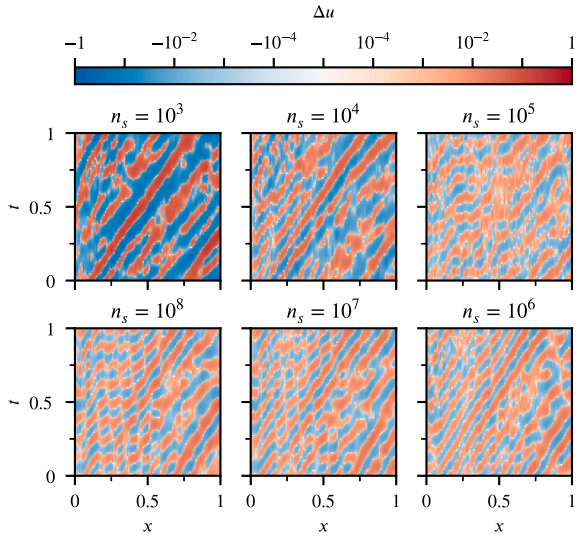
Fig. 14. Visualisations of the hybrid (H) and classical (C) solutions $u(x,t)$ for the baseline case and of the error $\Delta u = u_H - u_C$ between them. Hybrid solutions are plotted for different numbers of samples n_s used per QFT.



(a) Spacetime diagrams of the solutions.

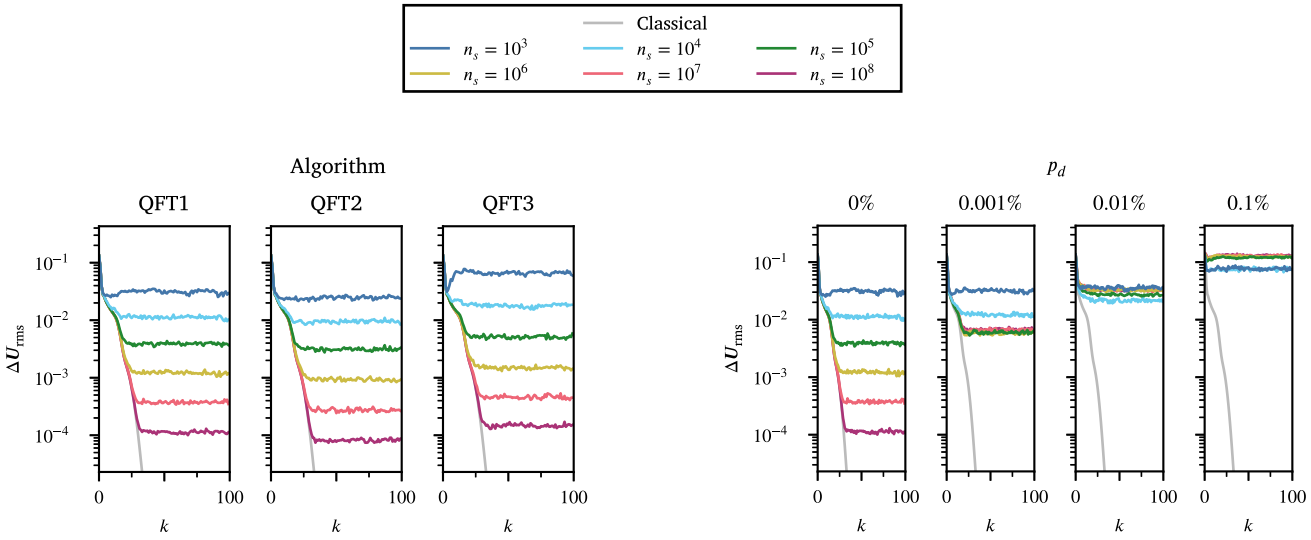


(b) Solution snapshots. Hybrid solutions for $n_s \geq 10^5$ are stacked onto the classical one.



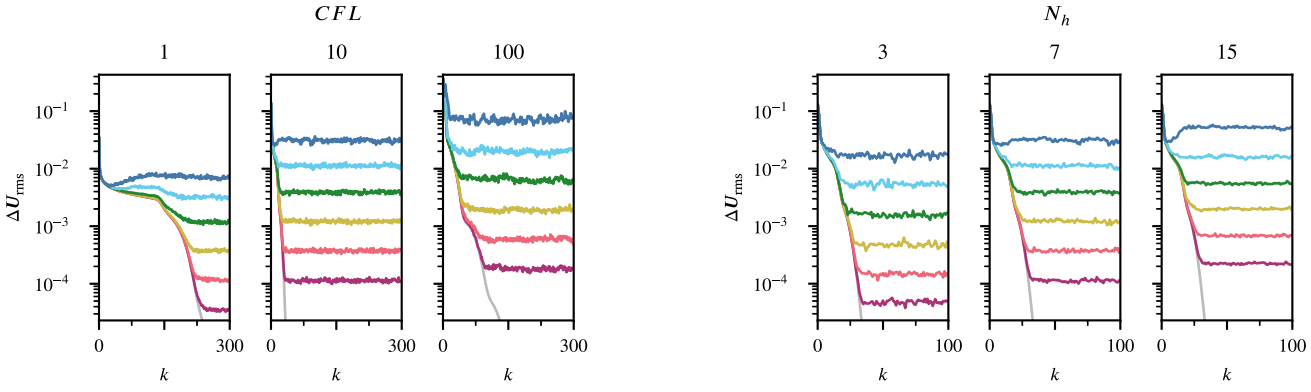
(c) Spacetime diagrams of the errors in the hybrid solutions. Colour scale is logarithmic.

Fig. 15. Visualisations of the hybrid (H) and classical (C) solutions $u(x,t)$ for the noisy case with $p_d = 0.001\%$, and of the error $\Delta u = u_H - u_C$ between them. Hybrid solutions are plotted for different numbers of samples n_s used per QFT. No improvement is observed for $n_s \geq 10^5$.



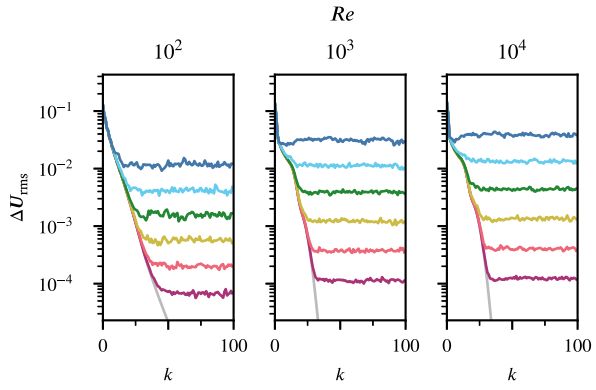
(a) QFT algorithm changed from QFT1 to QFT2 and QFT3.

(b) Noise level changed from $p_d = 0\%$ to 0.001%, 0.01% and 0.1%.



(c) CFL number changed from $CFL = 10$ to 1 and 100.

(d) Number of harmonics changed from $N_h = 7$ to 3 and 15.



(e) Reynolds number changed from $Re = 10^3$ to 10^2 and 10^4.

Fig. 16. Comparison of the RMS residuals ΔU_{rms} obtained by changing solver parameters from the baseline case, for different numbers of samples n_s used per QFT. The legend is at the top of the page.

the L2 errors fairly well in the range of n_s considered. Optimal sample distributions were also estimated.

All algorithms are highly sensitive to depolarising noise, with performance deteriorating for longer signals. This can be explained by the large number of gates present in the circuits. As the number of samples increases, the L2 errors first decrease like their noiseless scalings, before deviating from them and becoming independent of n_s when noise dominates the quantum error. The amplitude-estimating steps are the most sensitive to noise. Overall, QFT1 is the most robust, followed by QFT3 and QFT2. The algorithms can tolerate 0.001% of noise, but their performance degrades rapidly as the level increases.

Next, the behaviour of the hybrid solver was studied for a baseline case. QFTs were calculated using the QFT1 algorithm in a noiseless setting with samples evenly distributed between its steps. Numbers of samples ranging from $n_s = 10^3$ to 10^8 were considered. The RMS residuals of the hybrid solutions initially decrease like those of the classical solution, then level out. This occurs at $\Delta \boldsymbol{U}_{\mathrm{rms}}$ values of around $3 \cdot 10^{-2}$ for $n_s = 10^3$ and $1 \cdot 10^{-4}$ for $n_s = 10^8$. It can be explained by the fact that the error in the current solution is first dominated by perturbations present in the initial solution, but as the quantum algorithm introduces constant errors at each iteration, the solution quickly becomes dominated by sampling errors. At this point, the solver removes perturbations that are reintroduced right away. Because they are amplified, the errors introduced by the forward QFTs seem to have more impact on the total quantum error than those introduced by the inverse QFTs. As expected from the L2 error scaling, multiplying the number of samples by 10 roughly reduces the levelled-out residuals by a factor of $\sqrt{10}$. Nevertheless, the hybrid solver can produce good approximations to the classical solution, depending on the desired level of accuracy.

Finally, the impact of several solver parameters on the hybrid solution was evaluated. Of particular interest is the effect of depolarising noise, which rapidly degrades the solution. At low noise levels and for small numbers of samples, the quantum error remains dominated by sampling errors. Residuals can be further improved by increasing n_s, but noise inevitably begins to dominate the error at some point. With 0.001% of noise, this happens at $n_s = 10^5$. Beyond this point, the residuals cannot be reduced any further. However, decent approximations to the classical solution are still achievable with very low noise levels when low accuracy is sufficient for the application.

This work developed skills and understanding in the design, study and implementation of quantum algorithms for CFD applications, thereby highlighting opportunities and challenges presented by quantum computing. The results provide an insight into the performance of a hybrid CFD solver and how various parameters affect its solution. In particular, the hybrid solver can produce satisfactory approximations to the classical solution for a large but realistic number of samples per QFT, even with some noise, which is encouraging for the prospects of using quantum algorithms in CFD.

CRediT authorship contribution statement

Loïc Dewitte: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Jérémie Roland:** Conceptualization, Methodology, Validation, Formal analysis, Writing – original draft, Writing – review & editing. **Frank Eulitz:** Conceptualization, Supervision, Project administration.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: F.E. is guest editor of the special issue in which this paper is published. L.D. and J.R. declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors would like to thank their colleagues, as well as Luca di Mare from Oxford and Eric Michiels from IBM. L.D. and F.E. acknowledge support from the Flanders Innovation & Entrepreneurship (VLAIO) in Belgium under Grant No. HBC.2023.0897. L.D. is also grateful to the Fondation Wiener-Anspach (FWA) for the postgraduate fellowship that enabled the completion of this project. J.R. acknowledges support from the Belgian Fund for Scientific Research (F.R.S.–FNRS) under Grants No. R.8015.21 (QOPT) and O.0013.22 (EoS CHEQS).

Appendix A. Theoretical study of the QFT algorithms

A theoretical study of the behaviour of the QFT algorithms under noiseless conditions is detailed in this appendix.

A.1. Preliminaries

The algorithms presented in this manuscript estimate Fourier coefficients by sampling the output of several quantum circuits. In order to bound the error introduced in these algorithms, the probability distributions of the samples must be studied. Different probability distributions are involved at different steps of the algorithms, but they generally reduce to special cases of those in the following proposition.

Proposition 1. *Let Y_i be independent and identically distributed random variables such that*

$$\mathrm{P}\left(Y_i = \pm 1\right) = p\,\frac{1 \pm \delta}{2}, \qquad \mathrm{P}\left(Y_i = 0\right) = 1 - p,$$

and let $X_i = Y_i^2 = |Y_i|$. Then, the mean, variance and third central moment of

$$X := \frac{1}{n}\sum_{i=1}^{n} X_i \qquad \textit{and} \qquad Y := \frac{1}{n}\sum_{i=1}^{n} Y_i$$

are given by

$$\mathrm{E}(X) = p, \qquad \mathrm{E}(Y) = p\delta,$$

$$\mathrm{V}(X) = \frac{p(1-p)}{n}, \qquad \mathrm{V}(Y) = \frac{p\left(1 - p\delta^2\right)}{n},$$

$$\mu_3(X) = \frac{p(1-p)(1-2p)}{n^2}, \qquad \mu_3(Y) = \frac{p\delta\left(1 - 3p + 2p\delta^2\right)}{n^2}.$$

Moreover, the covariance of X and Y is given by

$$\mathrm{cov}(X, Y) = \frac{p(1-p)\,\delta}{n}.$$

Proof. Since $Y_i^c = Y_i$ for $c \in \mathbb{N}_0$ odd and $Y_i^c = Y_i^2 = X_i$ for $c \in \mathbb{N}_0$ even, the moments of Y_i and X_i are given by

$$\mathrm{E}\left(Y_i^c\right) = \mathrm{E}\left(Y_i\right) = p\delta \qquad \forall c \in \mathbb{N}_0 \text{ odd},$$
$$\mathrm{E}\left(Y_i^c\right) = \mathrm{E}\left(Y_i^2\right) = p \qquad \forall c \in \mathbb{N}_0 \text{ even},$$
$$\mathrm{E}\left(X_i^c\right) = \mathrm{E}\left(Y_i^2\right) = p \qquad \forall c \in \mathbb{N}_0.$$

Then, the mean, variance and third central moment of X and Y, as well as their covariance, can be obtained from their definition by expanding the sum and using the fact that the Y_i variables are independent.

Note that $nX \sim B(n, p)$ (binomial distribution), so the central moments of X can also be deduced from those of a binomial distribution. □

At some points in what follows, the expectation value of the square root of a random variable will be calculated. The following proposition will be useful.

Proposition 2. *If Z is a random variable taking non negative values, then*

$$\mathrm{E}\left(\sqrt{Z}\right) = \sqrt{\mathrm{E}(Z)}\left(1 - \frac{\mathrm{V}(Z)}{8E^2(Z)} + \mathscr{O}\left(\frac{\mu_3(Z)}{\mathrm{E}^3(Z)}\right)\right),$$
$$\sqrt{\mathrm{E}(Z)}\left(1 - \frac{\mathrm{V}(Z)}{E^2(Z)}\right) \leq \mathrm{E}\left(\sqrt{Z}\right) \leq \sqrt{\mathrm{E}(Z)}.$$

Proof. The first equality follows from the Taylor expansion of $\sqrt{U}$ around $U = 1$,

$$\sqrt{U} = 1 + \frac{U-1}{2} - \frac{(U-1)^2}{8} + \mathscr{O}\left((U-1)^3\right),$$

applied to the random variable $U = Z/\mathrm{E}(Z)$. Calculating the expectation value and multiplying by $\mathrm{E}(Z)$ then yields the claimed equality.

As for the second equation, the upper bound follows from the concavity of the square root function which implies that $\mathrm{E}(\sqrt{Z}) \leq \sqrt{\mathrm{E}(Z)}$, while the lower bound follows from applying the inequality

$$\sqrt{T} \geq 1 + \frac{T-1}{2} - \frac{(T-1)^2}{2}$$

to $T = Z/\mathrm{E}(Z)$ and calculating the expectation value of the resulting equation. □

Applied to the random variables defined in Proposition 1, Proposition 2 leads to the following corollary.

Corollary 1. *If X and Y are random variables distributed as in Proposition 1, then*

$$\mathrm{E}\left(\sqrt{X}\right) = \sqrt{p}\left(1 - \frac{1-p}{8np} + \mathscr{O}\left(\frac{1}{n^2p^2}\right)\right),$$
$$\sqrt{p}\left(1 - \frac{1-p}{2np}\right) \leq \mathrm{E}\left(\sqrt{X}\right) \leq \sqrt{p},$$
$$\mathrm{E}\left(\sqrt{Y}\right) = \sqrt{p\delta}\left(1 - \frac{1-p\delta^2}{8np\delta^2} + \mathscr{O}\left(\frac{1}{n^2p^2\delta^2}\right)\right),$$
$$\sqrt{p\delta}\left(1 - \frac{1-p\delta^2}{2np\delta^2}\right) \leq \mathrm{E}\left(\sqrt{Y}\right) \leq \sqrt{p\delta}.$$

Samples from a random variable Y as in Proposition 1 will also be used to estimate the sign of δ, which will be related to the sign of the real or imaginary part of Fourier coefficients. The following proposition will come in handy.

Proposition 3. *If Y is a random variable distributed as in Proposition 1, then*

$$\mathrm{E}(\mathrm{sgn}(\delta)\,\mathrm{sgn}(Y)) \geq 1 - \frac{2}{np\delta^2},$$
$$\mathrm{E}(\mathrm{sgn}(\delta)\,\mathrm{sgn}(Y)) \geq 1 - \frac{1}{2np\delta^2} + \mathscr{O}\left(\frac{1}{n^2p^2\delta^2}\right).$$

Proof. Chebyshev's inequality implies that, for all $\epsilon > 0$,

$$\mathrm{P}(|Y - p\delta| \geq \epsilon) \leq \frac{\mathrm{V}(Y)}{\epsilon^2},$$

such that, for $\epsilon = p\delta$,

$$\mathrm{P}(|Y - p\delta| \geq p\delta) \leq \frac{\mathrm{V}(Y)}{p^2\delta^2} = \frac{(1-p\delta^2)}{np\delta^2} \leq \frac{1}{np\delta^2}.$$

The fact that $\mathrm{sgn}(Y) = \mathrm{sgn}(\delta)$ whenever $|Y - p\delta| < p\delta$ implies that

$$\begin{aligned}\mathrm{E}(\mathrm{sgn}(\delta)\,\mathrm{sgn}(Y)) &= \mathrm{P}(\mathrm{sgn}(Y) = \mathrm{sgn}(\delta)) - \mathrm{P}(\mathrm{sgn}(Y) \neq \mathrm{sgn}(\delta)) \\ &= 1 - 2\,\mathrm{P}(\mathrm{sgn}(Y) \neq \mathrm{sgn}(\delta)) \\ &\geq 1 - 2\,\mathrm{P}(|Y - p\delta| \geq p\delta) \\ &\geq 1 - \frac{2}{np\delta^2}.\end{aligned}$$

This bound is valid for all values of the parameters but proves quite loose in the limit of large np. The second bound, which only holds asymptotically, might therefore be preferable in this regime. To prove it, note that when $np \gg 1$, the central limit theorem implies that Y can be approximated by a normal random variable U with mean $\mu := \mathrm{E}(Y) = p\delta$ and variance $\sigma^2 := \mathrm{V}(Y) = p(1 - p\delta^2)/n \leq p/n$. For such a variable, the previous equation becomes

$$\begin{aligned}\mathrm{E}(\mathrm{sgn}(\delta)\,\mathrm{sgn}(U)) &= 1 - 2\,\mathrm{P}(\mathrm{sgn}(U) \neq \mathrm{sgn}(\delta)) \\ &= 1 - 2\,\mathrm{P}(\mathrm{sgn}(\delta)\,U \leq 0) \\ &= \mathrm{erf}\left(\frac{\mu}{\sqrt{2}\sigma}\right) \geq \mathrm{erf}\left(\sqrt{\frac{np}{2}}\delta\right),\end{aligned}$$

where the following property of the error function $\mathrm{erf}(x)$ has been used for $L = 0$:

$$\mathrm{P}(U \leq L) = \frac{1}{2} - \frac{1}{2}\,\mathrm{erf}\left(\frac{\mu - L}{\sqrt{2}\sigma}\right).$$

A known expansion for the error function leads to

$$\mathrm{E}(\mathrm{sgn}(\delta)\,\mathrm{sgn}(U)) = 1 - \frac{e^{-x^2}}{x\sqrt{\pi}}\left(1 + \mathscr{O}\left(\frac{1}{x^2}\right)\right),$$

where $x = \sqrt{np/2}\,\delta$, such that the higher order term $\mathscr{O}(1/x^2) = \mathscr{O}(1/np)$ can be neglected. It remains to show that

$$\frac{e^{-x^2}}{x\sqrt{\pi}} \leq \frac{b}{x^2}$$

for $b = 1/4$. This inequality holds as soon as

$$b \geq \frac{1}{\sqrt{\pi}}\,xe^{-x^2}$$

for all $x \in \mathbb{R}^+$, and it can be verified that the function on the right-hand side reaches a maximum at $x = \sqrt{2}/2$, so the inequality holds whenever

$$b \geq \frac{1}{\sqrt{2\pi e}} \approx 0.242.$$ □

It will be shown that the L2 error of most algorithms is bounded by the sum of errors introduced in each step, which is itself typically inversely proportional to the number of samples used in that step. The following proposition will be useful to optimise the number of samples used in each step.

Proposition 4. *Assume that the total error introduced in a two-step estimation algorithm scales as*

$$\epsilon = \frac{\alpha_1}{n_{s,1}} + \frac{\alpha_2}{n_{s,2}},$$

where $n_{s,m}$ is the number of samples used in step m. Then, if the total number of samples is $n_s = n_{s,1} + n_{s,2}$, distributing the samples between the steps as

$$n_{s,1} = \frac{\sqrt{\alpha_1}\,n_s}{\sqrt{\alpha_1} + \sqrt{\alpha_2}} \quad \text{and} \quad n_{s,2} = \frac{\sqrt{\alpha_2}\,n_s}{\sqrt{\alpha_1} + \sqrt{\alpha_2}}$$

minimises the total error as

$$\epsilon = \frac{\left(\sqrt{\alpha_1} + \sqrt{\alpha_2}\right)^2}{n}.$$

In contrast, a uniform distribution $n_{s,1} = n_{s,2} = n_s/2$ leads to

$$\epsilon = \frac{2\left(\alpha_1 + \alpha_2\right)}{n_s}.$$

Proof. Let $n_{s,1} = \mu n_s$ and $n_{s,2} = (1-\mu)n_s$, so that

$$\epsilon_{tot} = \frac{\alpha_1}{\mu n_s} + \frac{\alpha_2}{(1-\mu)\,n_s} = \frac{f(\mu)}{n_s},$$

where

$$f(\mu) = \frac{\alpha_1}{\mu} + \frac{\alpha_2}{(1-\mu)}.$$

Calculating its derivative with respect to μ gives

$$f'(\mu) = -\frac{\alpha_1}{\mu^2} + \frac{\alpha_2}{(1-\mu)^2},$$

which is equal to zero for

$$\mu = \frac{\sqrt{\alpha_1}}{\sqrt{\alpha_1} + \sqrt{\alpha_2}},$$

leading to the claimed optimal values for $n_{s,m}$ and ϵ. The value for $n_{s,1} = n_{s,2} = n_s/2$ follows directly by setting $\mu = 1/2$. □

Note that the best possible improvement with this optimisation is to reduce the error by a factor of 2, when α_1 or α_2 is close to zero. As this modest improvement requires knowledge of the coefficients α_1 and α_2, setting $n_{s,1} = n_{s,2} = n_s/2$ remains a sensible option.

A.2. L2 error

Suppose that one wishes to estimate the QFT state

$$|\hat{\psi}\rangle = \sum_{k=0}^{N-1} \hat{a}_k |k\rangle = \sum_{k=0}^{N-1} |\hat{a}_k| \, e^{i\hat{\varphi}_k} |k\rangle$$

by sampling the output of various quantum circuits, yielding an estimate

$$|\tilde{\psi}\rangle = \sum_{k=0}^{N-1} \tilde{a}_k |k\rangle = \sum_{k=0}^{N-1} |\tilde{a}_k| \, e^{i\tilde{\varphi}_k} |k\rangle .$$

The L2 error on the estimate will then be

$$\epsilon_{\mathrm{L2}}^2 = \| |\hat{\psi}\rangle - |\tilde{\psi}\rangle \|_2^2 = \sum_{k=0}^{N-1} \left| \hat{a}_k - \tilde{a}_k \right|^2 = 2 - 2 \sum_{k=0}^{N-1} |\hat{a}_k| \, |\tilde{a}_k| \cos\left(\hat{\varphi}_k - \tilde{\varphi}_k\right).$$

The estimated Fourier coefficients $\tilde{a}_k$ will behave as random variables since sampling the output of a quantum circuit is a probabilistic procedure. The expected L2 error will therefore be

$$\mathrm{E}\left(\epsilon_{\mathrm{L2}}^2\right) = 2 - 2 \sum_{k=0}^{N-1} |\hat{a}_k| \, \mathrm{E}\left(|\tilde{a}_k| \cos\left(\hat{\varphi}_k - \tilde{\varphi}_k\right)\right) \tag{A.1}$$

$$= 2 - 2 \sum_{k=0}^{N-1} |\hat{a}_k| \left(\cos\hat{\varphi}_k \, \mathrm{E}\left(|\tilde{a}_k| \cos\tilde{\varphi}_k\right) + \sin\hat{\varphi}_k \, \mathrm{E}\left(|\tilde{a}_k| \sin\tilde{\varphi}_k\right)\right). \tag{A.2}$$

Bounding the expected L2 error then boils down to bounding $\mathrm{E}(|\tilde{a}_k| \cos\tilde{\varphi}_k)$ and $\mathrm{E}(|\tilde{a}_k| \sin\tilde{\varphi}_k)$ (algorithms QFT1 and QFT2), or $\mathrm{E}(|\tilde{a}_k| \cos(\hat{\varphi}_k - \tilde{\varphi}_k))$ (algorithm QFT3).

A.3. Algorithm QFT1

The general idea of the algorithm is to estimate the real and imaginary parts of the Fourier coefficients separately by preparing the states $|\hat{\psi}_R\rangle$ and $|\hat{\psi}_I\rangle$ defined by Eq. (18). Note that $|\hat{\psi}\rangle = R|\hat{\psi}_R\rangle + iI|\hat{\psi}_I\rangle$.

Practically, the algorithm first estimates $|\mathrm{Re}(\hat{a}_k)| = |\hat{a}_k \cos\hat{\varphi}_k|$ (step 1.1R) and $|\mathrm{Im}(\hat{a}_k)| = |\hat{a}_k \sin\hat{\varphi}_k|$ (step 1.1I), and then $\mathrm{sgn}(\mathrm{Re}(\hat{a}_k)) = \mathrm{sgn}(\cos\hat{\varphi}_k)$ (step 1.2R) and $\mathrm{sgn}(\mathrm{Im}(\hat{a}_k)) = \mathrm{sgn}(\sin\hat{\varphi}_k)$ (step 1.2I). Since these steps are independent and $\cos\tilde{\varphi}_k = |\cos\tilde{\varphi}_k| \, \mathrm{sgn}(\cos\tilde{\varphi}_k)$ (and similarly for $\sin\tilde{\varphi}_k$), it follows that

$$\mathrm{E}\left(|\tilde{a}_k| \cos\tilde{\varphi}_k\right) = \mathrm{E}\left(\left|\tilde{a}_k \cos\tilde{\varphi}_k\right|\right) \mathrm{E}\left(\mathrm{sgn}\left(\cos\tilde{\varphi}_k\right)\right),$$
$$\mathrm{E}\left(|\tilde{a}_k| \sin\tilde{\varphi}_k\right) = \mathrm{E}\left(\left|\tilde{a}_k \sin\tilde{\varphi}_k\right|\right) \mathrm{E}\left(\mathrm{sgn}\left(\sin\tilde{\varphi}_k\right)\right).$$

In the next subsections, these errors will be bounded one by one, leading to expressions of the type

$$\mathrm{E}\left(\left|\tilde{a}_k \cos\tilde{\varphi}_k\right|\right) \approx \left(1 - \epsilon_{1\mathrm{R},k}\right) \left|\hat{a}_k \cos\hat{\varphi}_k\right|, \tag{A.3a}$$
$$\mathrm{E}\left(\left|\tilde{a}_k \sin\tilde{\varphi}_k\right|\right) \approx \left(1 - \epsilon_{1\mathrm{I},k}\right) \left|\hat{a}_k \sin\hat{\varphi}_k\right|, \tag{A.3b}$$
$$\mathrm{E}\left(\mathrm{sgn}\left(\cos\hat{\varphi}_k\right) \mathrm{sgn}\left(\cos\tilde{\varphi}_k\right)\right) \gtrsim 1 - \epsilon_{2\mathrm{R},k}, \tag{A.3c}$$
$$\mathrm{E}\left(\mathrm{sgn}\left(\sin\hat{\varphi}_k\right) \mathrm{sgn}\left(\sin\tilde{\varphi}_k\right)\right) \gtrsim 1 - \epsilon_{2\mathrm{I},k}. \tag{A.3d}$$

Substituting these into Eq. (A.2) leads to

$$\begin{aligned}\mathrm{E}\left(\epsilon_{\mathrm{L2}}^2\right) &\lesssim 2 - 2 \sum_{k=0}^{N-1} |\hat{a}_k|^2 \left(\cos^2\hat{\varphi}_k \left(1 - \epsilon_{1\mathrm{R},k}\right)\left(1 - \epsilon_{2\mathrm{R},k}\right)\right. \\ &\qquad \left. + \sin^2\hat{\varphi}_k \left(1 - \epsilon_{1\mathrm{I},k}\right)\left(1 - \epsilon_{2\mathrm{I},k}\right)\right) \\ &\lesssim 2 \sum_{k=0}^{N-1} |\hat{a}_k|^2 \left(\cos^2\hat{\varphi}_k \left(\epsilon_{1\mathrm{R},k} + \epsilon_{2\mathrm{R},k}\right) + \sin^2\hat{\varphi}_k \left(\epsilon_{1\mathrm{I},k} + \epsilon_{2\mathrm{I},k}\right)\right),\end{aligned}$$

which can be rewritten as

$$\mathrm{E}\left(\epsilon_{\mathrm{L2}}^2\right) \lesssim \epsilon_{1\mathrm{R}} + \epsilon_{1\mathrm{I}} + \epsilon_{2\mathrm{R}} + \epsilon_{2\mathrm{I}}, \tag{A.4}$$

where the errors ϵ_i are defined as

$$\epsilon_{1\mathrm{R}} = 2 \sum_{k=0}^{N-1} \left|\hat{a}_k \cos\hat{\varphi}_k\right|^2 \epsilon_{1\mathrm{R},k}, \tag{A.5a}$$
$$\epsilon_{1\mathrm{I}} = 2 \sum_{k=0}^{N-1} \left|\hat{a}_k \sin\hat{\varphi}_k\right|^2 \epsilon_{1\mathrm{I},k}, \tag{A.5b}$$
$$\epsilon_{2\mathrm{R}} = 2 \sum_{k=0}^{N-1} \left|\hat{a}_k \cos\hat{\varphi}_k\right|^2 \epsilon_{2\mathrm{R},k}, \tag{A.5c}$$
$$\epsilon_{2\mathrm{I}} = 2 \sum_{k=0}^{N-1} \left|\hat{a}_k \sin\hat{\varphi}_k\right|^2 \epsilon_{2\mathrm{I},k}. \tag{A.5d}$$

A.3.1. Errors introduced in steps 1.1R and 1.1I

In step 1.1R, $|\hat{a}_k \cos\hat{\varphi}_k|$ is estimated from a random variable X as in Proposition 1 with $n = n_{s,1\mathrm{R}}$ and

$$p = p_k := \frac{\left|\hat{a}_k \cos\hat{\varphi}_k\right|^2}{R^2}.$$

Assuming that $p_k \ll 1$ and $n_{s,1\mathrm{R}} p_k \gg 1$, Proposition 1 and Corollary 1 lead to

$$\mathrm{E}\left(\frac{\left|\tilde{a}_k \cos\tilde{\varphi}_k\right|^2}{R^2}\right) = p_k = \frac{\left|\hat{a}_k \cos\hat{\varphi}_k\right|^2}{R^2},$$
$$\begin{aligned}\mathrm{E}\left(\frac{\left|\tilde{a}_k \cos\tilde{\varphi}_k\right|}{R}\right) &= \sqrt{p_k}\left(1 - \frac{1 - p_k}{8 n_{s,1\mathrm{R}} p_k} + \mathcal{O}\left(\frac{1}{n_{s,1\mathrm{R}}^2 p_k^2}\right)\right) \\ &\approx \frac{\left|\hat{a}_k \cos\hat{\varphi}_k\right|}{R}\left(1 - \frac{R^2}{8 n_{s,1\mathrm{R}} \left|\hat{a}_k \cos\hat{\varphi}_k\right|^2}\right).\end{aligned}$$

The last expression can be rewritten as

$$\mathrm{E}\left(\left|\tilde{a}_k \cos\tilde{\varphi}_k\right|\right) \approx \left(1 - \epsilon_{1\mathrm{R},k}\right) \left|\hat{a}_k \cos\hat{\varphi}_k\right|,$$

where the error $\epsilon_{1\mathrm{R},k}$ is defined as

$$\epsilon_{1\mathrm{R},k} = \frac{R^2}{8 n_{s,1\mathrm{R}} \left|\hat{a}_k \cos\hat{\varphi}_k\right|^2}.$$

As a result,

$$\epsilon_{1\mathrm{R}} = 2 \sum_{k=0}^{N-1} \frac{R^2}{8 n_{s,1\mathrm{R}}} = \frac{R^2}{4} \frac{N}{n_{s,1\mathrm{R}}}. \tag{A.6}$$

A similar development for the estimation of $|\hat{a}_k \sin\hat{\varphi}_k|$ in step 1.1I yields

$$\mathrm{E}\left(\left|\tilde{a}_k \sin\tilde{\varphi}_k\right|\right) \approx \left(1 - \epsilon_{1\mathrm{I},k}\right) \left|\hat{a}_k \sin\hat{\varphi}_k\right|,$$

where

$$\epsilon_{1\mathrm{I},k} = \frac{I^2}{8 n_{s,1\mathrm{I}} \left|\hat{a}_k \sin\hat{\varphi}_k\right|^2},$$

such that

$$\epsilon_{1\mathrm{I}} = \frac{I^2}{4} \frac{N}{n_{s,1\mathrm{I}}}. \tag{A.7}$$

A.3.2. Errors introduced in steps 1.2R and 1.2I

In step 1.2R, $\operatorname{sgn}(\cos\hat{\varphi}_k)$ is estimated from a random variable Y as in Proposition 1 with $n = n_{s,2\mathrm{R}}$,

$$p = p_k := \frac{1}{2}\left(\frac{1}{N} + \frac{|\hat{a}_k\cos\hat{\varphi}_k|^2}{R^2}\right), \qquad \delta = \delta_k := \frac{1}{p_k\sqrt{N}R}\,|\hat{a}_k|\cos\hat{\varphi}_k\,.$$

Proposition 3 implies that

$$\mathrm{E}\left(\operatorname{sgn}\left(\cos\hat{\varphi}_k\right)\operatorname{sgn}\left(\cos\tilde{\varphi}_k\right)\right) \gtrsim 1 - \epsilon_{2\mathrm{R},k}\,,$$

where

$$\epsilon_{2\mathrm{R},k} = \frac{1}{2n_{s,2\mathrm{R}}p_k\delta_k^2} = \frac{NR^2p_k}{2n_{s,2\mathrm{R}}\,|\hat{a}_k\cos\hat{\varphi}_k|^2}\,.$$

This leads to

$$\epsilon_{2\mathrm{R}} = 2\sum_{k=0}^{N-1}\frac{NR^2p_k}{2n_{s,2\mathrm{R}}} = R^2\frac{N}{n_{s,2\mathrm{R}}}\,. \tag{A.8}$$

A similar development for the estimation of $\operatorname{sgn}(\sin\hat{\varphi}_k)$ in step 1.2I gives

$$\mathrm{E}\left(\operatorname{sgn}\left(\sin\hat{\varphi}_k\right)\operatorname{sgn}\left(\sin\tilde{\varphi}_k\right)\right) \gtrsim 1 - \epsilon_{2\mathrm{I},k}\,,$$

where

$$\epsilon_{2\mathrm{I},k} = \frac{NI^2p_k}{2n_{s,2\mathrm{I}}\,|\hat{a}_k\sin\hat{\varphi}_k|^2}\,,$$

such that

$$\epsilon_{2\mathrm{I}} = I^2\frac{N}{n_{s,2\mathrm{I}}}\,. \tag{A.9}$$

A.3.3. Expected L2 error

The expected L2 error introduced in the algorithm for a total of $n_s = n_{s,1\mathrm{R}} + n_{s,1\mathrm{I}} + n_{s,2\mathrm{R}} + n_{s,2\mathrm{I}}$ samples is thus bounded by Eq. (A.4) as

$$\mathrm{E}\left(\epsilon_{\mathrm{L2}}^2\right) \lesssim R^2\left(\frac{1}{4}\frac{N}{n_{s,1\mathrm{R}}} + \frac{N}{n_{s,2\mathrm{R}}}\right) + I^2\left(\frac{1}{4}\frac{N}{n_{s,1\mathrm{I}}} + \frac{N}{n_{s,2\mathrm{I}}}\right). \tag{A.10}$$

A.3.4. Error minimisation

Assume that a total of $n_{s,1}$ samples are used in steps 1.1R and 1.1I, and are divided into $n_{s,1\mathrm{R}} = \mu_1 n_{s,1}$ samples to estimate the real parts and $n_{s,1\mathrm{I}} = (1-\mu_1)n_{s,1}$ samples to estimate the imaginary parts. Similarly, assume that, in steps 1.2R and 1.2I, $n_{s,2}$ samples are divided into $n_{s,2\mathrm{R}} = \mu_2 n_{s,2}$ samples for the real parts and $n_{s,2\mathrm{I}} = (1-\mu_2)n_{s,2}$ samples for the imaginary parts.

Eq. (A.10) then becomes

$$\mathrm{E}\left(\epsilon_{\mathrm{L2}}^2\right) \lesssim \underbrace{\frac{1}{4}\left(\frac{R^2}{\mu_1} + \frac{I^2}{1-\mu_1}\right)\frac{N}{n_{s,1}}}_{\epsilon_1} + \underbrace{\left(\frac{R^2}{\mu_2} + \frac{I^2}{1-\mu_2}\right)\frac{N}{n_{s,2}}}_{\epsilon_2}\,,$$

where ϵ_1 and ϵ_2 can be minimised by setting $\mu_1 = \mu_2 = R/(R+I)$, according to Proposition 4. This leads to

$$\mathrm{E}\left(\epsilon_{\mathrm{L2}}^2\right) \lesssim (1+2RI)\left(\frac{1}{4}\frac{N}{n_{s,1}} + \frac{N}{n_{s,2}}\right),$$

while the naive choice $\mu_1 = \mu_2 = 1/2$ results in

$$\mathrm{E}\left(\epsilon_{\mathrm{L2}}^2\right) \lesssim 2\left(\frac{1}{4}\frac{N}{n_{s,1}} + \frac{N}{n_{s,2}}\right).$$

Therefore, the number of samples required to estimate the Fourier coefficients with a given accuracy can be reduced by up to a factor of 2 when the coefficients are known to be purely real ($I = 0$) or imaginary ($R = 0$).

A.4. Algorithm QFT2

The algorithm first estimates $|\mathrm{Re}(\hat{a}_k)| = |\hat{a}_k\cos\hat{\varphi}_k|$ and $|\mathrm{Im}(\hat{a}_k)| = |\hat{a}_k\sin\hat{\varphi}_k|$ (step 2.1), and then $\operatorname{sgn}(\mathrm{Re}(\hat{a}_k)) = \operatorname{sgn}(\cos\hat{\varphi}_k)$ (step 2.2R) and $\operatorname{sgn}(\mathrm{Im}(\hat{a}_k)) = \operatorname{sgn}(\sin\hat{\varphi}_k)$ (step 2.2I). It is very similar to algorithm QFT1, except that the real and imaginary parts are estimated together. This analysis therefore follows the same lines as the previous one.

A.4.1. Error introduced in step 2.1

In step 2.1, $|\hat{a}_k\cos\hat{\varphi}_k|$ is estimated from a random variable X as in Proposition 1, with $n = n_{s,1}$ and $p = p_k := |\hat{a}_k\cos\hat{\varphi}_k|^2$. Then, assuming that $p_k \ll 1$ and $n_{s,1}p_k \gg 1$,

$$\begin{aligned}\mathrm{E}\left(\left|\tilde{a}_k\cos\tilde{\varphi}_k\right|^2\right) &= p_k = \left|\hat{a}_k\cos\hat{\varphi}_k\right|^2,\\ \mathrm{E}\left(\left|\tilde{a}_k\cos\tilde{\varphi}_k\right|\right) &= \sqrt{p_k}\left(1 - \frac{1-p_k}{8n_{s,1}p_k} + \mathcal{O}\left(\frac{1}{n_{s,1}^2p_k^2}\right)\right)\\ &\approx \left|\hat{a}_k\cos\hat{\varphi}_k\right|\left(1 - \frac{1}{8n_{s,1}\,|\hat{a}_k\cos\hat{\varphi}_k|^2}\right),\end{aligned}$$

which can be rewritten as

$$\mathrm{E}\left(\left|\tilde{a}_k\cos\tilde{\varphi}_k\right|\right) \approx \left(1 - \epsilon_{1\mathrm{R},k}\right)\left|\hat{a}_k\cos\hat{\varphi}_k\right|,$$

where

$$\epsilon_{1\mathrm{R},k} = \frac{1}{8n_{s,1}\,|\hat{a}_k\cos\hat{\varphi}_k|^2}\,.$$

This results in

$$\epsilon_{1\mathrm{R}} = 2\sum_{k=0}^{N-1}\frac{1}{8n_{s,1}} = \frac{1}{4}\frac{N}{n_{s,1}}\,. \tag{A.11}$$

Similarly,

$$\mathrm{E}\left(\left|\tilde{a}_k\sin\tilde{\varphi}_k\right|\right) \approx \left(1 - \epsilon_{1\mathrm{I},k}\right)\left|\hat{a}_k\sin\hat{\varphi}_k\right|,$$

where

$$\epsilon_{1\mathrm{I},k} = \frac{1}{8n_{s,1}\,|\hat{a}_k\sin\hat{\varphi}_k|^2}\,,$$

such that

$$\epsilon_{1\mathrm{I}} = \frac{1}{4}\frac{N}{n_{s,1}}\,. \tag{A.12}$$

A.4.2. Comparison with steps 1.1R and 1.1I

Recall the discussion in Section 7.4, which concluded that step 2.1 was similar to steps 1.1R and 1.1I when

$$n_{s,1\mathrm{R}} = \mu n_{s,1} \qquad \text{and} \qquad n_{s,1\mathrm{I}} = (1-\mu)\,n_{s,1}\,, \qquad \text{with} \qquad \mu = R^2\,.$$

Substituting these into the combined error of steps 1.1R and 1.1I gives

$$\epsilon_{1\mathrm{R}} + \epsilon_{1\mathrm{I}} = \frac{R^2}{4}\frac{N}{n_{s,1\mathrm{R}}} + \frac{I^2}{4}\frac{N}{n_{s,1\mathrm{I}}} = \frac{1}{4}\left(\frac{R^2}{\mu} + \frac{I^2}{1-\mu}\right)\frac{N}{n_{s,1}} = \frac{1}{2}\frac{N}{n_{s,1}}\,,$$

which is the exact same error as in step 2.1. Step 2.1 is therefore equivalent to steps 1.1R and 1.1I with the above sample distribution.

A.4.3. Error introduced in steps 2.2R and 2.2I

In step 2.2R, $\operatorname{sgn}(\cos\hat{\varphi}_k)$ is estimated from a random variable Y as in Proposition 1 with $n = n_{s,2\mathrm{R}}$,

$$p = p_k := \frac{1}{2}\left(\frac{1}{N} + |\hat{a}_k|^2\right), \qquad \delta = \delta_k := \frac{1}{p_k\sqrt{N}}\,|\hat{a}_k|\cos\hat{\varphi}_k\,.$$

Then,

$$\mathrm{E}\left(\operatorname{sgn}\left(\cos\hat{\varphi}_k\right)\operatorname{sgn}\left(\cos\tilde{\varphi}_k\right)\right) \gtrsim 1 - \epsilon_{2\mathrm{R},k}\,,$$

where

$$\epsilon_{2\mathrm{R},k} = \frac{1}{2n_{s,2\mathrm{R}}p_k\delta_k^2} = \frac{Np_k}{2n_{s,2\mathrm{R}}\,|\hat{a}_k\cos\hat{\varphi}_k|^2}\,.$$

This leads to

$$\epsilon_{2\mathrm{R}} = 2\sum_{k=0}^{N-1}\frac{Np_k}{2n_{s,2\mathrm{R}}} = \frac{N}{n_{s,2\mathrm{R}}}\,. \tag{A.13}$$

Similarly,

$$\mathrm{E}\left(\mathrm{sgn}\left(\sin\hat{\varphi}_k\right)\mathrm{sgn}\left(\sin\tilde{\varphi}_k\right)\right) \gtrsim 1-\epsilon_{2\mathrm{I},k}\,,$$

where

$$\epsilon_{2\mathrm{I},k} = \frac{Np_k}{2n_{s,2\mathrm{I}}\left|\hat{a}_k\sin\hat{\varphi}_k\right|^2}\,,$$

such that

$$\epsilon_{2\mathrm{I}} = \frac{N}{n_{s,2\mathrm{I}}}\,. \tag{A.14}$$

A.4.4. Expected L2 error

The expected L2 error introduced in the algorithm for a total of $n_s = n_{s,1} + n_{s,2\mathrm{R}} + n_{s,2\mathrm{I}}$ samples is thus bounded by Eq. (A.4) as

$$\mathrm{E}\left(\epsilon_{\mathrm{L2}}^2\right) \lesssim \frac{1}{2}\frac{N}{n_{s,1}} + \frac{N}{n_{s,2\mathrm{R}}} + \frac{N}{n_{s,2\mathrm{I}}}\,.$$

A.5. Algorithm QFT3

The algorithm proceeds in two phases. It first estimates $|\hat{a}_k|$ and $\cos 2\hat{\varphi}_k$ (step 3.1C) as well as $|\hat{a}_k|$ and $\sin 2\hat{\varphi}_k$ (step 3.1S), from which it derives a refined estimate of $|\hat{a}_k|$ and an estimate $\tilde{\phi}_k$ of $\hat{\phi}_k = \hat{\varphi}_k \bmod \pi$. It then evaluates whether $\tilde{\phi}_k$ is close to $\hat{\varphi}_k$ or to $\hat{\varphi}_k + \pi$ (step 3.2), that is, whether the final estimate $\tilde{\varphi}_k$ for $\hat{\varphi}_k$ should be $\tilde{\phi}_k$ or $\tilde{\phi}_k + \pi$.

Let $s_k = \pm 1$ be defined as $s_k = \mathrm{sgn}\left(\cos(\tilde{\varphi}_k - \tilde{\phi}_k)\right)$. Using the identity $\cos(x+\pi) = -\cos x$, Eq. (A.1) can be rewritten as

$$\begin{aligned}\mathrm{E}\left(\epsilon_{\mathrm{L2}}^2\right) &= 2 - 2\sum_{k=0}^{N-1}|\hat{a}_k|\,\mathrm{E}\left(|\tilde{a}_k|\,s_k\cos\left(\hat{\varphi}_k-\tilde{\phi}_k\right)\right)\\ &= 2 - 2\sum_{k=0}^{N-1}|\hat{a}_k|\,\mathrm{E}\left(s_k\,\mathrm{sgn}\left(\cos\left(\hat{\varphi}_k-\tilde{\phi}_k\right)\right)\left|\tilde{a}_k\cos\left(\hat{\varphi}_k-\tilde{\phi}_k\right)\right|\right).\end{aligned}$$

Note that $|\tilde{a}_k\cos\left(\hat{\varphi}_k - \tilde{\phi}_k\right)|$ and s_k are evaluated during steps 3.1 and 3.2, respectively. Indeed, as demonstrated in the next section, $|\tilde{a}_k\cos\left(\hat{\varphi}_k - \tilde{\phi}_k\right)|$ can be calculated from quantities estimated in step 3.1 only, and it can be verified that $s_k = \mathrm{sgn}\left(\sigma_k\right)$. Unlike the previous algorithms, the total expected error cannot be broken down into a sum of errors introduced in each step, because these are not independent. Indeed, the measurement basis used to estimate s_k in step 3.2 is based on the value of $\tilde{\phi}_k$ obtained in step 3.1.

Nevertheless, assuming that the errors in steps 3.1 and 3.2 are approximately independent and obey

$$\mathrm{E}\left(\left|\tilde{a}_k\cos\left(\hat{\varphi}_k-\tilde{\phi}_k\right)\right|\right) \approx \left(1-\epsilon_{1,k}\right)|\hat{a}_k|\,, \tag{A.15a}$$

$$\mathrm{E}\left(s_k\,\mathrm{sgn}\left(\cos\left(\hat{\varphi}_k-\tilde{\phi}_k\right)\right)\right) \gtrsim 1-\epsilon_{2,k}\,, \tag{A.15b}$$

the expected L2 error can be approximated as

$$\mathrm{E}\left(\epsilon_{\mathrm{L2}}^2\right) \lesssim 2 - 2\sum_{k=0}^{N-1}|\hat{a}_k|^2\left(1-\epsilon_{1,k}\right)\left(1-\epsilon_{2,k}\right) \le \epsilon_1+\epsilon_2\,, \tag{A.16}$$

where

$$\epsilon_1 = 2\sum_{k=0}^{N-1}|\hat{a}_k|^2\,\epsilon_{1,k} \qquad \text{and} \qquad \epsilon_2 = 2\sum_{k=0}^{N-1}|\hat{a}_k|^2\,\epsilon_{2,k}\,. \tag{A.17}$$

A.5.1. Error introduced in steps 3.1C and 3.1S

In order to estimate $\hat{\phi}_k$ and $|\hat{a}_k|$, step 3.1C uses $n = n_{s,1\mathrm{C}}$ random variables $Y_{C,i}$ as in Proposition 1, with $p = p_k := |\hat{a}_k|^2$ and $\delta = \delta_k := \cos 2\hat{\varphi}_k$, while step 3.1S uses $n = n_{s,1\mathrm{S}}$ random variables $Y_{S,i}$, also as in Proposition 1, but with $p = p_k := |\hat{a}_k|^2$ and $\delta = \delta_k := \sin 2\hat{\varphi}_k$.

Assuming that $n_{s,1} = n_{s,1\mathrm{C}} = n_{s,1\mathrm{S}}$ and defining

$$Y_C := \frac{1}{n_{s,1}}\sum_{i=1}^{n_{s,1}}Y_{C,i}\,, \qquad Y_S := \frac{1}{n_{s,1}}\sum_{i=1}^{n_{s,1}}Y_{S,i}\,,$$

$$X_C := \frac{1}{n_{s,1}}\sum_{i=1}^{n_{s,1}}Y_{C,i}^2\,, \qquad X_S := \frac{1}{n_{s,1}}\sum_{i=1}^{n_{s,1}}Y_{S,i}^2\,,$$

$$W := \frac{1}{2}\left(X_C + X_S\right),$$

Proposition 1 implies that

$$\mathrm{E}\left(Y_C\right) = |\hat{a}_k|^2\cos 2\hat{\varphi}_k\,, \qquad \mathrm{V}\left(Y_C\right) = \frac{|\hat{a}_k|^2\left(1-|\hat{a}_k|^2\cos^2 2\hat{\varphi}_k\right)}{n_{s,1}}\,,$$

$$\mathrm{E}\left(Y_S\right) = |\hat{a}_k|^2\sin 2\hat{\varphi}_k\,, \qquad \mathrm{V}\left(Y_S\right) = \frac{|\hat{a}_k|^2\left(1-|\hat{a}_k|^2\sin^2 2\hat{\varphi}_k\right)}{n_{s,1}}\,,$$

$$\mathrm{E}\left(W\right) = |\hat{a}_k|^2\,, \qquad \mathrm{V}\left(W\right) = \frac{|\hat{a}_k|^2\left(1-|\hat{a}_k|^2\right)}{2n_{s,1}}\,,$$

$$\mathrm{cov}\left(W,Y_C\right) = \frac{|\hat{a}_k|^2\left(1-|\hat{a}_k|^2\right)\cos 2\hat{\varphi}_k}{2n_{s,1}}\,,$$

$$\mathrm{cov}\left(W,Y_S\right) = \frac{|\hat{a}_k|^2\left(1-|\hat{a}_k|^2\right)\sin 2\hat{\varphi}_k}{2n_{s,1}}\,.$$

The third line allows to define an estimate $|\tilde{a}_k|^2 = W$ of $|\hat{a}_k|^2$, while the first two lines should be involved in estimating $\hat{\phi}_k$. However, an estimate $\tilde{\phi}_k$ cannot be obtained from

$$\cos 2\tilde{\phi}_k = \frac{Y_C}{W} \qquad \text{and} \qquad \sin 2\tilde{\phi}_k = \frac{Y_S}{W}\,,$$

as equality $Y_C^2 + Y_S^2 = W^2$ will in general only be approximately satisfied. Therefore, it is rather defined as

$$\tilde{\phi}_k = \frac{1}{2}\,\mathrm{atan2}\left(Y_S, Y_C\right).$$

Nevertheless, given the difficulty of deriving theoretical bounds for the expectation value of a random variable which involves a function such as atan2, it is assumed for the sake of error analysis that

$$Y_C = |\tilde{a}_k|^2\cos 2\tilde{\phi}_k \qquad \text{and} \qquad Y_S = |\tilde{a}_k|^2\sin 2\tilde{\phi}_k$$

hold. This allows to use the propositions outlined in Appendix A.1 to derive error bounds, although this assumption leads to a slight underestimation of the actual errors.

Recall that one must calculate

$$\mathrm{E}\left(\left|\tilde{a}_k\cos\left(\hat{\varphi}_k-\tilde{\phi}_k\right)\right|\right) = \mathrm{E}\left(\sqrt{Z}\right),$$

where the new random variable Z is defined as

$$\begin{aligned}Z &:= |\tilde{a}_k|^2\cos^2\left(\hat{\varphi}_k-\tilde{\phi}_k\right)\\ &= |\tilde{a}_k|\,\frac{1+\cos\left(2\left(\hat{\varphi}_k-\tilde{\phi}_k\right)\right)}{2}\\ &= \frac{1}{2}\left(|\tilde{a}_k|^2 + |\tilde{a}_k|^2\cos 2\tilde{\phi}_k\cos 2\hat{\varphi}_k + |\tilde{a}_k|^2\sin 2\tilde{\phi}_k\sin 2\hat{\varphi}_k\right)\\ &= \frac{1}{2}\left(W + Y_C\cos 2\hat{\varphi}_k + Y_S\sin 2\hat{\varphi}_k\right),\end{aligned}$$

such that $\mathrm{E}\left(\sqrt{Z}\right)$ can be evaluated using Proposition 2, which implies that

$$\mathrm{E}\left(\sqrt{Z}\right) \approx 1 - \frac{\mathrm{V}\left(Z\right)}{8\,\mathrm{E}^2\left(Z\right)}\,.$$

The expectation value and variance of Z can be worked out as

$$\begin{aligned}\mathrm{E}\left(Z\right) &= \frac{1}{2}\left(\mathrm{E}\left(W\right) + \mathrm{E}\left(Y_C\right)\cos 2\hat{\varphi}_k + \mathrm{E}\left(Y_S\right)\sin 2\hat{\varphi}_k\right)\\ &= \frac{1}{2}\left(|\hat{a}_k|^2 + |\hat{a}_k|^2\cos^2 2\hat{\varphi}_k + |\hat{a}_k|^2\sin^2 2\hat{\varphi}_k\right)\\ &= |\hat{a}_k|^2\,,\end{aligned}$$

$$\begin{aligned}\mathrm{V}(Z) &= \frac{1}{4}\left(\mathrm{V}(W) + \mathrm{V}(Y_C)\cos^2 2\hat{\varphi}_k + \mathrm{V}(Y_S)\sin^2 2\hat{\varphi}_k \right.\\ &\qquad \left. + 2\cos 2\hat{\varphi}_k \operatorname{cov}(W, Y_C) + 2\sin 2\hat{\varphi}_k \operatorname{cov}(W, Y_S)\right)\\ &\le \frac{1}{4}\left(\frac{|\hat{a}_k|^2}{2n_{s,1}} + \frac{|\hat{a}_k|^2}{n_{s,1}}\cos^2 2\hat{\varphi}_k + \frac{|\hat{a}_k|^2}{n_{s,1}}\sin^2 2\hat{\varphi}_k \right.\\ &\qquad \left. + \frac{|\hat{a}_k|^2}{n_{s,1}}\cos^2 2\hat{\varphi}_k + \frac{|\hat{a}_k|^2}{n_{s,1}}\sin^2 2\hat{\varphi}_k\right)\\ &\le \frac{5|\hat{a}_k|^2}{8n_{s,1}},\end{aligned}$$

which yields

$$\mathrm{E}\left(\sqrt{Z}\right) \approx 1 - \frac{5}{64\,|\hat{a}_k|^2\, n_{s,1}}.$$

This can be rewritten as

$$\mathrm{E}\left(\left|\tilde{a}_k \cos\left(\hat{\varphi}_k - \tilde{\phi}_k\right)\right|\right) \approx \left(1 - \epsilon_{1,k}\right)|\hat{a}_k|,$$

where

$$\epsilon_{1,k} = \frac{5}{64\,|\hat{a}_k|^2\, n_{s,1}},$$

resulting in

$$\epsilon_1 = 2\sum_{k=0}^{N-1} |\hat{a}_k|^2\, \epsilon_{1,k} = \frac{5}{32}\frac{N}{n_{s,1}}. \tag{A.18}$$

A.5.2. Error introduced in step 3.2

Step 3.2 calculates an estimate s_k of $\operatorname{sgn}(\cos(\hat{\varphi}_k - \tilde{\phi}_k))$ by evaluating the sign of a random variable Y as in Proposition 1 with $n = n_{s,2}$,

$$p = p_k := \frac{|\hat{a}_{C,k}|^2 + |\tilde{a}_{S,k}|^2}{2} \approx |\hat{a}_k|^2,$$

$$\delta = \delta_k := \frac{2\,|\hat{a}_k|\,|\tilde{a}_k|}{|\hat{a}_k|^2 + |\tilde{a}_k|^2}\cos\left(\hat{\varphi}_k - \tilde{\phi}_k\right) \approx \cos\left(\hat{\varphi}_k - \tilde{\phi}_k\right).$$

Proposition 3 leads to

$$\begin{aligned}\mathrm{E}\left(s_k \operatorname{sgn}\left(\cos\left(\hat{\varphi}_k - \tilde{\phi}_k\right)\right)\right) \gtrsim 1 - \epsilon_{2,k} &\ge 1 - \frac{1}{2n_{s,2}p_k\delta_k^2}\\ &\approx 1 - \frac{1}{2n_{s,2}\,|\hat{a}_k|^2\cos^2\left(\hat{\varphi}_k - \tilde{\phi}_k\right)}\\ &\approx 1 - \frac{1}{2n_{s,2}\,|\hat{a}_k|^2}.\end{aligned}$$

Consequently,

$$\epsilon_{2,k} \lesssim \frac{1}{2n_{s,2}\,|\hat{a}_k|^2},$$

such that

$$\epsilon_2 = 2\sum_{k=0}^{N-1} |\hat{a}_k|^2\, \epsilon_{2,k} \lesssim \frac{N}{n_{s,2}}. \tag{A.19}$$

A.5.3. Expected L2 error

The expected L2 error introduced in the algorithm for a total of $n_s = 2n_{s,1} + n_{s,2}$ samples is thus bounded by Eq. (A.16) as

$$\mathrm{E}\left(\epsilon_{\mathrm{L2}}^2\right) \lesssim \frac{5}{32}\frac{N}{n_{s,1}} + \frac{N}{n_{s,2}}.$$

Appendix B. Numerical study of the QFT algorithms

A numerical study of the behaviour of the QFT algorithms under noiseless and noisy conditions is detailed in this appendix and compared with the theoretical predictions from Appendix A. It focuses on the general case of calculating the QFT of arbitrary complex signals. The behaviour of the algorithms is independent of the input signal because the algorithms are invariant by rotation, i.e., coefficient phases do not influence algorithmic behaviour.

The ϵ_i errors introduced in the various steps of the algorithms are estimated numerically from a large number of simulations. These errors are defined by Eq. (A.5) for algorithms QFT1 and QFT2, and by Eq. (A.17) for QFT3, with $\epsilon_{i,k}$ derived from Eqs. (A.3) and (A.15), respectively. The expectation values in the latter expressions are approximated as the means of n_{rep} samples of the relevant variables obtained under identical conditions, i.e., by repeating the algorithm n_{rep} times for the same signal. The theoretical bounds on ϵ_i are given by Eqs. (A.6)–(A.9) for QFT1, Eqs. (A.11)–(A.14) for QFT2, and Eqs. (A.18)–(A.19) for QFT3. Since a large number of complex signals uniformly distributed around zero were considered (see below), on average, $R^2 = I^2 = 1/2$ for QFT1. Note that for QFT2, the bounds on $\epsilon_{1\mathrm{R}}$ and $\epsilon_{1\mathrm{I}}$ are both functions of $n_{s,1}$, whereas for QFT3 the bound on ϵ_1 is a function of $n_{s,1} = n_{s,1\mathrm{C}} = n_{s,1\mathrm{S}}$.

For greater clarity in what follows, steps 1.1R, 1.1I, 2.1, 3.1C and 3.1S will be referred to as 'amplitude-estimating steps', while steps 1.2R, 1.2I, 2.2R, 2.2I and 3.2 will be called 'ambiguity-lifting steps'.

B.1. Methodology

The Fourier coefficients were estimated by the QFT algorithms using the numerical configuration described in Section 8, while their true values were calculated using the FFT algorithm. The QFT algorithms were parameterised to distribute n_s samples evenly between their steps m, i.e., $n_{s,m} = n_s/4$ for QFT1 and $n_{s,m} = n_s/3$ for QFT2 and QFT3. Their behaviour was evaluated for normalised complex signals of $N = 4$ to 128 elements encoded on $n = 2$ to 7 qubits, over a range of total numbers of samples $n_s = \sum_m n_{s,m}$ from 10^2 to 10^7. For each (n, n_s) combination, $n_{sig} = 100$ signals were generated randomly following a uniform distribution centred around zero and normalised. For each signal, the algorithm was run n_{rep} times to estimate the expectation values, resulting in a single value per error ϵ_i. The number of repetitions n_{rep} was set to 1000 for the noiseless simulations and 100 for the noisy ones. Noise levels p_d of 0.001%, 0.01% and 0.1% were investigated.

B.2. Results

The means and 95% confidence intervals (CI) of the n_{sig} errors ϵ_i were calculated for each $(n, n_{s,m})$ combination and are plotted in Figs. B.17–B.19. Each column shows the errors for a given noise level whilst each row focuses on a single error. Asymptotic scalings of the errors were estimated by fitting power laws of the form

$$\epsilon_i = A\left(\frac{N}{n_{s,m}}\right)^{\alpha}$$

on the mean errors for $n_{s,m}/N > 10^1$. These are compared with the theoretical bounds in Table B.2, and both are shown on the graphs.

The means and 95% CI of the L2 errors ϵ_{L2}^2 were calculated similarly for each (n, n_s) and are shown in Fig. B.21. Each column shows the errors for a given noise level whilst each row focuses on the error of a single algorithm. They can be compared with their theoretical bounds and with the sums of the asymptotic fits of the errors introduced in each step, called 'ϵ_i fits' in what follows. Both were calculated as $\epsilon_{\mathrm{L2}}^2 = \sum_i \epsilon_i$ from the bounds and fits given in Table B.2, with the appropriate numbers of samples. Additional asymptotic scalings were estimated numerically by linear regression on the mean L2 errors for $n_s/N > 10^3$. Dubbed 'L2 fits', these are not shown on the graphs but are reported in Table B.3 alongside the constants of the theoretical bounds.

B.2.1. Noiseless behaviour

Fig. B.17 (a) shows the error scalings for algorithm QFT1 in the noiseless configuration. While $\epsilon_{1\mathrm{R}}$ and $\epsilon_{1\mathrm{I}}$ closely match their theoretical bounds, $\epsilon_{2\mathrm{R}}$ and $\epsilon_{2\mathrm{I}}$ outperform theirs. The asymptotic fits suggest that the errors introduced in the ambiguity-lifting steps scale as $\mathcal{O}((N/n_{s,m})^{3/2})$. As expected, similar conclusions are drawn for QFT2, whose noiseless scalings are shown in Fig. B.18 (a). However, an

Table B.2
Comparison of the observed asymptotic behaviour of the ϵ_i errors with their theoretical bounds, for all algorithms under noiseless conditions. The asymptotic scalings were estimated numerically by fitting power laws of the form $\epsilon_i = A(N/n_{s,m})^\alpha$ on the mean errors for $n_{s,m}/N > 10^1$. The theoretical bounds are transposed into the same power laws.

(a) QFT1						**(b)** QFT2						**(c)** QFT3				
		ϵ_{1R}	ϵ_{1I}	ϵ_{2R}	ϵ_{2I}			ϵ_{1R}	ϵ_{1I}	ϵ_{2R}	ϵ_{2I}			ϵ_1	ϵ_2
Theory	A	0.125	0.125	0.5	0.5	Theory	A	0.25	0.25	1	1	Theory	A	0.156	1
	α	1	1	1	1		α	1	1	1	1		α	1	1
Fit	A	0.130	0.133	0.144	0.145	Fit	A	0.282	0.284	0.768	0.784	Fit	A	0.369	2.72
	α	1	1	1.5	1.5		α	1	1	1.5	1.5		α	1	2

Table B.3
Comparison of the observed asymptotic behaviour of the L2 errors ϵ^2_{L2} with their theoretical bounds, for all algorithms under noiseless conditions. The algorithms scale as $\epsilon^2_{L2} = A(N/n_s)$ for large numbers of samples, with constants A reported in the table. The fits were estimated numerically by linear regression on the mean L2 errors for $n_s/N > 10^3$. The theoretical bounds were calculated as $\epsilon^2_{L2} = \sum_i \epsilon_i$.

	QFT1	QFT2	QFT3
Theory	5	7.5	3.313
Fit	0.931	1.556	1.060

interesting phenomenon is observed, where ϵ_{1R} and ϵ_{1I} initially match their bounds before deviating from them for larger values of $n_{s,m}/N$. Unpublished data suggests that the deviation is a 'cosmetic' effect due to the finite number of repetitions n_{rep} used in calculating the expectation values (the point at which deviation occurs is proportional to n_{rep}), and as such does not reflect algorithmic behaviour. Turning to QFT3, Fig. B.19 (a) shows that ϵ_1 scales as expected but with a constant about twice that predicted by theory. In contrast, ϵ_2 vastly outperforms its bound, with the asymptotic fit suggesting a $\mathcal{O}((N/n_{s,2})^2)$ scaling.[6] The stronger scaling of QFT3 compared with QFT1 and QFT2 can be explained by the fact that the phase ambiguity is lifted in a single efficient step. Indeed, circuit QFT-D can be understood as rotating the measurement basis to always measure (nearly) mutually orthogonal states.

B.2.2. Optimal sample distribution

An optimal distribution of samples between the steps can be estimated from the asymptotic fits. First, note that when the algorithm involves two amplitude-estimating steps, the theoretical study strongly supports using an equal number of samples in both steps. The same applies to the ambiguity-lifting steps. On this basis, n_s samples can be distributed as in Eq. (25), where $\lambda \in (0, 1)$ is a distribution variable that can be optimised. Along the same lines as for deriving the theoretical bounds, assume that the L2 error of an algorithm is equal to the sum of the errors introduced in each of its steps, $\epsilon^2_{L2} = \sum_i \epsilon_i$, and that these errors are defined by the asymptotic fits given in Table B.2. Then, after substituting the above expressions into ϵ^2_{L2} (which gives Eq. (26)), the optimal values of λ can be estimated by solving the non-linear equation

$$\frac{\partial \epsilon^2_{L2}}{\partial \lambda} = 0$$

numerically for each algorithm. The optimal values of λ and $(1 - \lambda)$ as functions of n_s/N are shown in Fig. B.20.

As could be expected, λ quickly increases with the number of samples, suggesting that for larger n_s/N, a larger share of samples should be allocated to the amplitude-estimating steps. The previous section gives possible explanations for the differences between the graphs.

[6] A slightly better fit was obtained for $\alpha = 1.85$ (with $A = 1.859$), but the smoothness of the data does not particularly support this scaling over the more intuitive $\alpha = 2$.

B.2.3. Noisy behaviour

Figs. B.17–B.19, (b)–(d), plot the error scalings in noisy environments. They show that the amplitude-estimating steps of all algorithms are very sensitive to noise. The errors introduced in these steps initially coincide with their noiseless scalings before deviating from them and becoming independent of $n_{s,m}/N$, with the deviation occurring earlier for higher noise levels. This is because errors are initially dominated by the statistical uncertainty associated with sampling, before noise takes over. Performance degrades for longer signals as the number of gates in the state preparation and QFT circuits is proportional to the number of qubits in the working register, and each gate is subject to noise.

Although the errors introduced in the ambiguity-lifting steps seem to undergo the same process, these steps are much more robust to noise. For a given $n_{s,m}/N$, deviation occurs at much higher noise levels, 1 to 2 orders of magnitude greater than in the amplitude-estimating steps. Alternatively, for a given p_d, deviation occurs at a much larger number of samples, and only really becomes noticeable (for the range investigated) for $p_d < 0.01\%$ for algorithms QFT1 and QFT2, and for $p_d < 0.001\%$ for QFT3. An interesting (and unexplained) phenomenon is that, for longer signals with $p_d < 0.01\%$, these errors remain constant for small values of $n_{s,m}/N$ before decreasing and then deviating.

Overall, QFT1 is the most robust to depolarising noise, followed by QFT3 and QFT2. Considering that the amplitude-estimating steps are the most sensitive, the better performance of QFT1 could be explained by the smaller number of qubits and gates in circuit QFT-A than in QFT-C. Unfortunately, increasing the number of samples to reduce the error only works up to a point, as noise inevitably becomes dominant.

B.2.4. L2 errors

Fig. B.21 shows that the L2 errors are well predicted by the ϵ_i fits over the entire range of n_s/N and for all algorithms under noiseless conditions (plots (a)–(c)). For larger numbers of samples, the errors introduced in the amplitude-estimating steps are dominant and ϵ^2_{L2} scales linearly with these errors. The ϵ_i fits coincide with the L2 errors (and with the L2 fits). For smaller values of n_s/N, errors introduced in the ambiguity-lifting steps become comparable to those introduced in the amplitude-estimating steps. Consequently, the L2 errors scale non-linearly, which is fairly well captured by the ϵ_i fits.

It is concluded that the L2 error can be well predicted by the asymptotic fits of the errors introduced in each step of the algorithms. This should hold for any sample distribution. However, caution is advised when extrapolating these results due to the limited range of n_s/N studied and the intrinsic uncertainty in the errors for which the fits were established.

As for the algorithmic behaviour under noisy conditions (plots (d)–(l)), the L2 errors follow their noiseless scalings until noise takes over, which makes ϵ^2_{L2} deviate from them. This reproduces the behaviour of the ϵ_i errors. The L2 error is mainly dominated by the noise introduced in the amplitude-estimating steps, except at high noise levels and for longer signals where the noise introduced in the phase-estimating steps predominates at times. This explains why some errors remain flat before decreasing and deviating, as observed previously. The difference is that the deviation is now quicker due to errors in the amplitude-estimating steps taking over again. QFT1 is confirmed as the most robust algorithm to depolarising noise, followed by QFT3 and QFT2.

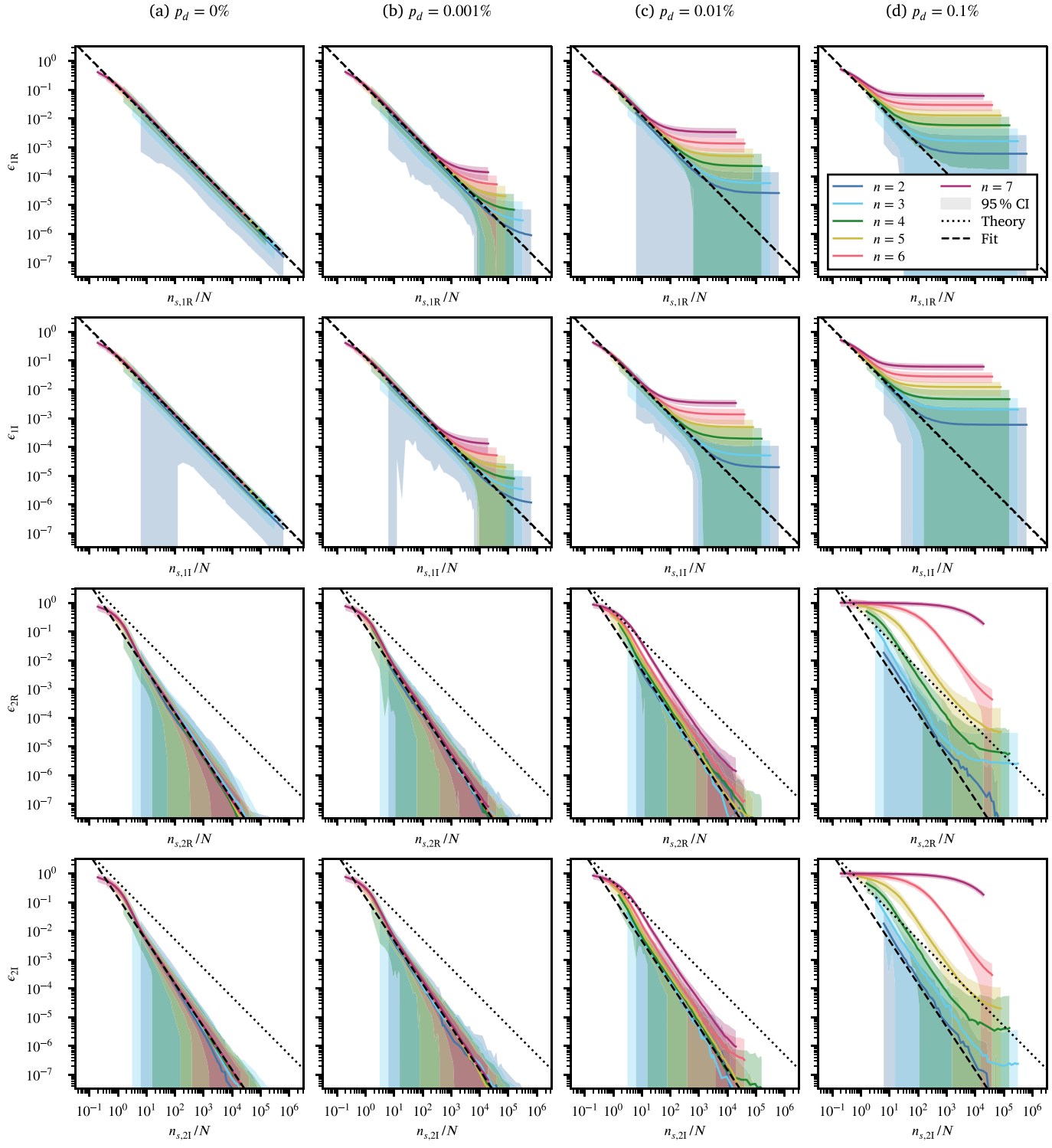


Fig. B.17. Errors introduced in the different steps of algorithm QFT1. Steps 1.1R, 1.1I, 1.2R and 1.2I introduce errors ϵ_{1R}, ϵ_{1I}, ϵ_{2R} and ϵ_{2I}, respectively. Each column (marked (a), (b), (c), (d)) shows the errors for a given noise level p_d whilst each row focuses on a single error. For each combination of the number of qubits n and the number of samples $n_{s,m}$ used in step m, $n_{sig} = 100$ complex signals of length $N = 2^n$ were generated randomly following a uniform distribution centred around zero and normalised. For each signal, the algorithm was run n_{rep} times to estimate the expectation values involved in the error definitions, resulting in a single value per error ϵ_i. The number of repetitions was $n_{rep} = 1000$ for the noiseless simulations and $n_{rep} = 100$ for the noisy ones. The means and 95% confidence intervals (CI) of the n_{sig} errors are plotted as functions of $n_{s,m}/N$ and can be compared with their theoretical bounds and asymptotic fits, detailed in Table B.2.

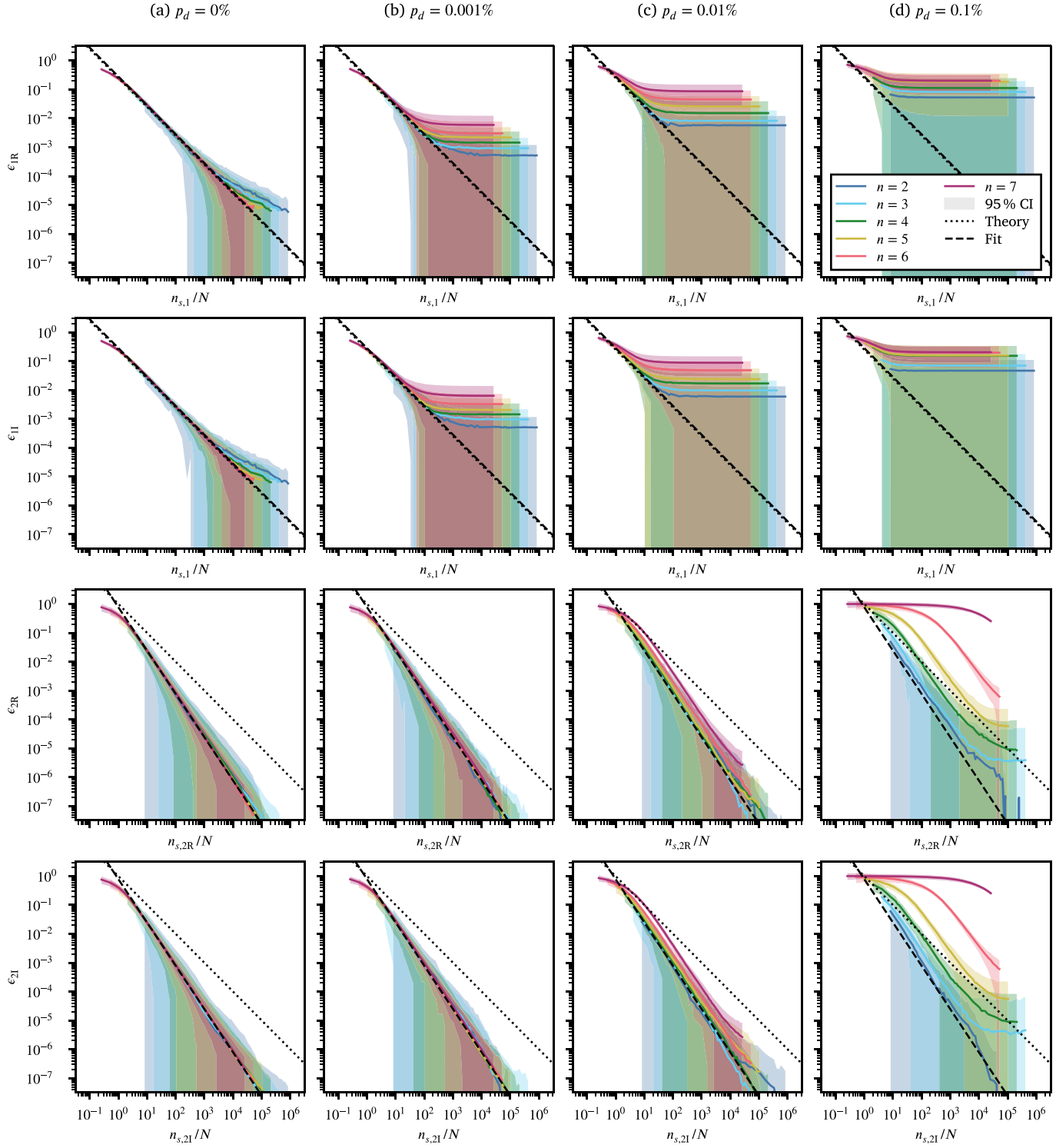


Fig. B.18. Errors introduced in the different steps of algorithm QFT2. Step 2.1 introduces errors ϵ_{1R} and ϵ_{1I}, while steps 1.2R and 1.2I introduce errors ϵ_{2R} and ϵ_{2I}, respectively. Each column (marked (a), (b), (c), (d)) shows the errors for a given noise level p_d whilst each row focuses on a single error. For each combination of the number of qubits n and the number of samples $n_{s,m}$ used in step m, $n_{sig} = 100$ complex signals of length $N = 2^n$ were generated randomly following a uniform distribution centred around zero and normalised. For each signal, the algorithm was run n_{rep} times to estimate the expectation values involved in the error definitions, resulting in a single value per error ϵ_i. The number of repetitions was $n_{rep} = 1000$ for the noiseless simulations and $n_{rep} = 100$ for the noisy ones. The means and 95% confidence intervals (CI) of the n_{sig} errors are plotted as functions of $n_{s,m}/N$ and can be compared with their theoretical bounds and asymptotic fits, detailed in Table B.2.

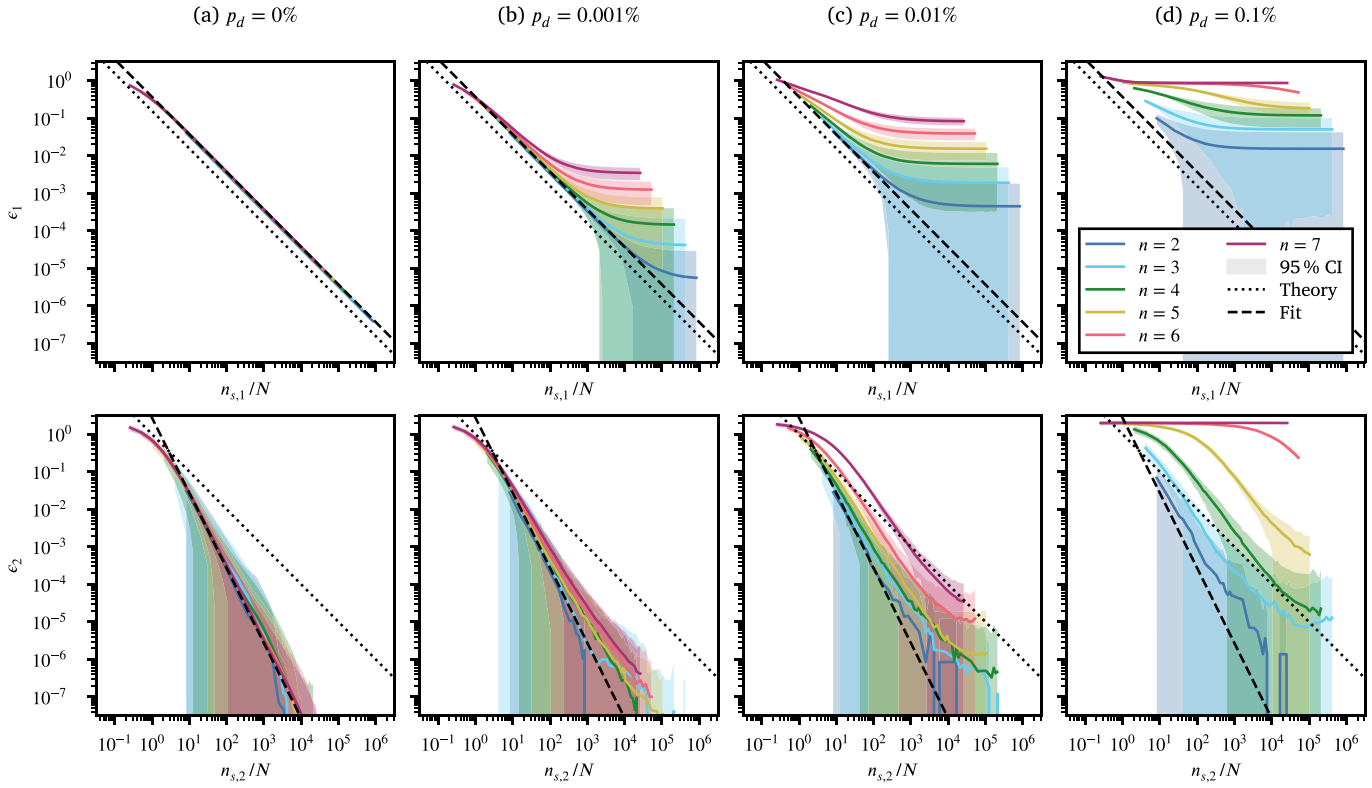


Fig. B.19. Errors introduced in the different steps of algorithm QFT3. Steps 3.1C and 3.1S together introduce error ϵ_1, while step 3.2 introduces error ϵ_2. Each column (marked (a), (b), (c), (d)) shows the errors for a given noise level p_d whilst each row focuses on a single error. Note that $n_{s,1} = n_{s,1C} = n_{s,1S}$. For each combination of the number of qubits n and the number of samples $n_{s,m}$ used in step m, $n_{sig} = 100$ complex signals of length $N = 2^n$ were generated randomly following a uniform distribution centred around zero and normalised. For each signal, the algorithm was run n_{rep} times to estimate the expectation values involved in the error definitions, resulting in a single value per error ϵ_i. The number of repetitions was $n_{rep} = 1000$ for the noiseless simulations and $n_{rep} = 100$ for the noisy ones. The means and 95% confidence intervals (CI) of the n_{sig} errors are plotted as functions of $n_{s,m}/N$ and can be compared with their theoretical bounds and asymptotic fits, detailed in Table B.2.

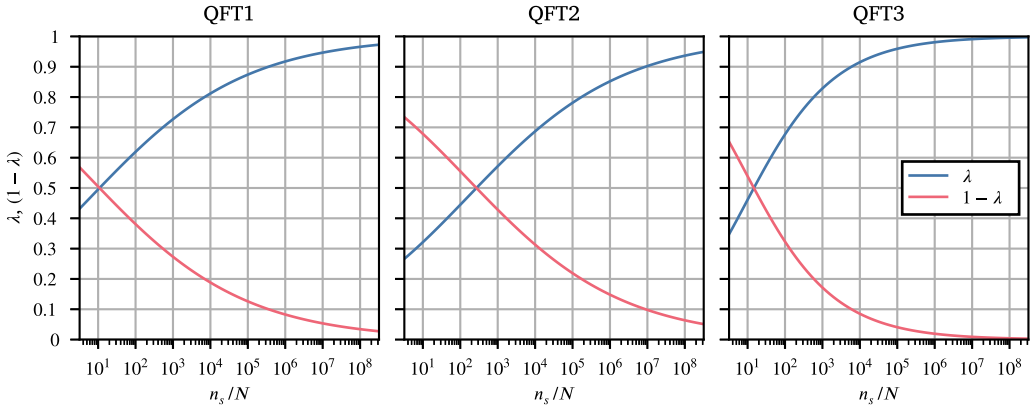


Fig. B.20. Estimated optimal values of λ and $(1-\lambda)$ for the distribution of n_s samples between the amplitude-estimating and ambiguity-lifting steps as functions of n_s/N, for all algorithms, with N the signal length. 'Amplitude-estimating steps' refers to steps 1.1R, 1.1I, 2.1, 3.1C and 3.1S, while 'ambiguity-lifting steps' refers to steps 1.2R, 1.2I, 2.2R, 2.2I and 3.2. Samples were distributed as $n_{s,1R} = n_{s,1I} = \lambda n_s/2$ and $n_{s,2R} = n_{s,2I} = (1-\lambda)n_s/2$ for QFT1, $n_{s,1} = \lambda n_s$ and $n_{s,2R} = n_{s,2I} = (1-\lambda)n_s/2$ for QFT2, and $n_{s,1S} = n_{s,1C} = \lambda n_s/2$ and $n_{s,2} = (1-\lambda)n_s$ for QFT3. The optimal values of λ were calculated by minimising the estimated L2 error $\epsilon_{L2}^2 = \sum_i \epsilon_i = f(n_s/N, \lambda)$ with respect to λ, using the ϵ_i fits given in Table B.2.

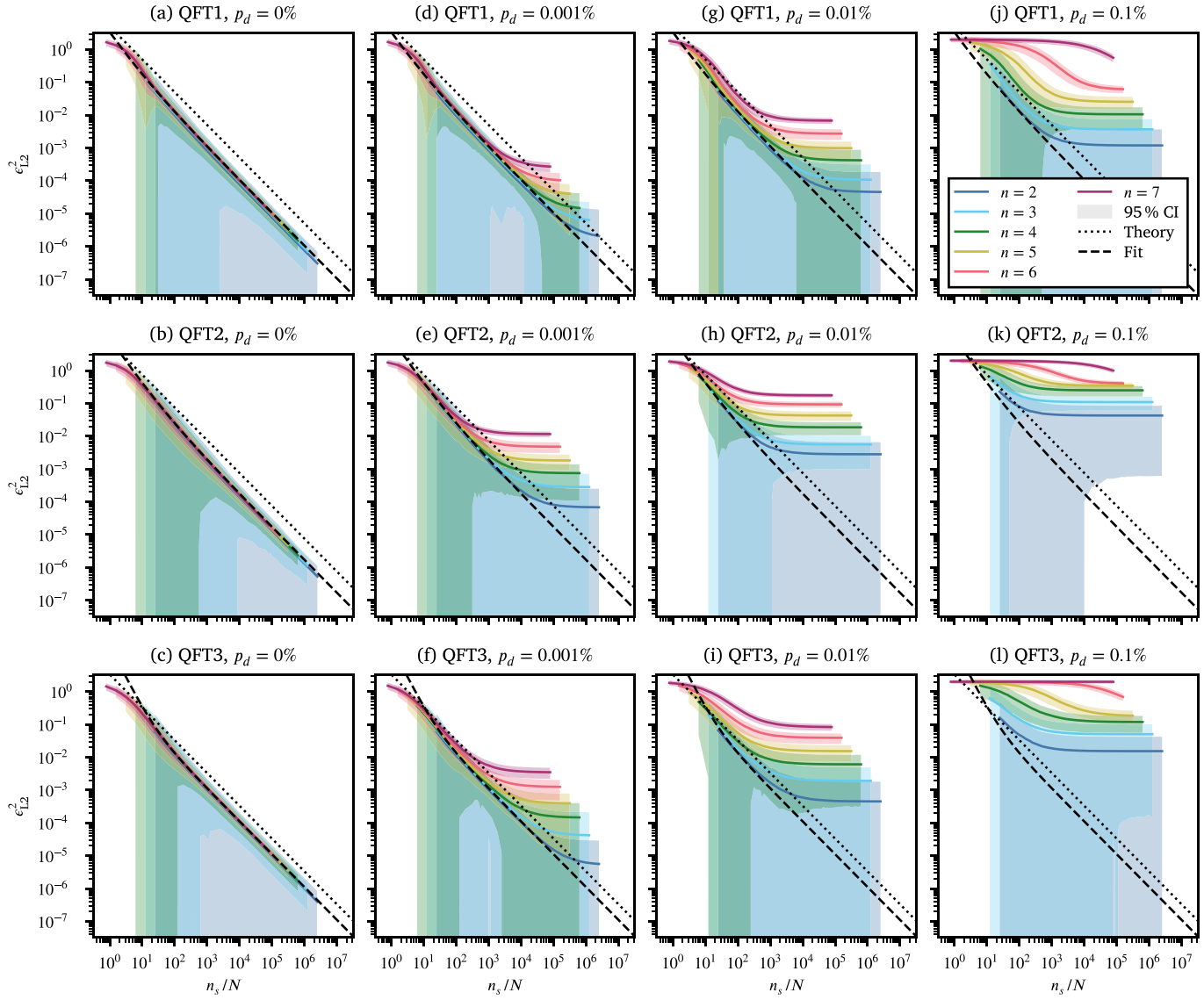


Fig. B.21. L2 errors of algorithms QFT1, QFT2 and QFT3 under noiseless and noisy conditions. Each column shows the errors for a given noise level p_d whilst each row focuses on the error of a single algorithm. For each combination of the number of qubits n and the total number of samples n_s, $n_{sig} = 100$ complex signals of length $N = 2^n$ were generated randomly following a uniform distribution centred around zero and normalised. For each signal, the algorithm was run n_{rep} times. The number of repetitions was $n_{rep} = 1000$ for the noiseless simulations and $n_{rep} = 100$ for the noisy ones. The means and 95% confidence intervals (CI) of the n_{sig} errors are plotted as functions of n_s/N and can be compared with their theoretical bounds and ϵ_i fits. These were calculated as $\epsilon^2_{L2} = \sum_i \epsilon_i$ from the bounds and fits given in Table B.2.

Data availability

Part of the numerical codes and data used in this study are available at http://dx.doi.org/10.17632/m4wxpzhdz5.1. The remaining data are available from the corresponding author, F.E., upon reasonable request.

References

[1] Castelvecchi D. IBM releases first-ever 1,000-qubit quantum chip. Nature 2023;624:238. http://dx.doi.org/10.1038/d41586-023-03854-1.

[2] Preskill J. Quantum computing in the NISQ era and beyond. Quantum 2018;2:79. http://dx.doi.org/10.22331/q-2018-08-06-79.

[3] Kim Y, Eddins A, Anand S, Wei KX, Van Den Berg E, Rosenblatt S, Nayfeh H, Wu Y, Zaletel M, Temme K, Kandala A. Evidence for the utility of quantum computing before fault tolerance. Nature 2023;618:500–5. http://dx.doi.org/10.1038/s41586-023-06096-3.

[4] von Karman Institute for Fluid Dynamics. In: Eulitz F, Munters W, editors. Introduction to quantum computing in fluid dynamics. STO-EN-AVT-377 lecture series, Sint-Genesius-Rode, Belgium: NATO Science and Technology Organization; 2023, http://dx.doi.org/10.14339/STO-EN-AVT-377.

[5] Montanaro A, Pallister S. Quantum algorithms and the finite element method. Phys Rev A 2015;93:032324. http://dx.doi.org/10.1103/physreva.93.032324.

[6] Ali M, Kabel M. Performance study of variational quantum algorithms for solving the Poisson equation on a quantum computer. Phys Rev Appl 2023;20:014054. http://dx.doi.org/10.1103/PhysRevApplied.20.014054.

[7] Chen Z-Y, Xue C, Chen S-M, Lu B-H, Wu Y-C, Ding J-C, Huang S-H, Guo G-P. Quantum approach to accelerate finite volume method on steady computational fluid dynamics problems. Quantum Inf Process 2022;21. http://dx.doi.org/10.1007/s11128-022-03478-w.

[8] Schalkers MA, Möller M. Efficient and fail-safe quantum algorithm for the transport equation. J Comput Phys 2024;502. http://dx.doi.org/10.1016/j.jcp.2024.112816.

[9] Kyriienko O, Paine AE, Elfving VE. Solving nonlinear differential equations with differentiable quantum circuits. Phys Rev A 2021;103:052416. http://dx.doi.org/10.1103/PhysRevA.103.052416.

[10] Pfeffer P, Heyder F, Schumacher J. Reduced-order modeling of two-dimensional turbulent Rayleigh-Bénard flow by hybrid quantum-classical reservoir computing. Phys Rev Res 2023;5:043242. http://dx.doi.org/10.1103/PhysRevResearch.5.043242.
[11] Sanavio C, Scatamacchia R, De Falco C, Succi S. Three Carleman routes to the quantum simulation of classical fluids. Phys Fluids 2024;36:057143. http://dx.doi.org/10.1063/5.0204955.
[12] Steijl R, Barakos GN. Parallel evaluation of quantum algorithms for computational fluid dynamics. Comput & Fluids 2018;173:22–8. http://dx.doi.org/10.1016/j.compfluid.2018.03.080.
[13] Hall KC, Thomas JP, Clark WS. Computation of unsteady nonlinear flows in cascades using a harmonic balance technique. AIAA J 2002;40:879–86. http://dx.doi.org/10.2514/2.1754.
[14] He L. Fourier methods for turbomachinery applications. Prog Aerosp Sci 2010;46:329–41. http://dx.doi.org/10.1016/j.paerosci.2010.04.001.
[15] Hall KC, Ekici K, Thomas JP, Dowell EH. Harmonic balance methods applied to computational fluid dynamics problems. Int J Comput Fluid Dyn 2013;27:52–67. http://dx.doi.org/10.1080/10618562.2012.742512.
[16] Junge L, Frey C, Ashcroft G, Kügeler E. A new harmonic balance approach using multidimensional time. J Eng Gas Turbines Power 2021;143:081007. http://dx.doi.org/10.1115/1.4049698.
[17] Wang F, di Mare L. Favre-averaged nonlinear harmonic method for compressible periodic flows. AIAA J 2019;57:1133–42. http://dx.doi.org/10.2514/1.J057523.
[18] IBM Quantum. Qiskit. 2024, URL https://www.ibm.com/quantum/qiskit.
[19] LeVeque RJ. Numerical methods for conservation laws. Basel, Switzerland: Birkhäuser Basel; 1992, http://dx.doi.org/10.1007/978-3-0348-8629-1.
[20] Nielsen MA, Chuang IL. Quantum computation and quantum information. Cambridge, UK: Cambridge University Press; 2010, http://dx.doi.org/10.1017/CBO9780511976667.
[21] IBM Quantum. Native gates and operations. 2024, URL https://docs.quantum.ibm.com/run/native-gates.
[22] Bruß D, Macchiavello C. Optimal state estimation for d-dimensional quantum systems. Phys Lett A 1999;253:249–51. http://dx.doi.org/10.1016/S0375-9601(99)00099-7.
[23] Shende VV, Bullock SS, Markov IL. Synthesis of quantum logic circuits. IEEE Trans Comput-Aided Des Integr Circuits Syst 2006;25:1000–10. http://dx.doi.org/10.1109/TCAD.2005.855930.
[24] Bravyi S, Cross AW, Gambetta JM, Maslov D, Rall P, Yoder TJ. High-threshold and low-overhead fault-tolerant quantum memory. Nature 2024;627:778–82. http://dx.doi.org/10.1038/s41586-024-07107-7.