

A Quantum Spectral Method for Non-Periodic Boundary Value Problems

Eky Febrianto^a, Yiren Wang^b, Burigede Liu^b, Michael Ortiz^c, Fehmi Cirak^b

^aGlasgow Computational Engineering Centre, James Watt School of Engineering, University of Glasgow, Glasgow, G12 8QQ, UK

^bDepartment of Engineering, University of Cambridge, Cambridge, CB2 1PZ, UK

^cDivision of Engineering and Applied Science, California Institute of Technology, Pasadena, CA 91125, USA

Abstract

Quantum computing holds the promise of solving computational mechanics problems in polylogarithmic time, meaning computational time scales as $O((\log N)^c)$, where N is the problem size and c a constant. We propose a quantum spectral method with polylogarithmic complexity for solving non-periodic boundary value problems with arbitrary Dirichlet boundary conditions. Our method extends the recently proposed approach by Liu et al. (2025), in which periodic problems are discretised using truncated Fourier series. In such spectral methods, the discretisation of boundary value problems with constant coefficients leads to a set of algebraic equations in the Fourier space. We implement the respective diagonal solution operator by first approximating it with a polynomial and then quantum encoding the polynomial. The mapping between the physical and Fourier spaces is accomplished using the quantum Fourier transform (QFT). To impose zero Dirichlet boundary conditions, we double the domain size and reflect all physical fields antisymmetrically. The respective reflection matrix defines the quantum sine transform (QST) by pre- and post-multiplying with the QFT. For non-zero Dirichlet boundary conditions, the solution is decomposed into a boundary-conforming and a homogeneous part. The homogeneous part is determined by solving a problem with a suitably modified forcing vector. We illustrate the basic approach with a Dirichlet-Poisson problem and demonstrate its generality by applying it to a fractional stochastic PDE for modelling spatial random fields. We discuss the circuit implementation of the proposed approach and provide numerical evidence confirming its polylogarithmic complexity.

Keywords: Quantum computing, Spectral methods, PDEs, fractional stochastic PDEs, Quantum Fourier transform

1. Introduction

The solution of boundary value problems from computational mechanics is one of the most promising application areas of future quantum computing [1]. The key appeal of quantum computing is that problems can be solved in $O(\log^c N)$ time, instead of $O(N^c)$ in classical computing, where N is the problem size and $c > 1$ a constant. In quantum computing, the N -dimensional forcing and solution vectors of a discretised boundary value problem are encoded as the complex-valued state vector of a system with $n = \log_2 N$ quantum particles, or qubits. The state vector of each qubit is of dimension two. The solution of the problem is formulated as the mapping of the forcing state vector to the solution state vector using a suitably constructed length and angle preserving unitary matrix, encoding the discretised solution operator. Quantum programming concerns the assembly of the problem-dependent unitary matrix from simpler elementary unitary matrices provided by the quantum hardware, considering that the set of unitary matrices is closed under multiplication [2, 3]. The elementary unitary matrices, or gates, operate on a single qubit or at most two qubits at a time. Informally, quantum speed-up is achieved by designing algorithms that have a polynomial number of one or two-qubit gates operating on the n qubits while the vectors and matrices operated on live in the Hilbert space $\mathbb{C}^N$ of dimension $N = 2^n$. However, this exponentially compressed representation makes it very costly to address and modify selected coefficients of a vector or a matrix, or to implement constructs like the if-then-else commonly used in classical computing.

*Corresponding author

Email addresses: eky.febrianto@glasgow.ac.uk (Eky Febrianto), yw680@cam.ac.uk (Yiren Wang), b1377@cam.ac.uk (Burigede Liu), ortiz@aero.caltech.edu (Michael Ortiz), fc286@cam.ac.uk (Fehmi Cirak)

It is evident that quantum computing is fundamentally different from classical computing. Hence, the present finite element approaches, primarily based on unstructured meshes and locally supported basis functions, may not be well-suited for quantum computing. For unstructured problems, preparing a desired quantum state on a quantum computer usually incurs a cost of $O(N)$, invalidating any potential quantum advantage in subsequent processing of the data. For instance, in the case of state preparation, which is an extensively studied problem, encoding a classical N -dimensional vector has complexity $O(N)$ when no additional ancillary/helper/temporary qubits are used, see e.g. [4, 5]. The same applies to the quantum encoding of a matrix, which can be thought of as N vectors of dimension N or less, when the matrix is sparse. For instance, the computational cost reduces to $O(\log_2 N)$ when the problem can be formulated in a way that allows vectors or matrices to be approximated by a function [6]. Setting aside the cost of state preparation, there are several algorithms, most prominently, Quantum Singular Value Transformation (QSVT) [7] and HHL [8], for solving linear systems of equations with polylogarithmic complexity. The original HHL algorithm has a complexity $O(\kappa^2/\epsilon \log N)$ and QSVT has a complexity $O(\kappa \log(\kappa/\epsilon))$, where κ is the condition number and ϵ the accuracy of the solution; see also the recent comprehensive review on quantum linear system solvers [9]. These and other complexity estimates from quantum computing assume that all vectors and matrices can be accessed via an oracle, i.e. a subroutine. The complexity and implementation of the oracles are not taken into account. The quantum encoding of (sparse) matrices, and the oracles, is discussed in more recent works [10–13]. Regardless of the cost of quantum encoding, the available linear system solvers, including QSVT and HHL, either require a large number of quantum operations or demand a large number of ancilla qubits, i.e., qubits beyond those used for data storage. This is exacerbated by the fact that the qubits must be fault-tolerant, which typically requires redundancy and replaces one (logical) qubit with several physical qubits. Consequently, the resulting hardware requirements for practically relevant problem sizes exceed what is expected to be available in the near term.

We conjecture that the simple structure and straightforward implementation of spectral methods on uniform grids make them especially appealing for quantum computing; see [14, 15] for an introduction to spectral methods. The commonly used basis functions in spectral methods are the trigonometric Fourier basis functions for periodic problems and Chebyshev basis functions for non-periodic problems. Both kinds of basis functions have appealing orthogonality properties and are global in the sense that they are supported over the entire problem domain. The discretisation of spectral methods leads to well-structured vectors and matrices with repetitive entries, which are easier to quantum encode than their fully unstructured equivalents from other discretisation methods. In particular, the Fourier-based approaches lead in the Fourier space to diagonal matrices or block-diagonal matrices when the coefficients of the differential equation are constant. The diagonality is independent of the dimension of the problem domain and the order of the differential operators, whether integer or fractional. Indeed, as recently demonstrated in Liu et al. [1] for FFT-based homogenisation [16–18], it is possible to solve specific problems from solid mechanics using a Fourier-based methodology with complexity $O(\log^c N)$. Recently, the feasibility of spectral methods on currently available hardware has also been demonstrated in [19] for a prototypical one-dimensional wave propagation problem. There are numerous works in quantum computing literature on spectral-like methods based on finite differences on structured grids [20, 21], Fourier basis functions [22], and Chebyshev basis functions [23]. Furthermore, the periodic/circulant system matrices in finite differences can be diagonalised using a discrete Fourier transform (DFT), or its equivalent, the quantum Fourier transform (QFT), leading to a spectral-like approach [24, Ch. 5]. The mentioned works mainly differ in terms of the problems investigated and in how the resulting systems of equations with diagonal or block-diagonal matrices are solved.

In this work, we introduce a novel quantum spectral solver for boundary value problems with arbitrary Dirichlet boundary conditions on axis-aligned d -dimensional hyperrectangles. We illustrate the basic approach with a Dirichlet-Poisson problem and demonstrate its generality by applying it to a fractional stochastic PDE used in modelling spatial random fields [25–28]. Our method builds on the recent foundational work by Liu et. al [1] on solving periodic Poisson problems in homogenisation. In [1], the periodic fields are discretised using a truncated Fourier basis, and the corresponding coefficients are determined by solving independent scalar equations in the Fourier space. The mapping between physical and Fourier spaces is carried out using the quantum Fourier transform (QFT), which has complexity $O(\log^2 N)$, in contrast to the classical DFT with complexity $O(N \log N)$. Scalar division in Fourier space is implemented via polynomial approximation of a diagonal matrix, which can be efficiently quantum encoded. The specific polynomial encoding technique we employ is the multivariate extension of the univariate scheme introduced in [29]. We extend the approach of Liu et al. [1] to non-periodic problems by doubling the physical domain size and antisymmetrically reflecting the source term. The new source and solution state vectors require only d additional

qubits compared to those for the original domain, where d is the spatial dimension. The application of the QFT to the extended source state vector cancels all cosine terms in the Fourier series, leaving only boundary conforming sine terms. A unitary reflection matrix is introduced to define the quantum sine transform (QST), the analogue of the discrete sine transform (DST), by pre- and post-multiplying the QFT matrix with the reflection unitary [30, 31]. Our quantum implementation of the reflection matrix follows [32]. We consider different implementations of its components, specifically the forward shift operator [11, 33], to reduce gate counts. In summary, replacing the QFT in the periodic PDE solver with DST yields a solver for homogeneous Dirichlet problems. To consider non-homogeneous boundary conditions, we use the superposition principle and decompose the solution into a boundary conforming part and a homogeneous part. The boundary conforming part is a sufficiently smooth function that can be chosen freely, while the homogeneous part is obtained by solving the homogeneous PDE with a modified source vector.

2. Classical Fourier spectral method

In this Section, we provide a brief review of the classical Fourier spectral method for solving partial differential equations and the treatment of homogeneous and inhomogeneous Dirichlet boundary conditions. For clarity, we consider a one-dimensional Poisson-Dirichlet problem with homogeneous boundary conditions, followed by its generalisation to other differential equations. In particular, we study a fractional differential equation motivated by the description of random fields. Finally, we introduce the treatment of inhomogeneous boundary conditions.

2.1. Poisson-Dirichlet problem

We consider on the domain $\Omega = (0, L) \subset \mathbb{R}^1$ the Poisson equation

$$-\frac{d^2 u(x)}{dx^2} = f(x), \quad \forall x \in \Omega, \quad (1)$$

with the homogeneous Dirichlet boundary conditions

$$u(0) = u(L) = 0. \quad (2)$$

We extend the source and solution fields, $f(x)$ and $u(x)$, to the domain $\Omega_E = (0, 2L)$ as follows

$$f_E(x) = \begin{cases} 0 & \text{if } x = 0, L, 2L, \\ f(x) & \text{if } x \in (0, L), \\ -f(2L - x) & \text{if } x \in (L, 2L). \end{cases} \quad (3a)$$

$$u_E(x) = \begin{cases} 0 & \text{if } x = 0, L, 2L, \\ u(x) & \text{if } x \in (0, L), \\ -u(2L - x) & \text{if } x \in (L, 2L); \end{cases} \quad (3b)$$

see Figure 1. It will be seen that this antisymmetric extension ensures that $u(x)$ satisfies the Dirichlet boundary conditions, making it compatible with a Fourier sine series representation. Note that the symmetry properties of the solution $u(x)$ depend on the order of the differential equation: for even-order differential equations, the source term $f(x)$ and the solution $u(x)$ share the same symmetry; for odd-order differential equations, the symmetry of $u(x)$ and $f(x)$ are opposite (i.e., one is symmetric while the other is antisymmetric).

On the extended domain Ω_E , the solution and source are approximated using truncated complex Fourier series

$$u_E(x) \approx u_E^h(x) = \frac{1}{\sqrt{N}} \sum_{k=-N/2}^{N/2-1} \hat{u}_{E,k} e^{i \frac{\pi k x}{L}}, \quad f_E(x) \approx f_E^h(x) = \frac{1}{\sqrt{N}} \sum_{k=-N/2}^{N/2-1} \hat{f}_{E,k} e^{i \frac{\pi k x}{L}}, \quad (4)$$

where $\hat{u}_{E,k}$ and $\hat{f}_{E,k}$ are Fourier coefficients, and $i^2 = -1$. To render the index k non-negative, we introduce the relabelling

$$r(k) = \begin{cases} k & 0 \leq k < N/2 \\ k - N & N/2 \leq k < N; \end{cases} \quad (5)$$

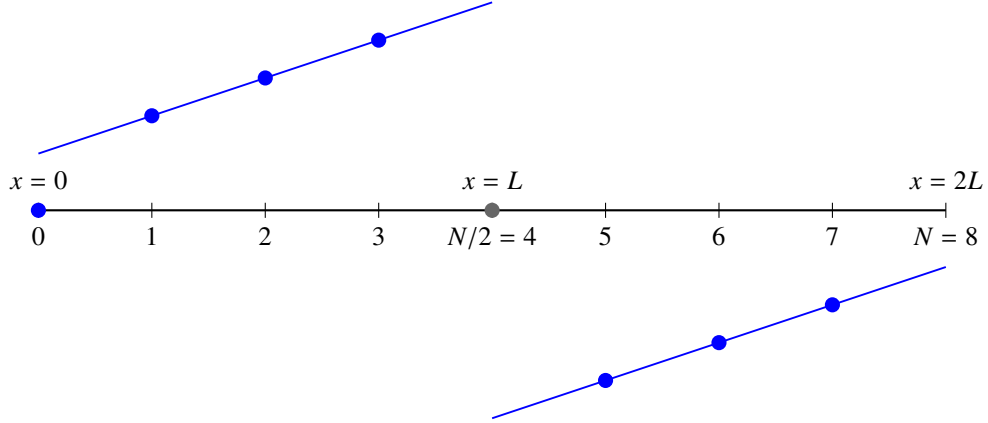


Figure 1: The extended domain $\Omega_E = (0, 2L)$ and its discretisation with $N = 8$ cells and $N = 8$ grid points. The original problem domain is $\Omega = (0, L)$. A sample reflected source term $f_E(x)$ is depicted in blue, and the components of the respective force vector $\mathbf{f}$ are depicted as circles. The extended domain problem is $2L$ periodic.

so that

$$u_E^h(x) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \hat{u}_{E,k} e^{i\xi_k x}, \quad f_E^h(x) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \hat{f}_{E,k} e^{i\xi_k x}, \quad \text{where } \xi_k = \frac{\pi r(k)}{L}. \quad (6)$$

Given the orthogonality of the basis functions, substituting the spectral discretisation into the governing equation (28) leads to the algebraic relation

$$\hat{u}_{E,k} = \frac{\hat{f}_{E,k}}{\xi_k^2}, \quad \forall k \neq 0. \quad (7)$$

The wavenumber $k = 0$ can be omitted because $f_E(x)$ has zero mean, and therefore $\hat{f}_{E,0} = 0$. Introducing the Fourier domain solution in the first equation in (6) yields

$$u_E^h(x) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \frac{\hat{f}_{E,k}}{\xi_k^2} e^{i\xi_k x}. \quad (8)$$

This solution is inherently periodic and exhibits antisymmetry about the centre of the domain.

2.2. Solution by discrete Fourier transform

We compute the Fourier coefficients $\hat{f}_{E,k}$ corresponding to the source field approximation $f_E^h(x)$ in (4) using the discrete Fourier transform (DFT). To achieve this, we discretise the extended domain Ω_E with N grid points and N grid cells, as illustrated in Figure 1. Thus, the physical domain Ω contains $N/2$ grid points and $N/2$ grid cells. The grid point coordinates are defined as

$$x_k = k \frac{2L}{N}, \quad k = 0, 1, \dots, N-1. \quad (9)$$

The source field values at the grid points are collected into the antisymmetrically extended vector

$$\mathbf{f}_E = \left(0 \quad f(x_1) \quad \dots \quad f(x_{N/2-1}) \quad 0 \quad -f(x_{N/2-1}) \quad \dots \quad -f(x_1) \right)^T. \quad (10)$$

The components corresponding to the first and centre grid points are zero as required by (3a). We now introduce the antisymmetric extension matrix $\mathbf{R} \in \mathbb{R}^{N \times N/2}$ to determine the reflected source vector $\mathbf{f}_E \in \mathbb{R}^N$ from the source vector $\mathbf{f} \in \mathbb{R}^{N/2}$ containing grid point values only associated to the physical domain, i.e.,

$$\mathbf{f}_E = \mathbf{R} \mathbf{f}, \quad (11)$$

where

$$\mathbf{f} = (0, f(x_1), \dots, f(x_{N/2-1}))^\top, \quad (12)$$

and

$$\mathbf{R} = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & -1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & -1 & \dots & 0 \\ 0 & -1 & 0 & \dots & 0 \end{pmatrix}. \quad (13)$$

The so-defined reflection matrix $\mathbf{R}$ ensures that the source vector $\mathbf{f}_E$ is antisymmetric.

The DFT of the source vector $\mathbf{f}_E$ is defined as

$$\hat{\mathbf{f}}_E = \mathbf{F}_N \mathbf{f}_E, \quad (14)$$

with the Fourier matrix

$$\mathbf{F}_N = \frac{1}{\sqrt{N}} (\omega_N^{jk})_{j,k=0,\dots,N-1} \quad (15)$$

and the N -th root of unity $\omega_N = e^{i\frac{2\pi}{N}}$. The Fourier matrix is unitary, i.e. $\mathbf{F}_N \mathbf{F}_N^\dagger = \mathbf{I}$, where the dagger $\dagger$ denotes the conjugate transpose, and can be written as

$$\mathbf{F}_N = \frac{1}{\sqrt{N}} \begin{pmatrix} 1 & 1 & 1 & 1 & \dots & 1 & \dots & 1 & 1 & 1 \\ 1 & \omega_N & \omega_N^2 & \omega_N^3 & \dots & \omega_N^{N/2} & \dots & \omega_N^{-3} & \omega_N^{-2} & \omega_N^{-1} \\ 1 & \omega_N^2 & \omega_N^4 & \omega_N^{-2} & \dots & 1 & \dots & \omega_N^2 & \omega_N^4 & \omega_N^{-2} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 1 & \omega_N^{N/2} & 1 & \omega_N^{N/2} & \dots & \omega_N^{N/2} & \dots & \omega_N^{N/2} & 1 & \omega_N^{N/2} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 1 & \omega_N^{-2} & \omega_N^4 & \omega_N^2 & \dots & 1 & \dots & \omega_N^{-2} & \omega_N^{-4} & \omega_N^2 \\ 1 & \omega_N^{-1} & \omega_N^{-2} & \omega_N^{-3} & \dots & \omega_N^{N/2} & \dots & \omega_N^3 & \omega_N^2 & \omega_N \end{pmatrix}, \quad (16)$$

taking into account $\omega_N^k = \omega_N^{(k \bmod N)}$ and $\omega_N^k = \omega_N^{-(N-k)}$. It is easy to verify that the product $\mathbf{F}_N \mathbf{R}$ yields a matrix with only sine terms with each row and column representing one of the Fourier basis functions. The cosine terms in the original $\mathbf{F}_N$ are eliminated by multiplication with the reflection matrix $\mathbf{R}$. The first $N/2$ rows and columns correspond to grid points within the domain of interest Ω .

With the help of definitions of $\mathbf{f}_E$ and DFT, (11) and (14), the solution (8) in the Fourier domain can be compactly written as

$$\mathbf{u}_E = \mathbf{F}_N^\dagger \mathbf{D}_E^{-1} \mathbf{F}_N \mathbf{R} \mathbf{f}, \quad (17)$$

where $\mathbf{D}_E \in \mathbb{R}^{N \times N}$ is a diagonal matrix with the entries

$$\mathbf{D}_E = \left(\frac{\pi}{L}\right)^2 \text{diag}(1 \quad 1 \quad 2^2 \quad 3^2 \quad \dots \quad (N/2)^2 \quad \dots \quad 3^2 \quad 2^2 \quad 1). \quad (18)$$

We can rewrite the solution equation such that

$$\mathbf{u} = \mathbf{R}^\top \mathbf{F}_N^\dagger \mathbf{R} \mathbf{D}^{-1} \mathbf{R}^\top \mathbf{F}_N \mathbf{R} \mathbf{f}, \quad (19)$$

where

$$\mathbf{D} = \left(\frac{\pi}{L}\right)^2 \text{diag}\left(1 \quad 1 \quad 2^2 \quad 3^2 \quad \dots \quad (N/2 - 1)^2\right). \quad (20)$$

In comparison to (17), this has the advantage that the size of the matrix $\mathbf{D} \in \mathbb{R}^{N/2 \times N/2}$ that must be quantum encoded is halved. Furthermore, we note that the expression $\mathbf{R}^\top \mathbf{F}_N \mathbf{R}$ is known as the discrete sine transform (DST) of type I [31].

2.3. Other boundary value problems

The introduced Fourier spectral method can easily be generalised to other boundary value problems and inhomogeneous Dirichlet boundary conditions. Because we use trigonometric sine functions as basis functions, the occurring derivatives in the differential equation must be all of even order (i.e., zeroth, second, fourth, ... order). We consider as an example problem on $\Omega = (0, L) \subset \mathbb{R}^1$ the fractional differential equation

$$\left(\kappa^2 - \frac{d^2}{dx^2}\right)^\beta u(x) = \frac{1}{\tau} f(x), \quad x \in \Omega, \quad (21)$$

with the homogeneous Dirichlet boundary conditions

$$u(0) = u(L) = 0, \quad (22)$$

and the three prescribed parameters $\kappa, \beta, \tau \in \mathbb{R}$.

As in the Poisson-Dirichlet problem, first the source and solution fields $f(x)$ and $u(x)$ are antisymmetrically extended to the larger domain $\Omega_E = (0, 2L)$. Subsequently, the obtained fields $f_E(x)$ and $u_E(x)$ are approximated using a truncated Fourier series. A straightforward derivation shows that the Fourier domain solution satisfies the algebraic equation

$$\hat{u}_{E,k} = \frac{\hat{f}_{E,k}}{\tau(\kappa^2 + \xi_k^2)^\beta} \quad \forall k \neq 0. \quad (23)$$

Hence, the solution of the fractional differential equation in the extended domain is given by

$$u_E^h(x) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \frac{\hat{f}_{E,k}}{\tau(\kappa^2 + \xi_k^2)^\beta} e^{i\xi_k x}. \quad (24)$$

Again, $\hat{u}_{E,k}$ and $\hat{f}_{E,k}$ coincide with the discrete Fourier transforms of $u_E(x)$ and $f_E(x)$ sampled over an equidistant grid. By retracing the same steps as in the Poisson-Dirichlet case, we obtain the discrete solution vector

$$\mathbf{u} = \mathbf{R}^\top \mathbf{F}^\dagger \mathbf{R} \mathbf{D}^{-1} \mathbf{R}^\top \mathbf{F} \mathbf{R} \mathbf{f}, \quad (25)$$

where

$$\mathbf{D} = \tau \text{diag}\left(\kappa^2 \quad \kappa^2 + \left(\frac{\pi}{L}\right)^2 \quad \kappa^2 + \left(\frac{2\pi}{L}\right)^2 \quad \kappa^2 + \left(\frac{3\pi}{L}\right)^2 \quad \dots \quad \kappa^2 + \left(\frac{(N/2-1)\pi}{L}\right)^2\right)^\beta. \quad (26)$$

2.4. Arbitrary Dirichlet boundary conditions

The considered boundary value problems are all linear, which allows us to use the superposition principle when applying inhomogeneous Dirichlet boundary conditions. The main idea is to choose first a sufficiently smooth function $g(x)$ that satisfies the prescribed Dirichlet boundary conditions, but otherwise has arbitrary values within the domain. For instance, in the case of the Poisson-Dirichlet problem, we have

$$\begin{aligned} -\frac{d^2 u(x)}{dx^2} &= f(x), \quad \forall x \in (0, L), \\ u(0) &= g(0), \quad u(L) = g(L). \end{aligned} \quad (27)$$

The function $g(x)$ can, for instance, be constructed by interpolating the prescribed Dirichlet boundary conditions using Lagrange basis functions. We can now define a solution field $v(x) = u(x) - g(x)$ that satisfies

$$\begin{aligned} -\frac{d^2 v(x)}{dx^2} &= f(x) + \frac{d^2 g(x)}{dx^2}, \quad \forall x \in (0, L), \\ v(0) &= 0, \quad v(L) = 0. \end{aligned} \quad (28)$$

Consequently, in the Fourier spectral method, arbitrary Dirichlet boundary conditions can be accommodated by suitably modifying the source field of a problem with homogeneous Dirichlet boundary conditions.

3. Quantum Fourier spectral method

We introduce next the quantum implementation of the Fourier spectral method for arbitrary Dirichlet boundary conditions. The forcing vector is encoded as the initial state of a quantum system, and the partial differential equation is solved by evolving the state of the quantum system. According to the postulates of quantum mechanics, the state vector of the quantum system must be of unit length and its evolution unitary. In this Section, we switch to Dirac notation, which is analogous to the standard vector notation, to distinguish between quantum mechanical and classical vectors. Introductions to Dirac notation can be found in standard texts on quantum computing [34–36] and earlier work [1].

3.1. Quantum Fourier transform (QFT)

In Dirac notation, the Fourier transform (14) is expressed as

$$|\widehat{f}\rangle = F_N |f\rangle, \quad (29)$$

where $|\widehat{f}\rangle, |f\rangle \in \mathbb{C}^N$ are the $N = 2^n$ dimensional state vectors of a quantum system with n qubits, before and after the application of the Fourier matrix $F_N \in \mathbb{C}^{N \times N}$. The two vectors in componentwise form are given by

$$|f\rangle = \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_{N-1} \end{pmatrix} = \sum_{k=0}^{N-1} f_k |k\rangle, \quad |\widehat{f}\rangle = \begin{pmatrix} \hat{f}_0 \\ \hat{f}_1 \\ \vdots \\ \hat{f}_{N-1} \end{pmatrix} = \sum_{k=0}^{N-1} \hat{f}_k |k\rangle, \quad (30)$$

where $|k\rangle$ with $k \in \{0, 1, \dots, N-1\}$ are the standard basis vectors, i.e.,

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}, \quad \dots, \quad |N-1\rangle = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}, \quad (31)$$

and f_k and $\hat{f}_k$ are the respective coefficients. In quantum computing, the standard basis is usually referred to as the computational basis and the coefficients as the amplitudes. The basis vectors can be indexed using binary numbers by noting that

$$k = k_0 2^{n-1} + \dots + k_{n-2} 2^1 + k_{n-1} 2^0 = \sum_{j=0}^{n-1} k_j 2^{n-1-j} \quad \text{and} \quad k_0, \dots, k_{n-2}, k_{n-1} \in \{0, 1\}, \quad (32)$$

and writing $|k\rangle \equiv |k_0 \dots k_{n-2} k_{n-1}\rangle \equiv |k_0\rangle \dots |k_{n-2}\rangle |k_{n-1}\rangle$.

The Fourier matrix is inherently unitary, i.e. $F_N F_N^\dagger = I$, given its row and column vectors are orthonormal. F_N is implemented by multiplicatively decomposing it into 2×2 and 4×4 unitary matrices, referred to as gates. It bears emphasis here that any product of unitary matrices is again a unitary matrix. In coming up with a suitable decomposition, it is convenient to consider the action F_N on a single basis vector $|k\rangle$, yielding the column k of F_N .

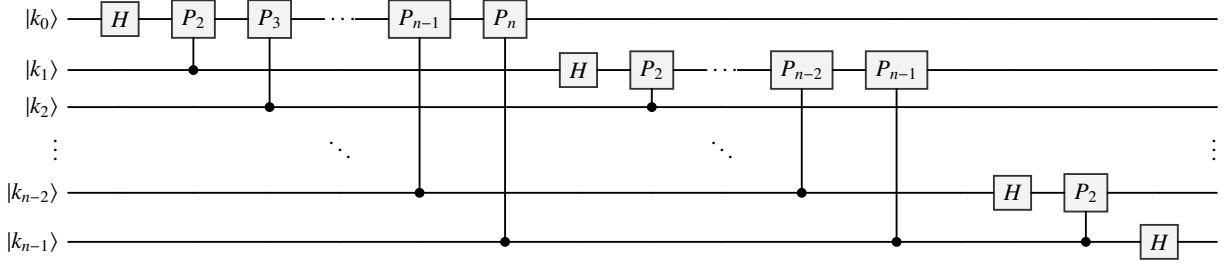


Figure 2: Quantum circuit for QFT. An input vector $\sum_{k=0}^{N-1} f_k |k\rangle$ is mapped to the output vector $\sum_{k=0}^{N-1} \hat{f}_k |k\rangle$ (pursuant to some relabelling of the components).

Subsequently, the derived decomposition can be applied to the entire vector $|f\rangle$ by making use of superposition. The key realisation in implementing F_N is that its row k can be expressed as the tensor product, i.e. Kronecker product, of n two-dimensional vectors,

$$\begin{aligned} \frac{1}{\sqrt{N}} \otimes_{j=0}^{n-1} (|0\rangle + \omega_N^{2^{n-1-j}k} |1\rangle) &= \frac{1}{\sqrt{N}} \left(\omega_N^{2^{n-1}k} \right) \otimes \cdots \otimes \left(\omega_N^{2^2k} \right) \otimes \left(\omega_N^{2^1k} \right) \otimes \left(\omega_N^{2^0k} \right) \\ &= \frac{1}{\sqrt{N}} \begin{pmatrix} 1 & \omega_N^k & \omega_N^{2k} & \omega_N^{3k} & \omega_N^{4k} & \cdots & \omega_N^{(2^n-1)k} \end{pmatrix}^\top \end{aligned} \quad (33)$$

cf. (16). Each two-dimensional vector represents the state vector of a single qubit. The first line in (33) is rewritten by expressing the index k as a binary, i.e.,

$$\frac{1}{\sqrt{N}} \left(e^{2\pi i (\frac{k_{n-1}}{2})} \right) \otimes \cdots \otimes \left(e^{2\pi i (\frac{k_2}{2} + \cdots + \frac{k_{n-1}}{2^{n-2}})} \right) \otimes \left(e^{2\pi i (\frac{k_1}{2} + \cdots + \frac{k_{n-1}}{2^{n-1}})} \right) \otimes \left(e^{2\pi i (\frac{k_0}{2} + \cdots + \frac{k_{n-1}}{2^n})} \right). \quad (34)$$

The identity $e^{2\pi i} \equiv 1$ and its periodicity are used in deriving this expression. Hence, the state vector of the entire system consists of the tensor-product of the states of n qubits and depends on the binary representation of the index k . We recall that the binary digits $k_0 \dots k_{n-3} k_{n-2} k_{n-1}$ encode the column index k of F_N . As depicted in the circuit diagram in Figure 2, the state vector (34) can be constructed by using only Hadamard and controlled phase gates defined as

$$H = \frac{1}{\sqrt{N}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad P_l = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i / 2^l} \end{pmatrix}. \quad (35)$$

As an example, consider the case $k = 0$ with the input state vector $|0 \dots 000\rangle$ yielding F_N 's first column

$$\frac{1}{\sqrt{N}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \otimes \cdots \otimes \begin{pmatrix} 1 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{N}} (1, 1, 1, 1, 1, \dots, 1)^\top. \quad (36)$$

This state is constructed by applying Hadamard gates to all qubits. None of the controlled phase gates are applied because the respective control qubits all have the value 0. On the other hand, when $k = 1$ with the input state vector $|0 \dots 001\rangle$ the state vector that must be constructed reads

$$\frac{1}{\sqrt{N}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \otimes \cdots \otimes \begin{pmatrix} 1 \\ e^{\frac{2\pi i}{2^{n-2}}} \end{pmatrix} \otimes \begin{pmatrix} 1 \\ e^{\frac{2\pi i}{2^{n-1}}} \end{pmatrix} \otimes \begin{pmatrix} 1 \\ e^{\frac{2\pi i}{2^n}} \end{pmatrix} = \frac{1}{\sqrt{N}} \left(1, e^{\frac{2\pi i}{2^n}}, e^{\frac{4\pi i}{2^n}}, e^{\frac{6\pi i}{2^n}}, e^{\frac{8\pi i}{2^n}}, \dots, e^{\frac{2(2^n-1)\pi i}{2^n}} \right)^\top. \quad (37)$$

This state is constructed by applying controlled phase gates $P_n, P_{n-1}, \dots, P_2$ gates conditioned on $k_{n-1} = 1$. In the circuit in Figure 2, the state (37) is first generated in reverse order and must be subsequently corrected using swap gates mapping $|k_{n-1} k_{n-2} k_{n-3} \dots k_0\rangle$ to $|k_0 \dots k_{n-3} k_{n-2} k_{n-1}\rangle$. These swap gates are omitted in Figure 2. The quantum circuit takes as an input the state vector $\sum_{k=0}^{N-1} f_k |k\rangle$ and returns as an output the state vector $\sum_{k=0}^{N-1} \hat{f}_k |k\rangle$. Hence, the entire quantum state is transformed in one fell swoop owing to quantum parallelism.

In passing, we note that the QFT algorithm has the complexity $O(n^2)$ where $n = \log N$ [34, Ch. 5]. In comparison, the complexity of fast Fourier transform (FFT) expressed in terms of n is $O(n2^n)$. That is, QFT is exponentially faster than FFT.

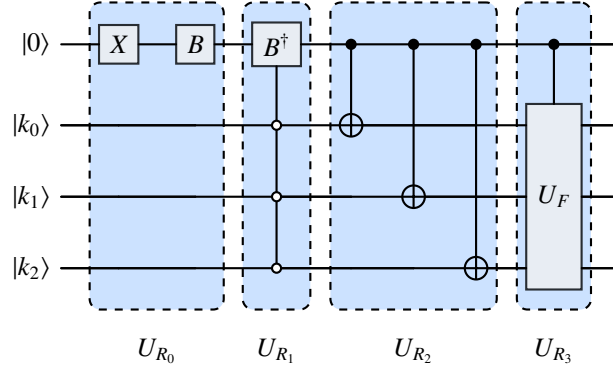


Figure 3: Quantum circuit for the reflection unitary $U_R = U_{R_3}U_{R_2}U_{R_1}U_{R_0}$. An input vector $\sum_{k=0}^7 f_k |0\rangle |k\rangle$ is mapped to the antisymmetrically extended output vector $\sum_{k=0}^{15} f_k |k\rangle$.

3.2. Unitary reflection matrix

We consider next the quantum implementation of the non-unitary reflection matrix $\mathbf{R} \in \mathbb{R}^{N/2 \times N}$ introduced in (13). To this end, the matrix $\mathbf{R}$ is embedded, or in quantum computing terminology block-encoded, in the unitary matrix

$$U_R = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & \sqrt{2} & 0 & 0 & \cdots & 0 \\ 0 & i & 0 & \cdots & 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & i & \cdots & 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & i & 0 & 0 & 0 & \cdots & 1 \\ \sqrt{2} & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & -i & 0 & 0 & 0 & \cdots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & -i & \cdots & 0 & 0 & 0 & 1 & \cdots & 0 \\ 0 & -i & 0 & \cdots & 0 & 0 & 1 & 0 & \cdots & 0 \end{pmatrix}. \quad (38)$$

Except for the first column, its right half is equal to $-i\mathbf{R}$ up to a scaling factor. The first column and the second half of U_R are chosen so that all rows and columns are mutually orthonormal, as necessary for unitarity. Although the column $N/2$ is different from $\mathbf{R}$'s first column, it is inconsequential for the final solution of a zero Dirichlet boundary value problem.

U_R is again implemented by decomposing it into elementary gates, i.e. 2×2 and 4×4 unitary matrices. One possible decomposition of U_R is shown in Figure 3; other decompositions are possible. For concreteness, we choose for the extended domain a discretisation with $N = 16$ gridpoints, i.e. four qubits. Consequently, the problem domain is discretised with 8 grid points and the respective field vector values are encoded as the state vector of the three qubits $|k_0 k_1 k_2\rangle$. The depicted decomposition can be understood as the product of the four unitaries U_{R_0} , U_{R_1} , U_{R_2} , U_{R_3} , each of which will be discussed in the following.

The leftmost unitary U_{R_0} in the circuit diagram is given by

$$U_{R_0} = (BX) \otimes I^{\otimes 3} = (HSX) \otimes I^{\otimes 3} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes I^{\otimes 3} = \frac{1}{\sqrt{2}} \begin{pmatrix} iI^{\otimes 3} & I^{\otimes 3} \\ -iI^{\otimes 3} & I^{\otimes 3} \end{pmatrix}, \quad (39)$$

where $I \in \mathbb{R}^{2 \times 2}$ and $I^{\otimes 3} \in \mathbb{R}^{8 \times 8}$ are identity matrices, and X , H and S are the Pauli X , Hadamard and phase gates which are available in most quantum computing platforms. The unitary $U_{R_0} \in \mathbb{C}^{16 \times 16}$ consists of the four blocks indicated in (39). Notice that the identity gates in (39) are, as usual, omitted in the circuit diagram.

The unitary U_{R_1} consists of the conjugate transpose $B^\dagger$ conditioned on the state $|k_0 k_1 k_2\rangle$. Specifically, $B^\dagger$ is only applied to the ancilla qubit when $|k_0 k_1 k_2\rangle \equiv |000\rangle$ as indicated by the empty circles. Consequently, it applies a unitary

transformation in the subspace spanned by the two basis vectors $|0\rangle|000\rangle$ and $|1\rangle|000\rangle$, else the identity gate I is applied. Based on the circuit diagram, we can write

$$U_{R_1} = B^\dagger \otimes (|0\rangle\langle 0|)^{\otimes 3} + \left(I \otimes \sum_{\substack{k_0=0, k_1=0, k_2=0 \\ k_0=k_1=k_2 \neq 0}}^1 |k_0\rangle\langle k_0| \otimes |k_1\rangle\langle k_1| \otimes |k_2\rangle\langle k_2| \right), \quad (40)$$

and, more explicitly

$$U_{R_1} = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & \cdots & 0 & \frac{1}{\sqrt{2}} & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 0 \\ \frac{-i}{\sqrt{2}} & 0 & \cdots & 0 & \frac{i}{\sqrt{2}} & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1 \end{pmatrix}. \quad (41)$$

The next unitary U_{R_2} is composed of three *CNOT* gates, which are applied only when the top (ancilla) qubit is in state $|1\rangle$, as indicated by the filled circles. If the ancilla qubit is in state $|0\rangle$ the identity gate I is applied. A single *CNOT* gate is a controlled X gate. Hence, U_{R_2} takes the form

$$U_{R_2} = |0\rangle\langle 0| \otimes I^{\otimes 3} + |1\rangle\langle 1| \otimes X^{\otimes 3} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}^{\otimes 3} + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}^{\otimes 3} = \begin{pmatrix} I^{\otimes 3} & 0 \\ 0 & X^{\otimes 3} \end{pmatrix}. \quad (42)$$

The multiplication of the unitaries $U_{R_1} U_{R_0}$ defined so far with U_{R_2} from the left has the effect of moving the diagonal entries in the two blocks on the lower half of $U_{R_1} U_{R_0}$ along the anti-diagonal. It bears emphasis that the order of the unitaries in the circuit diagram and matrix multiplication is reversed. The circuit diagram is read from left to right, whereas the matrix product is defined right to left.

The final unitary U_{R_3} in the circuit diagram is the controlled forward shift unitary, which applies a down shift by one to the entries in the lower half of $U_{R_2} U_{R_1} U_{R_0}$. It follows from the circuit diagram that

$$U_{R_3} = |0\rangle\langle 0| \otimes I^{\otimes 3} + |1\rangle\langle 1| \otimes U_F, \quad (43)$$

where the forward shift unitary is defined as $U_F: |k\rangle \mapsto |(k+1) \bmod 2^3\rangle$, i.e.,

$$U_F = \begin{pmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix}. \quad (44)$$

There are several possible ways to implement the unitary U_F . In Figure 4a an implementation of U_F based on a cascade of multi-controlled *CNOT*, i.e. multi-controlled X , gates is shown. On a quantum computer, the multi-controlled *CNOT* gates are implemented by decomposing them into standard *CNOT* gates. This process leads to a significant increase in the number of gates and possibly the introduction of ancilla qubits [3], [34, Ch. 4.3]. The proliferation of *CNOT* gates can be significantly reduced by switching to the circuit depicted in Figure 4b, which is based on quantum ripple-carry addition [11, 33]. The ripple-carry incrementer uses *CNOT* and Toffoli gates, that is, X -gates with one or two controls, and ancillas for intermediate carry propagation.

As detailed in Appendix A, any quantum circuit can be expressed using only the two-qubit *CNOT* and the single-qubit U_3 gates. We analyse the scaling of the two different U_F implementations by expressing the two circuits using only the *CNOT* and U_3 gates and counting the total number of gates. As evident from the scaling plots for the forward shift unitary U_F and the entire reflection unitary U_R in Figure 5, the ripple-carry incrementer leads to significantly lower gate counts compared to the multi-controlled *CNOT* incrementer. In passing, we note that other implementations of the forward shift unitary are possible, for instance, using the QFT-based Drapper adder [37, 38].

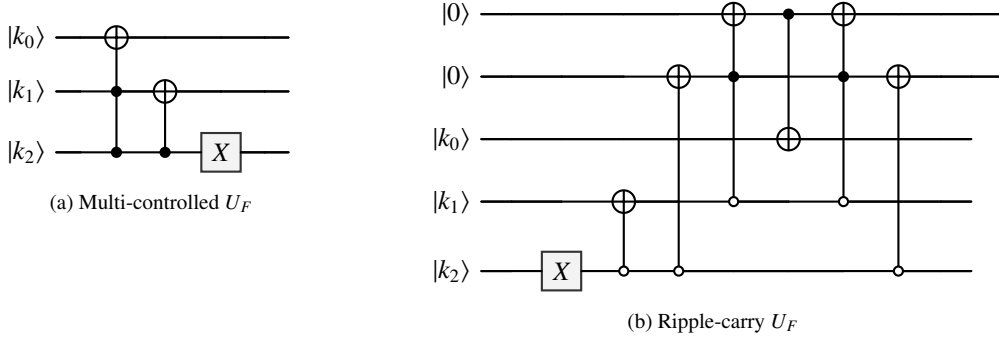


Figure 4: Two alternative quantum circuits for the forward shift unitary U_F , (a) using a cascade of controlled X gates and (b) based on ripple-carry addition. In (b), the top two qubits are ancilla qubits. Both circuits map an input vector $\sum_{k=0}^7 f_k |k\rangle$ to an output vector $\sum_{k=0}^7 |(k+1) \bmod 2^3\rangle$. The ancilla qubits have the value $|0\rangle$ before and after execution of the circuit.

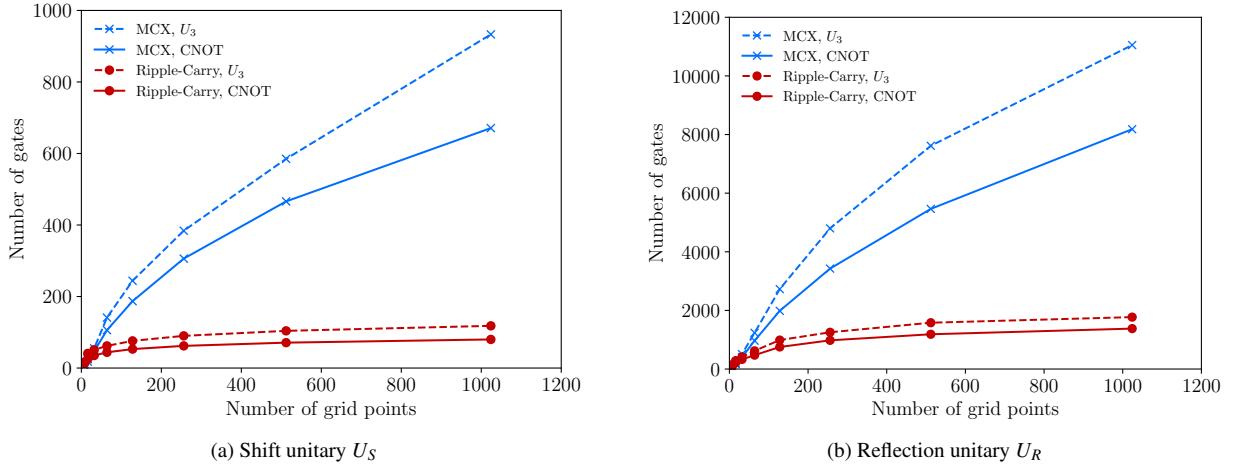


Figure 5: Total number of universal gates for (a) the shift unitary U_S and (b) the reflection unitary U_R . The label *MCX* refers to the implementation using multi-controlled X gates and the label *ripple-carry* to the implementation based on ripple-carry adder in Figures 4a and 4b, respectively.

3.3. Solution in the spectral domain

The solution of boundary value problems in the Fourier domain requires the encoding of the solution operator given by the diagonal matrix $\mathbf{D}^{-1}$. Focusing on the Poisson-Dirichlet problem, we consider the encoding of the function

$$d(k) = \begin{cases} \left(\frac{L}{\pi k}\right)^2 & 1 \leq k < N/2 \\ 0 & N/2 \leq k < N, \end{cases} \quad (45)$$

representing the diagonal entries of $\mathbf{D}^{-1}$. The function value $d(0)$ for $k = 0$ does not affect the solution and can be arbitrarily chosen. To facilitate the encoding of $d(k)$, we consider the function

$$\tilde{d}(k) = \arcsin\left(\frac{1}{k^2}\right). \quad (46)$$

The factor L^2/π^2 in (45) can be eliminated by rescaling the forcing term by the same amount so that $d(k)$ and $\tilde{d}(k)$ are the same up to the arcsin reparameterisation. The function $\tilde{d}(k)$ is evaluated on a quantum computer via the unitary

$$U_P(\tilde{d}(k)): |k\rangle|0\rangle \mapsto |k\rangle R_Y(2\tilde{d}(k))|0\rangle = \sqrt{1 - \frac{1}{k^4}} |k\rangle|0\rangle + \frac{1}{k^2} |k\rangle|1\rangle, \quad (47)$$

where $|k\rangle|0\rangle \equiv |k_0 \dots k_{n-2} k_{n-1}\rangle|0\rangle$. The rotation gate $R_Y(2\tilde{d}(k))$ is defined in terms of the matrix exponential of the Pauli Y gate and is given by

$$R_Y(2\tilde{d}(k)) = e^{-iY\tilde{d}(k)} = \begin{pmatrix} \cos \tilde{d}(k) & -\sin \tilde{d}(k) \\ \sin \tilde{d}(k) & \cos \tilde{d}(k) \end{pmatrix} = \begin{pmatrix} \sqrt{1 - \frac{1}{k^4}} & -\frac{1}{k^2} \\ \frac{1}{k^2} & \sqrt{1 - \frac{1}{k^4}} \end{pmatrix}. \quad (48)$$

Hence, the unitary $U_P(\tilde{d}(k)) \in \mathbb{R}^{2N \times 2N}$ can be identified as a 2D rotation in the plane spanned by the two vectors $|k\rangle|0\rangle$ and $|k\rangle|1\rangle$. The last qubit in (47), initially in state $|0\rangle$, is an ancilla and the amplitude $1/k^2$ of the component $|k\rangle|1\rangle$ is the sought function value $\tilde{d}(k)$. The amplitude of the component $|k\rangle|0\rangle$ is of no further interest. The ancilla qubit is required to ensure that the output vector is of unit length, despite the scaling applied to the basis vector $|k\rangle$ by $\tilde{d}(k)$. It is worth noting that instead of the arcsin parameterisation, an arccos parameterisation is equally possible.

Next, we approximate $\tilde{d}(k)$ by a polynomial

$$\tilde{d}(k) \approx \sum_{j=0}^p \alpha_j k^j, \quad (49)$$

where p is the chosen degree. The coefficients α_j are determined on a classical computer using standard polynomial approximation techniques. With a slight abuse of notation, we use the same symbol for $\tilde{d}(k)$ and its polynomial approximation in what follows. Moreover, we assume that the interval $1 \leq k \leq N/2$ is approximated with only one polynomial. In our computations, we also consider the decomposition of the interval into several sub-intervals; see [Appendix B](#) for the treatment of multiple intervals and multivariate polynomials. Expressing the argument k of the polynomial (49) as a binary and using the multinomial theorem yields

$$\begin{aligned} \tilde{d}(k) &= \sum_{j=0}^p \alpha_j (k_0 2^{n-1} + \dots + k_{n-2} 2^1 + k_{n-1} 2^0)^j = \sum_{j=0}^p \alpha_j \left(\sum_s k_s 2^{n-1-s} \right)^j \\ &= \sum_{j=0}^p \alpha_j \left(\sum_{r_0+r_1+\dots+r_{n-1}=j} \frac{j!}{r_0! r_1! \dots r_{n-1}!} \prod_{s=0}^{n-1} (k_s 2^{n-1-s})^{r_s} \right). \end{aligned} \quad (50)$$

Note that $k_s = (k_s)^{r_s}$ when $r_s \neq 0$ (since $k_s \in \{0, 1\}$). Hence, we can write

$$\tilde{d}(k) = \alpha_0 + \sum_{j=1}^p \alpha_j \left(\sum_{r_0+r_1+\dots+r_{n-1}=j} \frac{j!}{r_0! r_1! \dots r_{n-1}!} \prod_{s=0}^{n-1} k_s 2^{r_s(n-1-s)} \right). \quad (51)$$

Introducing this expression in (48) and (47), the summation reduces to a multiplication owing to the commutativity of 2D rotations. The rotation angles depend on k_s representing the state of the qubit s , hence leading to controlled rotations.

As an example, Figure 6 shows the circuit for evaluating the quadratic polynomial

$$\tilde{d}(k) = \alpha_0 + \alpha_1 k + \alpha_2 k^2 \quad (52)$$

for $k \in \{0, 1, 2, 3\}$, i.e., $k = k_0 2^1 + k_1 2^0$ with $k_0, k_1 \in \{0, 1\}$. This function expands to

$$\tilde{d}(k_0, k_1) = \alpha_0 + k_0(2\alpha_1 + 4\alpha_2) + k_1(\alpha_1 + \alpha_2) + k_0 k_1(4\alpha_2). \quad (53)$$

Consequently, the circuit can be implemented using the controlled rotations $R_Y(2\alpha_0)$, $R_Y(4\alpha_1 + 8\alpha_2)$, $R_Y(2\alpha_1 + 2\alpha_2)$ and $R_Y(8\alpha_2)$, or the multiplicative decomposition

$$U_P(\tilde{d}(k_0, k_1)) = U_{P_3}(8\alpha_2) U_{P_2}(2\alpha_1 + 2\alpha_2) U_{P_1}(4\alpha_1 + 8\alpha_2) U_{P_0}(2\alpha_0), \quad (54)$$

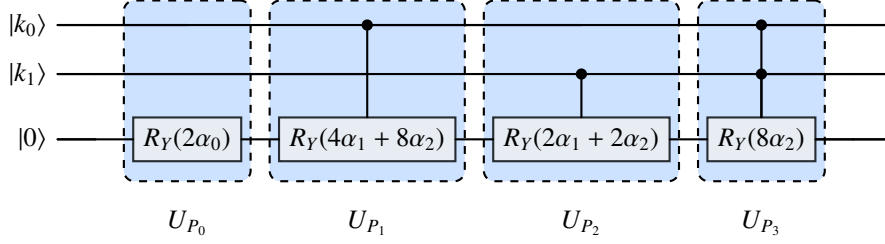


Figure 6: Quantum circuit for the unitary U_P evaluating the function $\tilde{d}(k) = \alpha_0 + \alpha_1 k + \alpha_2 k^2$ for $k \in \{0, 1, 2, 3\}$, where $k = k_0 2^1 + k_1 2^0$ and $k_0, k_1 \in \{0, 1\}$. The circuit is composed of four unitaries with each unitary implementing a rotation in a plane defined by the vectors $|k\rangle|0\rangle$ and $|k\rangle|1\rangle$, equivalently $|k_0 k_1\rangle|0\rangle$ and $|k_0 k_1\rangle|1\rangle$.

with each unitary implementing a 2D rotation in a specific plane given by the vectors $|k_0 k_1\rangle|0\rangle$ and $|k_0 k_1\rangle|1\rangle$. For instance, the first two unitaries are given by

$$U_{P_0}(2\alpha_0) = I \otimes I \otimes R_Y(2\alpha_0) = \begin{pmatrix} R_Y(2\alpha_0) & & & \\ & R_Y(2\alpha_0) & & \\ & & R_Y(2\alpha_0) & \\ & & & R_Y(2\alpha_0) \end{pmatrix}, \quad (55)$$

$$U_{P_1}(4\alpha_1 + 8\alpha_2) = |0\rangle\langle 0| \otimes I \otimes I + |1\rangle\langle 1| \otimes I \otimes R_Y(4\alpha_1 + 8\alpha_2) = \begin{pmatrix} I & & & \\ & I & & \\ & & R_Y(4\alpha_1 + 8\alpha_2) & \\ & & & R_Y(4\alpha_1 + 8\alpha_2) \end{pmatrix}. \quad (56)$$

3.4. Quantum spectral solver

We are now ready to introduce the quantum circuit for solving Dirichlet boundary value problems by combining state preparation, QFT, reflection, and polynomial encoding. For state preparation, any of the numerous state preparation algorithms can be used. The other necessary algorithms have all been discussed in this section. We first introduce the circuit for one-dimensional problems and then briefly extend it to two-dimensional problems.

The one-dimensional circuit in Figure 7 uses $n = 3$ qubits to represent the field vectors on the extended domain and one ancilla qubit for polynomial encoding. The vectors corresponding to the extended domain have $N = 2^n$ components, and the ones corresponding to the problem domain have $N/2$ components and $n - 1 =$ qubits. The unitary $U_S = U_R F_N U_R^\dagger$ implements the quantum sine transform and is composed of the unitary matrices U_R implementing the antisymmetric reflection and F_N implementing the quantum Fourier transform for an N -dimensional vector. All the qubits in the circuit are initialised in state $|0\rangle$. Using the state preparation unitary U_I , the prescribed source vector is encoded into the amplitudes of the state vector

$$I^{\otimes 2} \otimes U_I: |0\rangle^{\otimes 2} |0\rangle^{\otimes 2} = \sum_{k=0}^{N/2-1} f_k |0\rangle^{\otimes 2} |k\rangle, \quad (57)$$

where $|k\rangle = |k_0 k_1\rangle = |k_0\rangle |k_1\rangle$, with $k \in \{0, 1, 2, 3\}$ and $k_0, k_1 \in \{0, 1\}$. The sequence of subsequent mappings is as follows

$$I \otimes U_S: \sum_{k=0}^{N/2-1} f_k |0\rangle^{\otimes 2} |k\rangle \mapsto \sum_{k=0}^{N/2-1} \hat{f}_k |0\rangle^{\otimes 2} |k\rangle, \quad (58a)$$

$$U_P: \sum_{k=0}^{N/2-1} \hat{f}_k |0\rangle^{\otimes 2} |k\rangle \mapsto \sum_{k=0}^{N/2-1} \hat{f}_k \cos \tilde{d}(k) |0\rangle |0\rangle |k\rangle + \sum_{k=0}^{N/2-1} \hat{f}_k \sin \tilde{d}(k) |1\rangle |0\rangle |k\rangle, \quad (58b)$$

$$I \otimes U_S^\dagger: \sum_{k=0}^{N/2-1} \hat{f}_k \cos \tilde{d}(k) |0\rangle |0\rangle |k\rangle + \sum_{k=0}^{N/2-1} \hat{f}_k \sin \tilde{d}(k) |1\rangle |0\rangle |k\rangle \mapsto \sum_{k=0}^{N/2-1} u_k |1\rangle |0\rangle |k\rangle + \text{Junk}. \quad (58c)$$

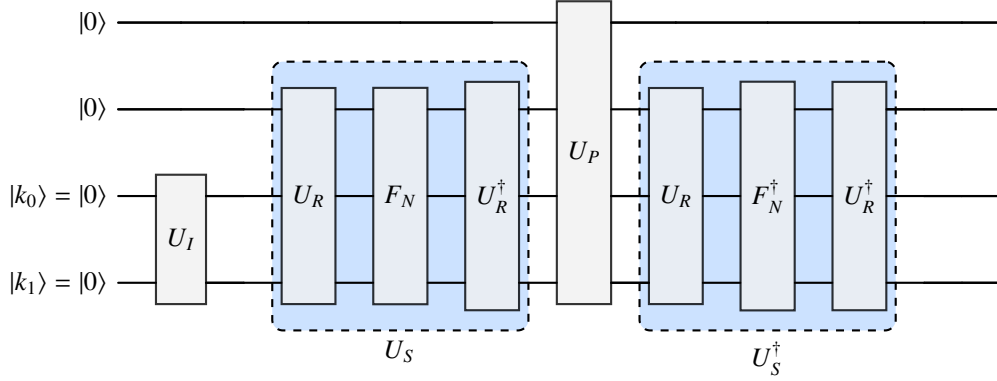


Figure 7: Quantum circuit for solving a one-dimensional Dirichlet boundary value problem. The extended domain Ω_E is discretised with $N = 2^3 = 8$ grid points so that the physical domain Ω is discretised with 4 grid points. The unitary U_I encodes the source $|f\rangle$ corresponding to the physical domain, and the output of the circuit is the solution $|u\rangle$ of the boundary value problem.

After the circuit is executed, the solution of the boundary value problem corresponds to the amplitudes of the states with the first qubit in state $|1\rangle$. The irrelevant terms with the first qubit in state $|0\rangle$ are denoted as “Junk”. The second qubit is only used within the implementation of U_S and will be in state $|0\rangle$ outside of U_S (on a noiseless quantum computer). Although we assumed in this section that the solution in the Fourier space is encoded using a single polynomial, in practice, we use a piecewise polynomial approximant. This affects only the design of the unitary U_P and requires ancilla qubits beyond the ones present in Figure 7.

The circuit diagram for two-dimensional Dirichlet boundary value problems is shown in Figure 8. The extended domain is discretised with $N \times N = 8 \times 8$ grid points. That is, the field vectors of the extended domain are encoded as the state vector of $n \times n = 3 \times 3$ qubits and the ones of the problem domain as the state vector of $n - 1 \times n - 1$ qubits. The grid points are labelled with the multi-index

$$|\mathbf{k}\rangle = |k^0\rangle |k^1\rangle = |k_0^0\rangle |k_1^0\rangle |k_0^1\rangle |k_1^1\rangle, \quad k^0, k^1 \in \{0, 1, 2, 3\}, \quad k_0^0, k_1^0, k_0^1, k_1^1 \in \{0, 1\}. \quad (59)$$

The two-dimensional solver is constructed mostly as the tensor product of two one-dimensional solvers. In Figure 8, the unitary U_S is exactly the same as for one-dimensional problems; it is applied simultaneously in the coordinate directions k^0 and k^1 . Although not evident from the circuit diagram, the qubits for the two coordinate directions k^0 and k^1 are independent from each other so that U_S is applied simultaneously. The primary difference between the one- and two-dimensional problems concerns the solution of the boundary value problem in the spectral domain. Hence, the unitary U_P is different from the one-dimensional case and depends on the specific boundary value problem considered.

4. Examples

We present a series of numerical experiments to assess the convergence and computational complexity of the proposed quantum approach. All circuits are implemented in Qiskit and executed on a noiseless simulator [39]. Convergence is studied by comparing the L_2 -norm error of the quantum solution with that of the classical spectral solution. We assess computational complexity by expressing the circuits in terms of only two-qubit $CNOT$ gates and single-qubit rotation gates U_3 . The gate set $\{CNOT, U_3\}$ is universal, meaning that any quantum circuit can be expressed using these two gates. We express the circuits with the $\{CNOT, U_3\}$ gate set by compiling them either in Qiskit or with TKET [39, 40], see Appendix A for details. TKET usually yields significantly lower gate counts by applying different algorithms for circuit optimisation. We first study the quantum solution of one-dimensional boundary value problems, specifically the Poisson-Dirichlet problem with both homogeneous and inhomogeneous boundary conditions, as well as a fractional differential equation. Subsequently, we study the quantum solution of these problems on two-dimensional domains.

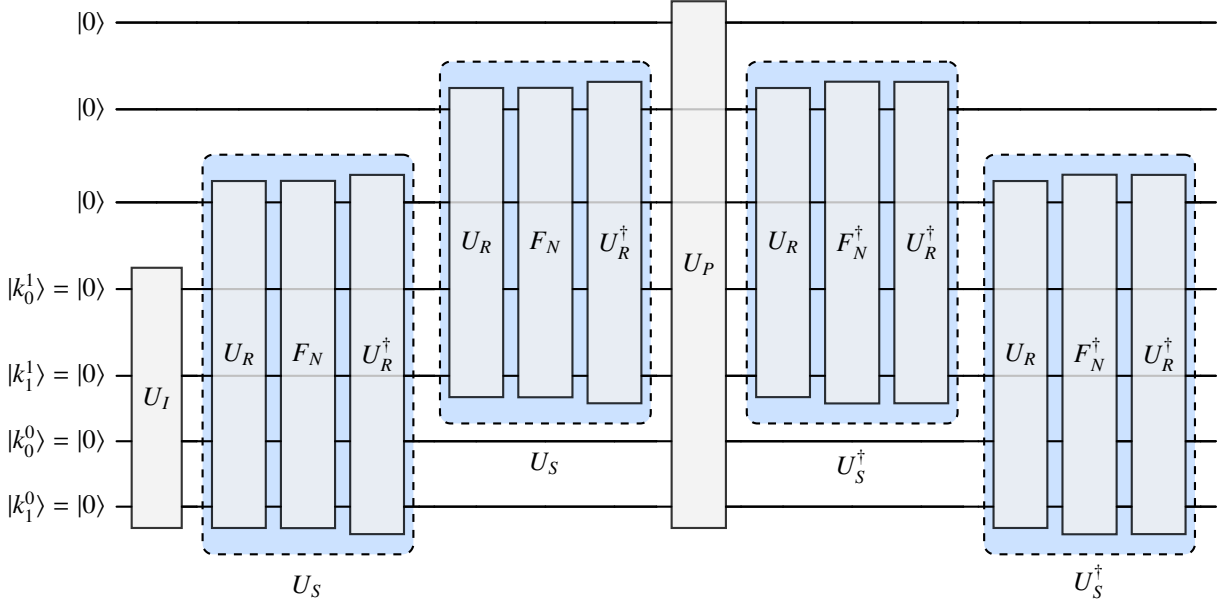


Figure 8: Quantum circuit for solving a two-dimensional Dirichlet boundary value problem. The extended domain Ω_E is discretised with $N \times N = 2^3 \times 2^3 = 8 \times 8$ grid points so that the physical domain Ω is discretised with 4×4 grid points. The unitary U_I encodes the source $|f\rangle$ corresponding to the physical domain, and the output of the circuit is the solution $|u\rangle$ of the boundary value problem. The qubits on which a gate does not act are rendered transparent, e.g., the first U_S does not act on the qubits $|k_0^1\rangle$ and $|k_1^1\rangle$.

4.1. One-dimensional boundary value problems

4.1.1. Homogeneous Poisson-Dirichlet problem

We seek in the domain $\Omega = (0, 1)$ the solution of the Poisson-Dirichlet boundary value problem with the boundary conditions $u(0) = u(1) = 0$ and the source field

$$f(x) = 100 \cos(2\pi x) \cos(5\pi x). \quad (60)$$

Its analytical solution is given by

$$u(x) = \frac{100(-58 + 116x + 49 \cos(3\pi x) + 9 \cos(7\pi x))}{882\pi^2}; \quad (61)$$

see Figure 9. The domain Ω is discretised with N uniformly distributed grid points with a spacing of $h = 1/N$. The problem is then solved with the quantum circuit introduced in Section 3.4 by applying a sequence of unitary transformations. The input quantum state vector $|f\rangle$ is obtained by evaluating $f(x)$ at the grid points and encoding it using a state preparation algorithm. The state vector $|f\rangle$ must have a unit length, which is achieved by scaling $f(x)$. The computed solution is rescaled to account for the initial scaling of $f(x)$. As discussed, the solution of the Poisson-Dirichlet problem reduces to algebraic equations in the Fourier space. We encode the solution operator $\mathbf{D}^{-1}$ introduced in (20) using piecewise polynomials of degrees $p = 3$ and $p = 4$ as described in Section 3.3. To improve the approximation of $\mathbf{D}^{-1}$, we choose smaller segment sizes closer to the origin. The segment size distribution is optimised by trial and error and has not been further formalised in this work.

The convergence of the classical and quantum spectral approximation is shown in Figure 10a. The classical approximation exhibits uniform convergence, whereas the convergence of the quantum approximation exhibits a flattening when refining the grid and depends on the degree p in the polynomial approximation of the solution operator $\mathbf{D}^{-1}$. However, the accuracy of the quantum solution for finer meshes can be systematically increased by choosing either a higher p or smaller segment sizes in piecewise polynomial approximation of $\mathbf{D}^{-1}$. As visible in Figure 10b, p has an effect on the number of $CNOT$ and U_3 gates in the quantum circuit. Higher polynomial degrees and smaller segment sizes lead to an increase in gate counts. Irrespective of their total number, the number of gates always depends polylogarithmically on the number of grid points.

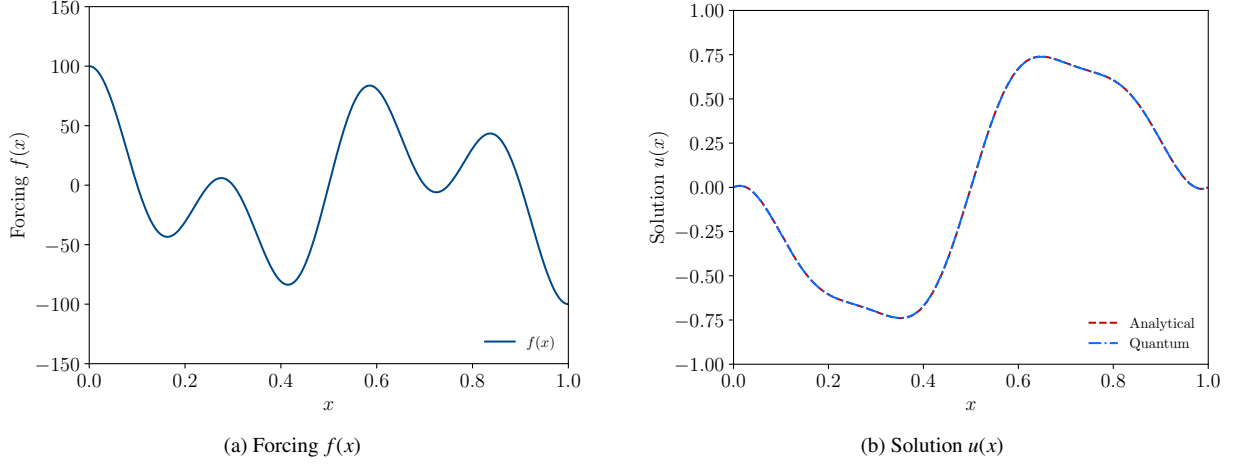


Figure 9: One-dimensional homogenous Poisson-Dirichlet problem.

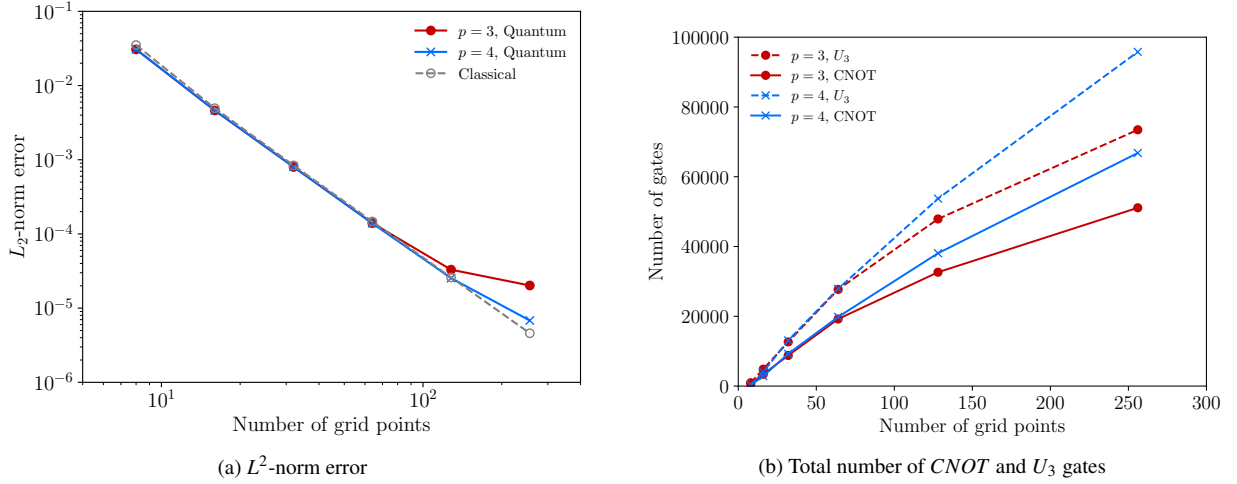


Figure 10: One-dimensional homogenous Poisson-Dirichlet problem.

4.1.2. Inhomogeneous Poisson-Dirichlet problem

Building on the previous example, we consider the inhomogeneous Poisson-Dirichlet problem on the domain $\Omega = (0, 1)$. The boundary conditions are now $u(x = 0) = 0.5$ and $u(x = 1) = 1$, and the source field $f(x)$ is the same as in (60). The analytical solution becomes

$$u(x) = 0.5(x + 1) + \frac{100(-58 + 116x + 49 \cos(3\pi x) + 9 \cos(7\pi x))}{882\pi^2}. \quad (62)$$

To solve this problem, we consider the homogeneous Poisson-Dirichlet problem

$$-\frac{d^2 v(x)}{dx^2} = f(x) + \frac{d^2 g(x)}{dx^2}, \quad \forall x \in \Omega, \quad (63)$$

$$v(0) = 0, \quad v(1) = 0.$$

The auxiliary function is chosen as

$$g(x) = -0.5(x - 1)^3 + x^3, \quad (64)$$

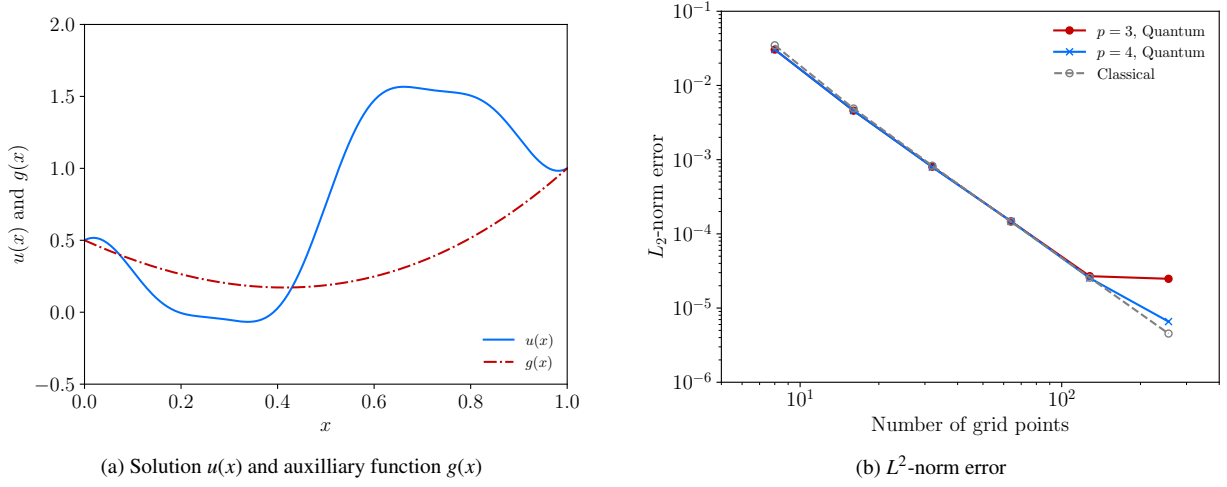


Figure 11: One-dimensional inhomogeneous Poisson-Dirichlet problem.

and satisfies the Dirichlet boundary conditions and is sufficiently smooth. After computing $v(x)$ the solution of the original problem is recovered using the relationship $u(x) = v(x) + g(x)$, see Figure 11a.

The homogenous Poisson-Dirichlet problem for $v(x)$ is solved using the classical or quantum spectral methods following the procedure in the previous example. In the quantum approach, we first compute classically the modified source term and then classically recover the solution from the quantum solution of the homogeneous problem. The convergence of the classical and quantum spectral approximations is shown in Figure 11b.

4.1.3. Fractional stochastic differential equation

We next consider a problem motivated by the representation of random fields as the solution of fractional stochastic differential equations. For convenience, we restate the fractional differential equation (21) as

$$\left(\kappa^2 - \frac{d^2}{dx^2}\right)^\beta u(x) = \frac{1}{\tau} f(x), \quad x \in (0, 1). \quad (65)$$

Here, the source $f(x)$ is a Gaussian white-noise field

$$f(x) \sim \mathcal{GP}(0, \delta(x - x')), \quad (66)$$

with the Dirac delta covariance function $\delta(x - x')$. As is well known, see e.g. [25, 27], on an unbounded domain the solution $u(x)$ is a Gaussian process

$$u(x) \sim \mathcal{GP}(0, c_u(x, x')), \quad (67)$$

where $c_u(x, x')$ is the Matérn covariance function which has a closed-form; see Appendix C. The length-scale, smoothness and variance of the random solution $u(x)$ are governed by the parameters κ, β and τ , respectively.

The fractional stochastic differential equation (65) provides a means of sampling from the Gaussian process $u(x)$. Specifically, a sample of $u(x)$ is obtained by first drawing a sample from the Gaussian white-noise field $f(x)$ and then solving (65). The discretisation of the Gaussian white-noise field yields the random vector

$$\mathbf{f} = \frac{1}{N} \begin{pmatrix} 0 & f(x_1) & \dots & f(x_{N-1}) \end{pmatrix}^\top, \quad (68)$$

where each component $f_k \sim \mathcal{N}(0, 1/\tau^2)$ is a univariate Gaussian random variable with variance $1/\tau^2$, and the entire vector is scaled by dividing by the number of grid points N . The forcing vector $\mathbf{f}$ is sampled by drawing independent

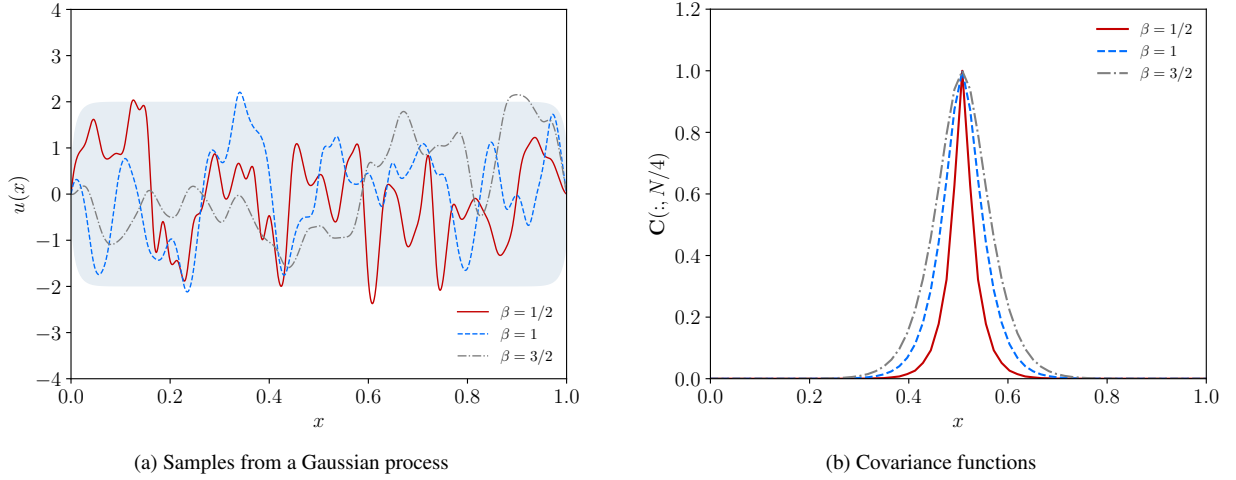


Figure 12: One-dimensional fractional stochastic differential equation.

samples from a normal distribution. The respective solution vector $\mathbf{u}$ is obtained from the quantum equivalent of (25), restated for convenience,

$$\mathbf{u} = \mathbf{R}^\top \mathbf{F}^\dagger \mathbf{R} \mathbf{D}^{-1} \mathbf{R}^\top \mathbf{F} \mathbf{R} \mathbf{f}.$$

In Figure 12a, three sampled functions for $\kappa = 40$, $\tau = 4.279 \cdot 10^{-5}$, and three different $\beta \in \{1/2, 1, 3/2\}$ are depicted. As expected, the sample becomes progressively smoother with increasing β .

It can be shown that the discrete solution vector $\mathbf{u}$ has the multivariate Gaussian probability density

$$\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}) = \mathcal{N}\left(\mathbf{0}, \frac{1}{N} \mathbf{R}^\top \mathbf{F}^\dagger \mathbf{R} \mathbf{D}^{-2} \mathbf{R}^\top \mathbf{F} \mathbf{R}\right), \quad (69)$$

where $\mathbf{C}$ is the covariance matrix. The entries of $\mathbf{C}$ express the correlation between the components of the solution vector and correspond to the Matérn covariance function $c_u(x, x')$ of the continuous problem. In Figure 12b, the central row of the covariance matrix $\mathbf{C}(:, N/4)$, corresponding to the function $c_u(x, x' = 0)$, for the three different considered smoothness parameters is plotted. The L_2 norm of the difference between $c_u(x, x' = 0)$ and its approximation by $\mathbf{C}$ for different numbers of grid points is shown in Figure 13. The convergence depends on the smoothness parameter β , with higher convergence rates for larger β . In the quantum approximation of the operator $\mathbf{D}^{-2}$, we consider piecewise polynomials of degrees $p = 3$ and $p = 4$. The flattening in the convergence curves can again be attributed to the polynomial approximation error. The total number of the $CNOT$ and U_3 gates in dependence of the total number of grid points N is plotted in Figure 14, confirming again their polylogarithmic scaling.

4.2. Two-dimensional boundary value problems

4.2.1. Inhomogeneous Poisson-Dirichlet problem

We proceed now to study the quantum spectral solution of two-dimensional boundary value problems. To begin with, we seek on the domain $\Omega = (0, L)^2 \subset \mathbb{R}^2$ the solution of the Poisson-Dirichlet problem with the source field

$$f(\mathbf{x}) = 13\pi^2 \sin(2\pi x_0) \sin(3\pi x_1) + 17\pi^2 \sin(\pi x_0) \sin(4\pi x_1) - 9x_0 - 15x_1, \quad (70)$$

and the Dirichlet boundary conditions

$$\begin{aligned} u(x_0 = 0, x_1) &= \frac{1}{2} + \frac{5}{2}x_1^2, & u(x_0 = 1, x_1) &= 2 + \frac{5}{2}x_1^2, \\ u(x_0, x_1 = 0) &= \frac{1}{2} + \frac{3}{2}x_0^3, & u(x_0, x_1 = 1) &= 3 + \frac{3}{2}x_0^2. \end{aligned} \quad (71)$$

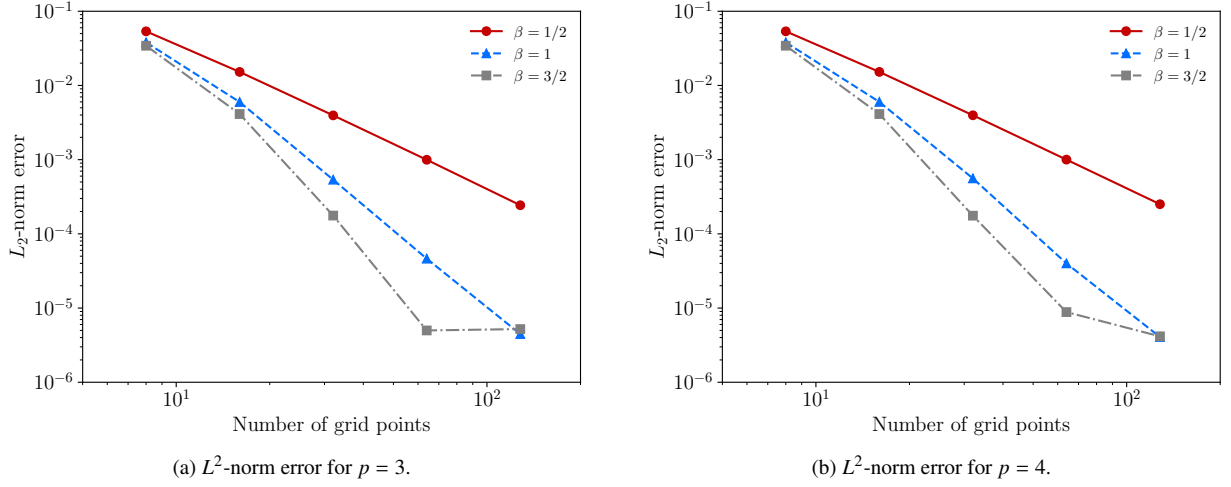


Figure 13: One-dimensional fractional stochastic differential equation. Convergence of the L^2 -norm error in approximating $c(x, x' = 0)$.

We consider again the decomposition $u(\mathbf{x}) = v(\mathbf{x}) + g(\mathbf{x})$, where $v(\mathbf{x})$ is the solution of the homogeneous Poisson-Dirichlet problem and $g(\mathbf{x})$ is a sufficiently smooth function which satisfies the boundary conditions. We choose

$$g(\mathbf{x}) = \frac{1}{2} (3x_0^2 + 5x_1^2 + 1). \quad (72)$$

In general, this auxiliary function can be, for instance, constructed by approximating the prescribed boundary data using polynomials. The homogeneous Dirichlet-Poisson problem to solve reads

$$\begin{aligned} -\nabla^2 v(\mathbf{x}) &= f(\mathbf{x}) + \nabla^2 g(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega, \\ v(\mathbf{x}) &= 0, \quad \forall \mathbf{x} \in \partial\Omega. \end{aligned} \quad (73)$$

The bivariate function for computing the Fourier space solution of this problem is given by

$$d(k^0, k^1) = \begin{cases} \left(\frac{L}{\pi}\right)^2 \frac{1}{(k^0)^2 + (k^1)^2} & (k^0, k^1) \in (0, N/2) \times (0, N/2) \\ 0 & \text{otherwise.} \end{cases} \quad (74)$$

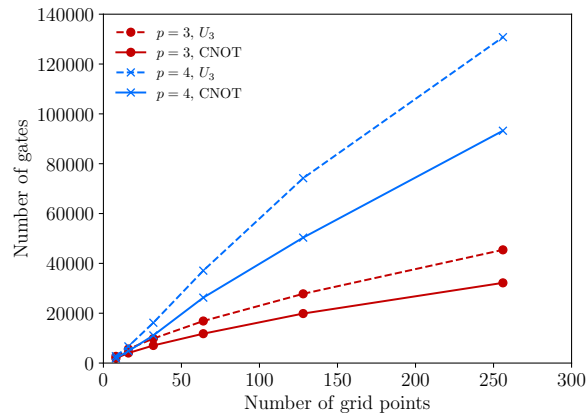


Figure 14: One-dimensional fractional stochastic differential equation. The total number of *CNOT* and U_3 gates.

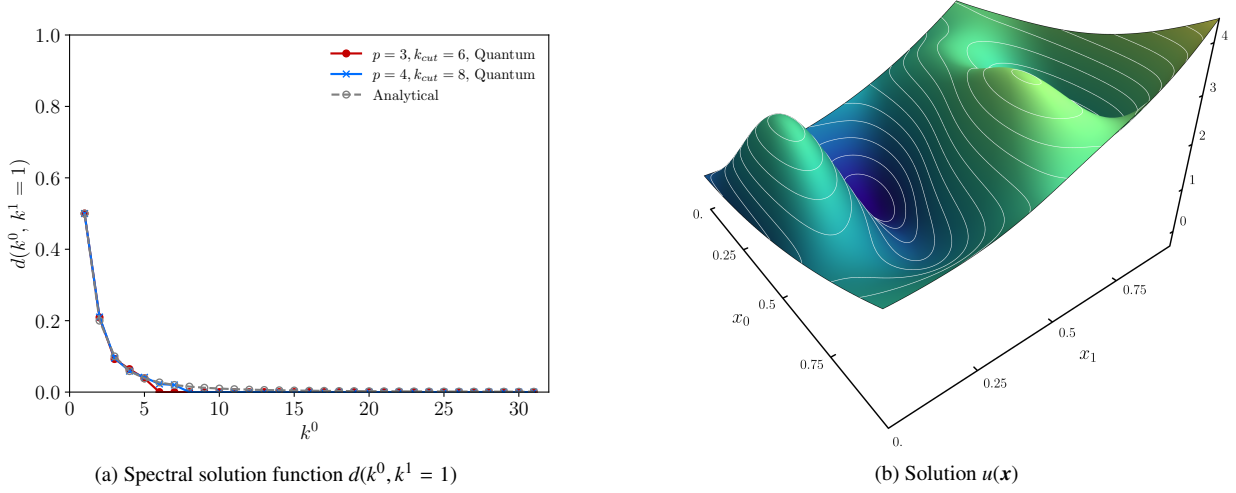


Figure 15: Two-dimensional inhomogeneous Poisson-Dirichlet problem. In (a), the spectral solution function and its approximation with polynomials of degree $p = 3$ and $p = 4$ are shown.

As in one-dimensional problems, $d(k^0, k^1)$ is first approximated with a bivariate piecewise polynomial and then encoded as a unitary U_P , see [Appendix B](#). In two-dimensional problems, the number of polynomials must be restricted to limit the number of gates so that the quantum circuits can still be simulated in Qiskit in a reasonable time. This limitation is relevant to the simulation of the circuit using a classical computer and will not be a problem for future quantum computers. To limit the number of gates, we decompose the Fourier domain $(0, N/2) \times (0, N/2)$ into four subdomains and approximate $d(k^0, k^1)$ only on the subdomain $(0, k_{\text{split}}) \times (0, k_{\text{split}})$. In the remaining three subdomains, $d(k^0, k^1)$ is set to zero. We choose multivariate polynomials of degree $p = 3$ and $p = 4$ as approximants and choose $k_{\text{split}} = 6$ for $p = 3$ and $k_{\text{split}} = 8$ for $p = 4$. In Figure 15a, the exact function $d(k^0, k^1)$ and its two different approximations for $k^1 = 1$ are plotted. Notice the very fast decay of the solution function $d(k^0, k^1 = 1)$ with increasing k^0 , which justifies the neglecting of the higher modes.

Figure 15b shows the contour plot of the quantum solution using a discretisation with 64×64 grid points and a quartic approximation of the solution function $d(k^0, k^1)$. The convergence of L^2 -norm error is depicted in Figure 16a, confirming the uniform convergence of the proposed quantum approach. It is worth noting that the chosen cut-off value k_{split} for the solution function is kept for all discretisations the same. The associated scaling of the total number of $CNOT$ and U_3 gates is plotted in Figure 16b. Notice the significant increase in the gate counts in comparison to one-dimensional problems. Irrespective of this, the polylogarithmic scaling of the total number of gates is evident.

4.2.2. Fractional stochastic partial differential equation

We now apply the proposed quantum approach to the fractional stochastic partial differential representation of two-dimensional random fields. The two-dimensional generalisation of the stochastic ordinary differential equation (21) takes the form

$$(\kappa^2 - \nabla^2)^\beta u(\mathbf{x}) = \frac{1}{\tau} f(\mathbf{x}), \quad \mathbf{x} \in (0, 1) \times (0, 1), \quad (75)$$

where $f(\mathbf{x})$ denotes white noise and κ, β and τ are prescribed parameters. The corresponding Fourier-space solution is given by the bivariate function

$$d(k^0, k^1) = \begin{cases} \frac{1}{\tau \left(\kappa^2 + \frac{\pi^2}{L^2} ((k^0)^2 + (k^1)^2) \right)^\beta} & (k^0, k^1) \in (0, N/2) \times (0, N/2) \\ 0 & \text{otherwise.} \end{cases} \quad (76)$$

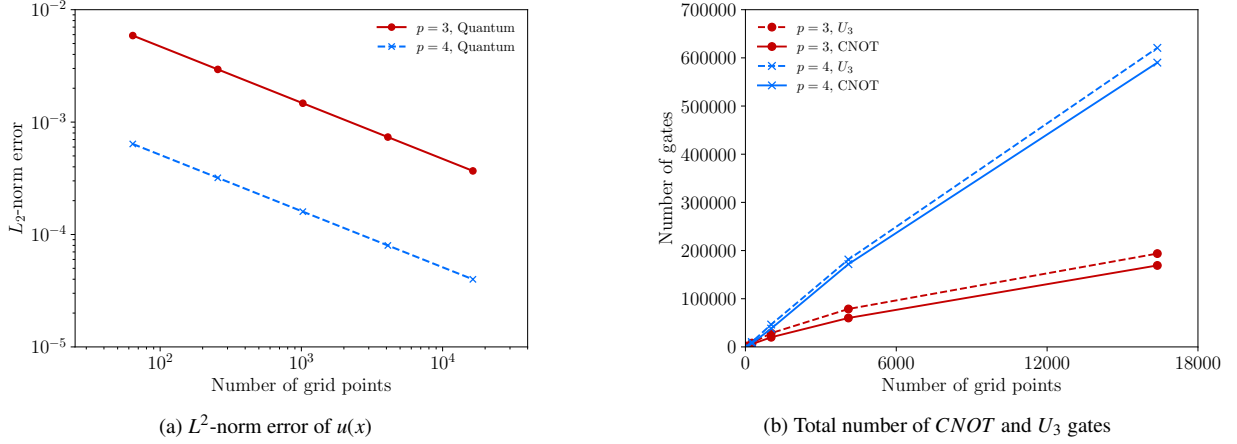


Figure 16: Two-dimensional inhomogeneous Poisson-Dirichlet problem.

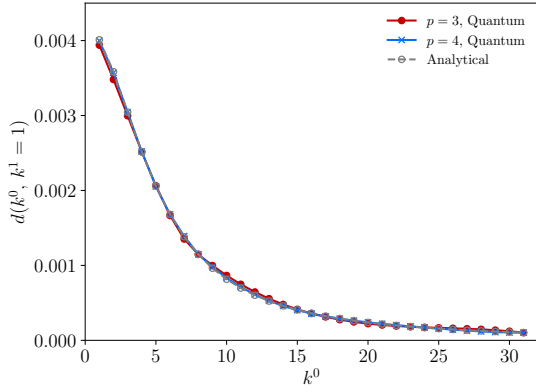
The three parameters are chosen as $\kappa = 10/\sqrt{2}$, $\tau = 0.01995$, and $\beta = 1$. As shown in Figure 17a, the solution function $d(k^0, k^1)$ is relatively large throughout the Fourier domain. This contrasts with the Fourier space solution function (74) of the Poisson-Dirichlet problem, which decays rapidly at large wavenumbers. We decompose the Fourier domain $(0, N/2) \times (0, N/2)$ into four subdomains by introducing the wavenumber k_{split} along each axis as discussed in the previous section. Different from the Poisson-Dirichlet problem, we approximate $d(k^0, k^1)$ in each of the four subdomains with a polynomial. We choose $k_{\text{split}} = 8$ and bivariate polynomials of degree $p = 3$ and $p = 4$ as approximants. The resulting approximations for $k^1 = 1$ are compared with the exact $d(k^0, 1)$ in Figure 17a.

As shown in Figure 17b the two different approximations for $d(k^0, k^1 = 1)$ yield essentially the same approximation to the covariance function $c_u(x, x' = 0)$. The approximate covariance is given by the entries of the covariance matrix. The good agreement between the quantum approximations and the exact function is evident. Sample realisations of the random field are generated by sampling a white noise forcing vector and solving the partial differential equation. In Figure 18, two different realisations of the random field are depicted.

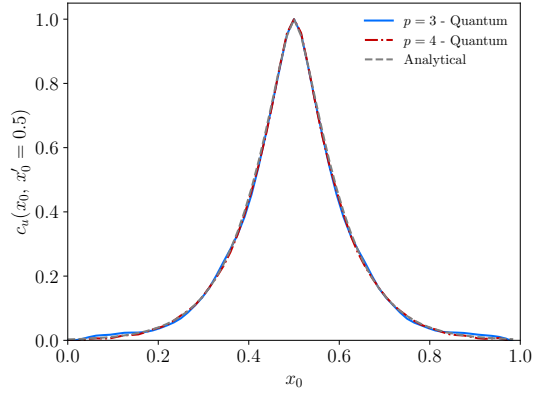
5. Conclusions

We have introduced a quantum spectral approach to solve non-periodic Dirichlet boundary value problems with polylogarithmic complexity. This represents an exponential speedup relative to the at-best-linear, often worse, complexity achievable with classical approaches. The quantum speedup is obtained by representing the N -dimensional forcing and solution vectors using $n = \log_2 N$ qubits, owing to quantum entanglement, and by designing a unitary solution operator with a polynomial number of one and two-qubit gates. It appears unlikely that most classical discretisation approaches can be efficiently ported to quantum computing platforms due to the inherent difficulty of converting classical vectors and matrices into the exponentially compressed quantum representation. Therefore, amongst the classical methods, well-structured and straightforward discretisation methods, such as the considered Fourier-based spectral approach, are most suited for quantum computing. In the spectral domain, the solution operator simplifies to a diagonal matrix, due to the orthogonality property of the Fourier basis functions, and is relatively easy to invert. Vectors and matrices are converted between the real and spectral domains using the QFT algorithm, which is perhaps the most widely used algorithm throughout quantum computing. As demonstrated, homogeneous Dirichlet boundary conditions can be enforced by adding d additional qubits to the discretisation, where d is the spatial dimension, thereby doubling the domain size and the number of discretisation points. Non-homogeneous Dirichlet boundary conditions are enforced by making use of the superposition principle.

We close by noting a few possible extensions and algorithmic improvements of the proposed approach. Although we considered only Dirichlet-Dirichlet boundary conditions, other boundary conditions can be enforced by choosing other extensions [31]. Specifically, Neumann-Dirichlet boundary conditions can be enforced by adding $2d$ qubits,

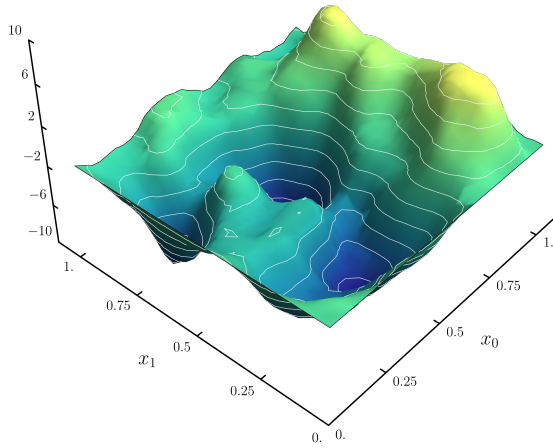


(a) Spectral solution function $d(k^0, k^1 = 1)$

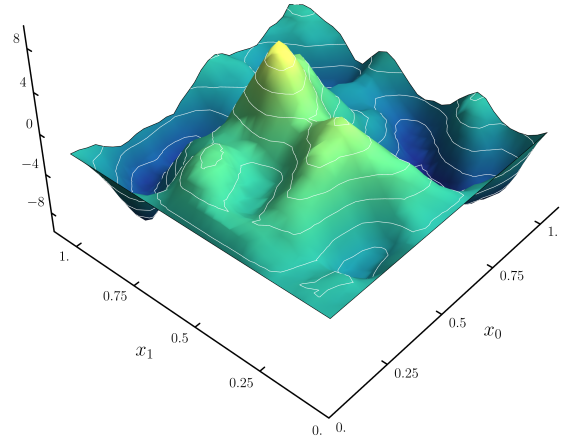


(b) Covariance function

Figure 17: Two-dimensional fractional stochastic partial differential equation. Analytic and approximate spectral solution and Matérn functions for polynomial approximants of degree $p = 3$ and $p = 4$.



(a) $u(x_0, x_1)$



(b) $u(x_0, x_1)$

Figure 18: Two-dimensional fractional stochastic partial differential equation. Sample solution fields from the Gaussian random field with Matérn covariance.

where d is the spatial dimension, and quadrupling the domain size. In the present work, our main aim was to demonstrate polylogarithmic complexity irrespective of the total number of gates used. A reduction in gate counts is relevant for near-term quantum computing due to the limited number of qubits and the impossibility of error correction in current and upcoming quantum computers. The gate counts can be reduced by choosing alternative algorithms for each of the operations in the proposed quantum spectral approach. For instance, it may be possible to replace the used QFT algorithm with an approximate QFT algorithm that omits gates with a very small rotation [41]. Similarly, it is possible to compute or approximate the inverse of the diagonal solution matrix with alternative techniques, such as QSVT [7] or recent Fourier- or Chebyshev-based polynomial approximation techniques [6], which may or may not yield a reduction in gate counts. We did not consider in this paper the extraction of classical data of engineering interest from the obtained quantum states. To this end, recent advances in shadow tomography [42] and quantum

machine learning [43] may provide practical solutions and warrant further investigation. In terms of future applications, solving non-periodic boundary value problems on d -dimensional hyperrectangles is, among others, relevant in homogenisation [44–46]. Consequently, our quantum solver can be used as a component in the quantum homogenisation approach introduced earlier by Liu et al. [1]. Finally, quantum approaches offer distinct advantages when applied to stochastic problems that require the repeated solution of the same problem with varying inputs [47]. The inputs are samples from a random field, which, as demonstrated in this paper, can be obtained by solving a stochastic differential equation.

Acknowledgments

EF is supported through Reward for Excellence Award of the Early Career Development Program at the College of Science and Engineering, University of Glasgow.

Appendix A. Algorithmic complexity

We assess the complexity of quantum algorithms by expressing circuits using the gate set $\{CNOT, U_3\}$ with the two-qubit gate

$$CNOT = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad (\text{A.1})$$

and the single qubit gate

$$U_3(\theta, \phi, \lambda) = \begin{pmatrix} \cos \frac{\theta}{2} & -e^{i\lambda} \sin \frac{\theta}{2} \\ e^{i\phi} \sin \frac{\theta}{2} & e^{i(\phi+\lambda)} \cos \frac{\theta}{2} \end{pmatrix}, \quad (\text{A.2})$$

where $\theta, \phi, \lambda \in \mathbb{R}$ are three angles. The gate set $\{CNOT, U_3\}$ is universal in the sense that any quantum circuit can be expressed using its two gates.

All of our circuits are implemented in Qiskit [39]. Subsequently, they are expressed using only $\{CNOT, U_3\}$ gates by compiling them either in Qiskit or TKET [40]. In Qiskit, we use the `transpile` command, and in TKET, the `qiskit_to_tk` and `get_compiled_circuit` commands, with optimisation level 3. Usually, the number of gates in TKET is significantly lower than in Qiskit.

Appendix B. Encoding of bivariate piecewise polynomials

Consider the two-dimensional domain $\Omega \subset \mathbb{Z}^2$ and its partitioning into $M \times M$ non-overlapping subdomains ω_c such that

$$\Omega = \bigcup_c \omega_c, \quad \mathbf{c} = (c^0, c^1), \quad c^0, c^1 = 0, \dots, M-1. \quad (\text{B.1})$$

Each subdomain ω_c is defined in terms of its bounding box

$$\omega_c = [l_c^0, u_c^0] \times [l_c^1, u_c^1], \quad (\text{B.2})$$

where l_c^0 and u_c^0 are the lower and upper grid points in the coordinate direction k^0 , and l_c^1 and u_c^1 are the respective grid points in the coordinate direction k^1 . Each subdomain ω_c is associated with a bivariate polynomial

$$\tilde{d}_c(k^0, k^1) = \begin{pmatrix} 1 \\ k^0 \\ (k^0)^2 \\ \vdots \\ (k^0)^p \end{pmatrix}^\top \begin{pmatrix} \alpha_{c,00} & \alpha_{c,01} & \alpha_{c,02} & \cdots & \alpha_{c,0p} \\ \alpha_{c,10} & \alpha_{c,11} & \alpha_{c,12} & \cdots & \alpha_{c,1p} \\ \alpha_{c,20} & \alpha_{c,21} & \alpha_{c,22} & \cdots & \alpha_{c,2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_{c,p0} & \alpha_{c,p1} & \alpha_{c,p2} & \cdots & \alpha_{c,pp} \end{pmatrix} \begin{pmatrix} 1 \\ k^1 \\ (k^1)^2 \\ \vdots \\ (k^1)^p \end{pmatrix} = \begin{pmatrix} 1 \\ k^0 \\ (k^0)^2 \\ \vdots \\ (k^0)^p \end{pmatrix}^\top \alpha_c \begin{pmatrix} 1 \\ k^1 \\ (k^1)^2 \\ \vdots \\ (k^1)^p \end{pmatrix}, \quad (\text{B.3})$$

where p is the polynomial degree and $\alpha_c \in \mathbb{R}^{(p+1) \times (p+1)}$ is the coefficient matrix.

The support of $\tilde{d}_c(k^0, k^1)$ is restricted to the subdomain ω_c with the same index $\mathbf{c}$. Hence, to evaluate it at a given grid point $\mathbf{k} = (k^0, k^1)$ first the subdomain index $\mathbf{c}$ must be found, such that $l_c^0 \leq k^0 < u_c^0$ and $l_c^1 \leq k^1 < u_c^1$. Because cell boundaries are axis-aligned, the subdomain index $\mathbf{c}$ can be found using $4M$ integer comparisons. The number of integer comparisons can be reduced by a factor of two by only identifying the set of all subdomain indices $\{\mathbf{c}\}$ such that $k^0 < u_c^0$ and $k^1 < u_c^1$ and evaluating all the associated polynomials $\{d_c(\mathbf{k})\}$. To facilitate this, the polynomials are

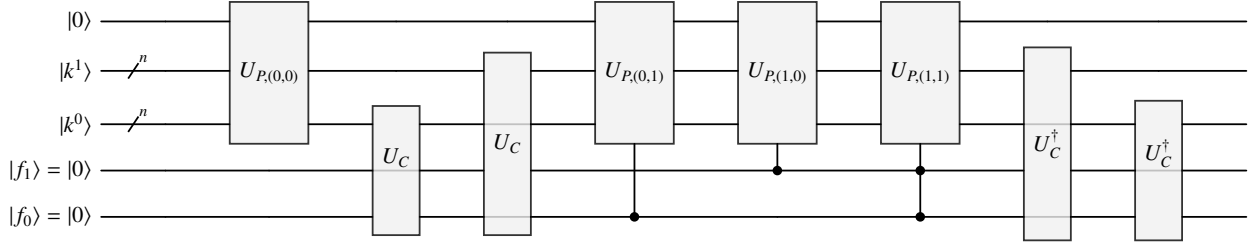


Figure B.19: Quantum circuit for encoding a piecewise bivariate polynomial. The domain $[0, 2^n - 1]^2$ is partitioned into 2×2 subdomains of equal size. The values of the two flag qubits, $|f_0\rangle$ and $|f_1\rangle$, are set by the integer comparator U_C and determine the relevant polynomials for a given point evaluation point $\mathbf{k} = (k^0, k^1)$. The first flag is in state $|f_0\rangle = |1\rangle$ when $k^0 \geq 2n - 1$ and the second flag is in state $|f_1\rangle = |1\rangle$ when $k^0 \geq 2n - 1$. After the circuit is executed the function value is the amplitude of the state $|1\rangle|k^1\rangle|k^0\rangle|0\rangle|0\rangle$.

modified as follows

$$\begin{aligned}
 \tilde{d}_{(0,0)}(\mathbf{k}) &\leftarrow \tilde{d}_{(0,0)}(\mathbf{k}) \\
 \tilde{d}_{(1,0)}(\mathbf{k}) &\leftarrow \tilde{d}_{(1,0)}(\mathbf{k}) - \tilde{d}_{(0,0)}(\mathbf{k}) \\
 \tilde{d}_{(2,0)}(\mathbf{k}) &\leftarrow \tilde{d}_{(2,0)}(\mathbf{k}) - \tilde{d}_{(1,0)}(\mathbf{k}) \\
 &\vdots \\
 \tilde{d}_{(0,1)}(\mathbf{k}) &\leftarrow \tilde{d}_{(0,1)}(\mathbf{k}) - \tilde{d}_{(0,0)}(\mathbf{k}) \\
 \tilde{d}_{(1,1)}(\mathbf{k}) &\leftarrow \tilde{d}_{(1,1)}(\mathbf{k}) - \tilde{d}_{(0,1)}(\mathbf{k}) - \tilde{d}_{(1,0)}(\mathbf{k}) + \tilde{d}_{(0,0)}(\mathbf{k}) \\
 \tilde{d}_{(2,1)}(\mathbf{k}) &\leftarrow \tilde{d}_{(2,1)}(\mathbf{k}) - \tilde{d}_{(1,1)}(\mathbf{k}) - \tilde{d}_{(2,0)}(\mathbf{k}) + \tilde{d}_{(1,0)}(\mathbf{k}) \\
 &\vdots
 \end{aligned} \tag{B.4}$$

As an example, the domain $\Omega = [0, 2^n - 1]^2$ consists of four subdomains ω_c so that $\mathbf{c} \in \{(0, 0), (0, 1), (1, 0), (1, 1)\}$. The four non-overlapping subdomains are of equal size so that, e.g., the subdomain $\omega_{(0,0)}$ has the domain $[0, 2^{n-1} - 1]^2$. The piecewise polynomial defined over Ω is evaluated with the circuit shown in Figure B.19. The circuit consists of four unitaries U_{P_e} for evaluating the four polynomials defined according to (B.4). The unitaries are conditioned on the two ancilla qubits f_0 and f_1 , which are set by the integer comparator $U_C(2^{n-1})$ as applied to the first and second coordinate direction k^0 and k^1 , i.e.,

$$U_C(2^{n-1}): |k^0\rangle|f_0=0\rangle \mapsto |k^0\rangle|k^0 \geq 2^{n-1}\rangle \quad U_C(2^{n-1}): |k^1\rangle|f_1=0\rangle \mapsto |k^1\rangle|k^1 \geq 2^{n-1}\rangle. \tag{B.5}$$

Focusing on the first comparator unitary, it has the output $|k^0\rangle|1\rangle$ for all grid points with $k^0 \geq 2^{n-1}$ and $|k^0\rangle|0\rangle$ otherwise. Most quantum computing libraries, including Qiskit, provide an implementation of U_C with polylogarithmic complexity. U_C can be, for instance, defined in terms of binary addition.

Appendix C. Stochastic differential equation representation of random fields

A Matérn-type random field $u(x)$ is the solution of the stochastic ordinary differential equation

$$\left(\kappa^2 - \frac{d^2}{dx^2}\right)^\beta u(x) = \frac{1}{\tau} f(x), \quad x \in (-\infty, +\infty), \tag{C.1}$$

where the Gaussian white-noise forcing $f(x)$ is given by

$$f(x) \sim \mathcal{GP}(0, \delta(x - x')) , \tag{C.2}$$

and has the covariance function

$$\text{cov}(f(x), f(x')) = c_f(x, x') = \mathbb{E}[f(x)f(x')] = \delta(x - x'). \tag{C.3}$$

The solution of (C.1) for a single realisation of $f(x)$ can be determined via the Greens function $g(x, x')$ as

$$u(x) = \int g(x, x') f(x') dx. \quad (\text{C.4})$$

By linearity of expectation, the random solution field $u(x)$ has the covariance

$$\text{cov}(u(x), u(x')) = \mathbb{E}[u(x)u(x')] = \int \int g(x, x'') \mathbb{E}[f(x'')f(x''')] g(x', x''') dx'' dx''' = \int g(x, x'') g(x', x'') dx''. \quad (\text{C.5})$$

This expression is equal to the classical Matérn covariance given by

$$\text{cov}(u(x), u(x')) = c_u(x, x') = \frac{1}{2^{\nu-1} \Gamma(\nu)} \left(\frac{\sqrt{2\nu}}{\ell} \|x - x'\| \right)^\nu K_\nu \left(\frac{\sqrt{2\nu}}{\ell} \|x - x'\| \right). \quad (\text{C.6})$$

The parameters κ , β , and τ of the differential equation are related to the Matérn covariance parameters as follows

$$\kappa = \frac{\sqrt{2\nu}}{\ell}, \quad \beta = \frac{\nu}{2} + \frac{d}{4}, \quad \tau^2 = \frac{\Gamma(\nu)}{\Gamma(\nu + d/2)(4\pi)^{d/2} \kappa^{2\nu}}, \quad (\text{C.7})$$

where $\Gamma(\cdot)$ is the Gamma function, and $K_\nu(\cdot)$ is the modified Bessel function of the second kind and d denotes the spatial dimension.

Finally, since the stochastic differential equation is linear and the random forcing is Gaussian, the solution is also a Gaussian field, which can be compactly written as

$$u(x) \sim \mathcal{GP} \left(0, \frac{1}{\tau^2} c_u(x, x') \right). \quad (\text{C.8})$$

For further details see [25, 27].

References

- [1] B. Liu, M. Ortiz, F. Cirak, Towards quantum computational mechanics, *Computer Methods in Applied Mechanics and Engineering* 432 (2024) 117403:1–117403:29.
- [2] D. E. Deutsch, Quantum computational networks, *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 425 (1989) 73–90.
- [3] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, H. Weinfurter, Elementary gates for quantum computation, *Physical Review A* 52 (1995) 3457–3467.
- [4] M. Möttönen, J. J. Vartiainen, V. Bergholm, M. M. Salomaa, Transformation of quantum states using uniformly controlled rotations, *Quantum Information & Computation* 5 (2005) 467–473.
- [5] X. Sun, G. Tian, S. Yang, P. Yuan, S. Zhang, Asymptotically optimal circuit depth for quantum state preparation and general unitary synthesis, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 42 (2023) 3301–3314.
- [6] M. Rosenkranz, E. Brunner, G. Marin-Sanchez, N. Fitzpatrick, S. Dilkes, Y. Tang, Y. Kikuchi, M. Benedetti, Quantum state preparation for multivariate functions, *Quantum* 9 (2025) 1703.
- [7] J. M. Martyn, Z. M. Rossi, A. K. Tan, I. L. Chuang, Grand unification of quantum algorithms, *PRX Quantum* 2 (4) (2021) 040203:1–040203:40.
- [8] A. W. Harrow, A. Hassidim, S. Lloyd, Quantum algorithm for linear systems of equations, *Physical Review Letters* 103 (2009) 150502.
- [9] M. E. S. Morales, L. Pira, P. Schleich, K. Koor, P. C. S. Costa, D. An, A. Aspuru-Guzik, L. Lin, P. Rebentrost, D. W. Berry, Quantum linear system solvers: A survey of algorithms and applications, *arXiv preprint arXiv:2411.02522* (2024).
- [10] D. Camps, R. Van Beeumen, FABLE: Fast approximate quantum circuits for block-encodings, in: 2022 IEEE International Conference on Quantum Computing and Engineering (QCE), 104–113, 2022.
- [11] D. Camps, L. Lin, R. Van Beeumen, C. Yang, Explicit quantum circuits for block encodings of certain sparse matrices, *SIAM Journal on Matrix Analysis and Applications* 45 (2024) 801–827.
- [12] L. Lapworth, Evaluation of block encoding for sparse matrix inversion using QSVT, *arXiv preprint arXiv:2402.17529* (2024).
- [13] C. Sünderhauf, E. Campbell, J. Camps, Block-encoding structured matrices for data input in quantum computing, *Quantum* 8 (2024) 1226.
- [14] L. N. Trefethen, *Spectral methods in MATLAB*, SIAM, 2000.
- [15] C. Canuto, M. Y. Hussaini, A. Quarteroni, T. A. Zang, *Spectral Methods: Fundamentals in Single Domains*, Springer, 2007.
- [16] H. Moulinec, P. Suquet, A numerical method for computing the overall response of nonlinear composites with complex microstructure, *Computer Methods in Applied Mechanics and Engineering* 157 (1998) 69–94.
- [17] M. Schneider, A review of nonlinear FFT-based computational homogenization methods, *Acta Mechanica* 232 (2021) 2051–2100.
- [18] C. Gierden, J. Kochmann, J. Waimann, B. Svendsen, S. Reese, A review of FE-FFT-based two-scale methods for computational modeling of microstructure evolution and macroscopic material behavior, *Archives of Computational Methods in Engineering* 29 (2022) 4115–4135.
- [19] L. Wright, C. Mc Keever, J. T. First, R. Johnston, J. Tillay, S. Chaney, M. Rosenkranz, M. Lubasch, Noisy intermediate-scale quantum simulation of the one-dimensional wave equation, *Physical Review Research* 6 (4) (2024) 043169:1–043169:10.
- [20] Y. Cao, A. Papageorgiou, I. Petras, J. Traub, S. Kais, Quantum algorithm and circuit design solving the Poisson equation, *New Journal of Physics* 15 (2013) 013021:1–013021:30.
- [21] M. Lubasch, Y. Kikuchi, L. Wright, C. M. Keever, Quantum circuits for partial differential equations in Fourier space, *arXiv preprint arXiv:2505.16895* (2025).
- [22] A. M. Childs, J.-P. Liu, A. Ostrander, High-precision quantum algorithms for partial differential equations, *Quantum* 5 (2021) 574.
- [23] H.-Y. Wu, A. E. Paine, E. Philip, A. A. Gentile, O. Kyriienko, Quantum algorithm for solving nonlinear differential equations based on physics-informed effective Hamiltonians, *arXiv preprint arXiv:2504.13174* (2025).
- [24] G. Strang, *Introduction to Applied Mathematics*, 1986.
- [25] F. Lindgren, H. Rue, J. Lindström, An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 73 (2011) 423–498.
- [26] F. Lindgren, D. Bolin, H. Rue, The SPDE approach for Gaussian and non-Gaussian fields: 10 years and still running, *Spatial Statistics* (2022) 100599.
- [27] K. J. Koh, F. Cirak, Stochastic PDE representation of random fields for large-scale Gaussian process regression and statistical finite element analysis, *Computer Methods in Applied Mechanics and Engineering* 417 (2023) 116358:1–116358:41.
- [28] I. Ben-Yelun, A. O. Yukseil, F. Cirak, Robust topology optimisation of lattice structures with spatially correlated uncertainties, *Structural and Multidisciplinary Optimization* 67 (2024) 16:1–16:19.
- [29] S. Woerner, D. J. Egger, Quantum risk analysis, *npj Quantum Information* 5 (2019) 1–15.
- [30] M. V. Wickerhauser, *Adapted wavelet analysis: from theory to software*, IEEE Press, 1994.
- [31] G. Strang, The discrete cosine transform, *SIAM Review* 41 (1999) 135–147.
- [32] A. Klappenecker, M. Rotteler, Discrete cosine transforms on quantum computers, in: ISPA 2001, 464–468, 2001.
- [33] S. Wang, X. Li, W. J. B. Lee, S. Deb, E. Lim, A. Chattopadhyay, A comprehensive study of quantum arithmetic circuits, *Philosophical Transactions A* 383 (2025) 20230392:1–20230392:37.
- [34] M. A. Nielsen, I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 10th anniversary edn., 2010.
- [35] P. Kaye, R. Laflamme, M. Mosca, *An Introduction to Quantum Computing*, Oxford University Press, 2006.
- [36] E. G. Rieffel, W. H. Polak, *Quantum Computing: A Gentle Introduction*, MIT Press, 2011.
- [37] T. G. Draper, Addition on a quantum computer, *arXiv preprint quant-ph/0008033* (2000).
- [38] M. A. Schalkers, M. Möller, Efficient and fail-safe quantum algorithm for the transport equation, *Journal of Computational Physics* 502 (2024) 112816:1–112816:25.
- [39] A. Javadi-Abhari, M. Treinish, K. Krsulich, C. J. Wood, J. Lishman, J. Gacon, S. Martiel, P. D. Nation, L. S. Bishop, A. W. Cross, B. R. Johnson, J. M. Gambetta, Quantum computing with Qiskit, *arXiv preprint arXiv:2405.08810* (2024).

- [40] S. Sivarajah, S. Dilkes, A. Cowtan, W. Simmons, A. Edgington, R. Duncan, *t|ket*: A Retargetable Compiler for NISQ Devices, arXiv preprint arXiv:2003.10611 (2020).
- [41] Y. Nam, Y. Su, D. Maslov, Approximate quantum Fourier transform with $O(n \log(n))$ T gates, NPJ Quantum Information 6 (2020) 26.
- [42] H.-Y. Huang, R. Kueng, J. Preskill, Predicting many properties of a quantum system from very few measurements, Nature Physics 16 (2020) 1050–1057.
- [43] C. A. Williams, S. Scali, A. A. Gentile, D. Berger, O. Kyriienko, Addressing the readout problem in quantum differential equation algorithms with quantum scientific machine learning, arXiv preprint arXiv:2411.14259 (2024).
- [44] L. Morin, J. Paux, A fast numerical method for the conductivity of heterogeneous media with Dirichlet boundary conditions based on discrete sine–cosine transforms, Computer Methods in Applied Mechanics and Engineering 421 (2024) 116772:1–116772:18.
- [45] L. Risthaus, M. Schneider, FFT-based computational micromechanics with Dirichlet boundary conditions on the rotated staggered grid, International Journal for Numerical Methods in Engineering 125 (2024) e7569:1–e7569:31.
- [46] J. Paux, L. Morin, L. Gélébart, A. M. A. Sanoko, A discrete sine–cosine based method for the elasticity of heterogeneous materials with arbitrary boundary conditions, Computer Methods in Applied Mechanics and Engineering 433 (2025) 117488:1–117488:22.
- [47] M. Deiml, D. Peterseim, Quantum Sampling and Moment Estimation for Transformed Gaussian Random Fields, arXiv preprint arXiv:2508.13879 (2025).