UAM Universidad Autónoma de Madrid | Escuela de Doctorado

Tesis Doctoral

Quantum computing and quantum-inspired numerical methods.

Application to problems in condensed matter physics and other fields.

Paula García Molina

Programa de Doctorado en Física de la Materia Condensada, Nanociencia y Biofísica

Dirección:
Juan José García Ripoll

Madrid, 2024

Para mi familia y Álvaro,
que, aunque a veces estén lejos,
siempre están conmigo.

"The most important step a man can take. It's not the first one, is it?
It's the next one. Always the next step, Dalinar."

— Brandon Sanderson, *Oathbringer*

Agradecimientos

“Long live all the mountains we moved
I had the time of my life fighting dragons with you.”

— Taylor Swift, *Long Live (Taylor’s version)*

La tesis que podéis leer a continuación es el resultado material de cuatro años de trabajo y marca el final de mi era como estudiante de doctorado. Sin embargo, este documento difícilmente refleja lo que ha sido esta experiencia, que va mucho más allá de escribir esta tesis. Quizás hayáis oído pronunciar las palabras “*Viaje antes que destino*”, que para mí son una forma de expresar este sentimiento. No solo es importante lo que conseguimos, sino también cómo lo hacemos y lo que nos llevamos por el camino. Si hubiera hecho esto sola no habría podido llegar al final; por ello, antes de hablar sobre mi investigación, me gustaría dedicar unas palabras a todas las personas que me han ayudado en este viaje.

Quiero empezar los agradecimientos con mi director de tesis, Juan José García Ripoll, que me ha enseñado cómo ser científica, además de a ser mejor programadora. Gracias por allanar mi camino siempre que encontraba dificultades, por ayudarme a mejorar y por estar disponible cuando necesitaba orientación. Quiero también reconocer a mis colaboradores del grupo de algoritmos de inspiración cuántica de QUINFOG, especialmente a Luca por su supervisión, así como a mis compañeros doctorandos, Juanjo y Jorge. Juanjo me dio la oportunidad de enseñar a alguien cómo utilizar mi trabajo y construir sobre él, y eso es algo que no tiene precio. Jorge me demostró que el trabajo en equipo es menos fome y se convirtió en un amigo con el que compartir éxitos y fracasos. Siempre estaré agradecida por su apoyo en estos últimos meses. También estoy muy agradecida con el resto de compañeros del grupo QUINFOG, quienes han creado un ambiente de trabajo acogedor del que me siento afortunada de formar parte. Me alegro de contar con un grupo lleno de estudiantes como yo, ya que compartir esta experiencia hace que sea más llevadera. Muchas gracias a María, quien rápidamente se convirtió en una amiga en esta nueva ciudad, por escucharme siempre que lo necesitaba y por hacer más fácil el reto de organizar un congreso. Me gustaría agradecer a Pablo por todos los buenos momentos, así como por su paciencia para aguantarme cuando las cosas no salían y para escucharme sin quejarse. Ojalá encuentre un compañero de despacho tan genial en mi siguiente etapa. Gracias también a mis otros compañeros, Luciano y Tomás—y recientemente Alejandro—, por compartir nuestro tiempo en el despacho y por los deliciosos dulces. A Cristian le agradezco su amabilidad, así como ser mi compañero otaku. No puedo dejar de mencionar a Carlos y Guillermo, en especial por organizar las actividades de divulgación que tanto disfruto, y a Gabriel, quien ha amenizado mis almuerzos con las mejores historias. Estoy muy agradecida a Alberto por ser tan atento, y, junto con Álvaro Rubio, haber sido una gran compañía. Gracias también a Álvaro Gómez, Adrián, Aleix y Jan. Nada de todo esto habría ocurrido sin Alejandro González Tudela, quien se acordó de mí en el momento adecuado.

Mi trabajo en el grupo QUINFOG me abrió las puertas del Max Planck Institut für Quantum Optik, en Múnich, donde realicé una estancia de tres meses. Esto no hubiera sido posible sin Mari Carmen Bañuls, quien me acogió bajo su tutela y, junto con David Pérez García, me ayudó a ampliar mis conocimientos y me descubrió nuevas formas de entender la investigación. Gracias a Andrea por ayudar a que todo saliera bien durante mi estancia, incluso aunque no fuera su trabajo, y a Marta y Denise por amenizar tantas horas juntas en el despacho. Danke, Alina, ich freue mich immer, dich zu sehen. Por último, gracias a Manu, con quien la vida me reunió de nuevo, esta vez en Múnich.

Investigar es solo una de las muchas tareas que he realizado durante mi doctorado. He tenido la oportunidad de participar en numerosos eventos de divulgación, así como de organizar dos congresos—¡gracias a mis compañeros del PYSQT!—y de asistir a muchos otros. Además, mi contrato FPU me ha permitido impartir clases en la Universidad Autónomca de Madrid. Convertirme en profesora universitaria era mi objetivo cuando comencé el doctorado, y me alegra decir que lo he conseguido. Gracias al Departamento de Física Teórica de la Materia Condensada por hacerlo posible.

Aunque el doctorado son cuatro años, el recorrido necesario para llegar hasta aquí ha sido mucho más largo. En el instituto, mis profesores Marta “Mates” y Ángel Cabedo me mostraron que este podía ser mi camino. Hace ya nueve años que empecé a estudiar Física en la Universidad de Córdoba, y puedo decir honestamente que no podría haber elegido un lugar mejor. Estoy profundamente agradecida a todos mis compañeros ucofísicos—especialmente a Álvaro, Borja, Toni, Pablo y Cristina que me proporcionaron tanto ayuda como buena compañía—por hacer de esta experiencia algo que siempre recordaré con cariño. Gracias a Mercedes Marín por enseñarme métodos numéricos; tenía razón cuando decía que iban a ser muy útiles. También quiero destacar a mis profesores de física cuántica: José Ignacio Palop, que siempre se preocupaba por nosotros, y a Antonio Cuesta, por esforzarse al máximo para ser el mejor profesor posible y seguir ayudándome incluso a día de hoy. Si algún día consigo ser profesora, espero ser para mis alumnos lo que Antonio fue para nosotros. En general, estoy muy agradecida a los profesores del grado de Física de la UCO, quienes me motivaron a querer serlo. Por último, no puedo olvidar a mis compañeros del Bilbopiso—Marina, Manu y Cobos—quienes hicieron más ameno mi paso por el máster y con los que también tuve la suerte de compartir buenos momentos en el grado.

Aunque doctorarse es un logro académico, la vida va mucho más allá, y para alcanzar mis metas ha sido clave rodearme de buenas personas que me han apoyado estos años. Quiero expresar mi gratitud a mi amiga y compañera Henar, así como al resto de mis amigos, Estrella, Álvaro Infantes, Laura, Juan, Ana, Dani y Raquel por su compañía estos años. También agradezco a Javi por preocuparse por mí desde la distancia. No olvido a mi gran familia, especialmente a mi abuela Esperanza y a mi tita Maribel; soy muy afortunada de teneros a todos. Para acabar, quiero destacar a las personas más importantes de mi vida, quienes siempre creen en mí y constituyen mi mayor apoyo. En primer lugar, muchas gracias a mis padres, Paqui y Jose, y a mis hermanos, María y Jose. Gracias por ser el hogar al que volver, por los incontables buenos momentos y risas, así como las interminables llamadas. Todo esto ha sido indispensable para seguir adelante y alcanzar mis objetivos más fácilmente. Por último, muchas gracias a Álvaro por acompañarme en todo momento, ya sea en persona o en la distancia. Gracias por escucharme y ayudarme siempre que lo necesito y, por supuesto, por hacerme más feliz.

Funding

This thesis was funded by MCIN/AEI/10.13039/501100011033 and “FSE invierte en tu futuro” through an FPU Grant FPU19/03590. The author acknowledges support from Spanish Project No. PGC2018-094792-B-I00 (MCIU/AEI/FEDER, UE), CAM/FEDER Project No. S2018/TCS-4342 (QUITEMAD-CM), Proyecto Sinérgico CAM 2020 Y2020/TCS-6545 (NanoQuCoCM), Spanish Projects No. PID2021-127968NB-I00 and No. PDC2022-133486-I00, funded by MCIN/AEI/10.13039/5011000 11033 and by the European Union “NextGenerationEU”/PRTR”1, and CSIC Interdisciplinary Thematic Platform (PTI) Quantum Technologies (PTIQTEP+). The author also gratefully acknowledges the Scientific computing Area (AIC), SGAI-CSIC, for their assistance while using the TRUENO and DRAGO Supercomputers for performing the simulations, and Centro de Supercomputación de Galicia (CESGA) for access to the supercomputer FinisTerrae.

Abstract

Partial differential equations (PDEs) are relevant for solving real-world problems across many areas. However, their solution may be challenging, especially for large-dimensional or high-resolution problems with high memory demands. This thesis develops new quantum and quantum-inspired numerical analysis methods for solving PDEs with potential memory and time savings while maintaining high accuracy. First, we resort to quantum computing, which benefits from exponential encoding advantages and speedups in key operations. Due to the lack of error correction of existing quantum computers, we propose a variational quantum algorithm to solve Hamiltonian PDEs, combining a classical and a quantum computer to exploit the properties of the quantum register. However, the noise sources and limited number of measurements of current quantum devices restrict the scalability of this approach. The high efficiency of the quantum register function encoding motivates its use in developing quantum-inspired algorithms. The second part of the thesis focuses on creating a matrix product state (MPS) finite precision algebra and applying it to quantum-inspired numerical analysis. More concretely, we develop MPS methods to solve static and time-dependent PDEs, motivated by the solution of problems of physical interest: the study of superconducting circuits and the expansion of a particle's wavefunction in the context of levitodynamics. Using a two-dimensional squeezed harmonic oscillator of up to 2^{30} points as a benchmark, MPS methods for Hamiltonian PDEs show exponential memory advantage compared to vector implementations and asymptotic advantage in time while achieving a low error in the solution approximation. Similarly, the time evolution MPS techniques demonstrate exponential memory compression and comparable accuracy and cost to standard vector methods. We conclude that the MPS framework constitutes a memory-efficient and accurate tool for solving PDEs. These findings present new opportunities for applying quantum-inspired algorithms to a wider range of PDEs and numerical analysis problems, opening exciting avenues for future research and applications.

Keywords: numerical analysis, partial differential equations, quantum computing, quantum-inspired, variational quantum algorithms, tensor networks, matrix product states

Resumen

Las ecuaciones en derivadas parciales (EDPs) son relevantes para resolver problemas del mundo real en diversas áreas. Sin embargo, su resolución puede ser un desafío, especialmente en problemas de gran dimensión o alta resolución, que requieren mucha memoria. Esta tesis desarrolla nuevos métodos de análisis numérico cuánticos y de inspiración cuántica para resolver EDPs, con el potencial de ahorrar memoria y tiempo, manteniendo una alta precisión. En primer lugar, recurrimos a la computación cuántica, que se beneficia de ventajas exponenciales en la codificación y de aceleraciones en operaciones clave. Debido a la falta de corrección de errores en los ordenadores cuánticos actuales, proponemos un algoritmo cuántico variacional para resolver EDPs Hamiltonianas, combinando un ordenador clásico y uno cuántico para aprovechar las propiedades del registro cuántico. Sin embargo, las fuentes de ruido y el número limitado de medidas en los dispositivos cuánticos actuales limitan la escalabilidad de esta técnica. La alta eficiencia en la codificación de funciones del registro cuántico motiva su uso en el desarrollo de algoritmos inspirados en la computación cuántica. La segunda parte de la tesis se centra en la creación de un álgebra de precisión finita de estados producto de matriz (MPS, por sus siglas en inglés) y su aplicación al análisis numérico inspirado en la computación cuántica. Concretamente, desarrollamos métodos de MPS para resolver EDPs estáticas y dependientes del tiempo, motivados por la resolución de problemas de interés físico: el estudio de circuitos superconductores y la expansión de la función de onda de una partícula en el contexto de la levitodinámica. Utilizando un oscilador armónico bidimensional de hasta 2^{30} puntos como referencia, los métodos MPS para EDPs Hamiltonianas muestran una ventaja exponencial en memoria en comparación con las implementaciones vectoriales y una ventaja asintótica en tiempo, logrando a su vez un bajo error en la aproximación de la solución. De manera similar, las técnicas de MPS para evolución temporal demuestran una compresión exponencial de memoria y una precisión y coste comparables a los métodos vectoriales estándar. Concluimos que los MPS constituye una herramienta eficiente en memoria y precisa para resolver EDPs. Estos hallazgos presentan nuevas oportunidades para aplicar algoritmos inspirados en la computación cuántica a una gama más amplia de EDPs y problemas de análisis numérico, abriendo emocionantes vías para futuras investigaciones y aplicaciones.

Palabras clave: análisis numérico, ecuaciones en derivadas parciales, computación cuántica, inspiración cuántica, algoritmos variacionales cuánticos, redes de tensores, estados producto de matriz

Works in this thesis

Publications

- Paula García-Molina, Javier Rodríguez-Mediavilla, and Juan José García-Ripoll. Quantum Fourier analysis for multivariate functions and applications to a class of Schrödinger-type partial differential equations, *Physical Review A* 105 (1), 012433, 2022.

- Paula García-Molina, Luca Tagliacozzo, and Juan José García-Ripoll. Global optimization of MPS in quantum-inspired numerical analysis, *arXiv e-prints*, arXiv:2303.09430, 2023.

- Jorge Gidi, Paula García-Molina, Luca Tagliacozzo, and Juan José García-Ripoll. Pseudospectral method for solving PDEs using Matrix Product States, *arXiv e-prints*, arXiv:2409.02916, 2024.

Contributions to conferences

- Oral contribution *Solving partial differential equations in quantum computers* at the 6th Quantum Information Conference in Spain (ICE-6), 2021.

- Oral contribution *Variational quantum algorithm for eigenvalue problems of a class of Schrödinger-type partial differential equations* at Machine Learning for Quantum X (MLQX), 2021.

- Poster contribution *Variational quantum algorithm for eigenvalue problems of a class of Schrödinger-type partial differential equations* at Munich Conference of Science and Technology (MCQST), 2021.

- Poster contribution *Variational quantum algorithm for eigenvalue problems of a class of Schrödinger-type partial differential equations* at the Tensor Network based approaches to Quantum Many-Body Systems ICCUB School, 2021.

- Poster contribution *Variational quantum algorithm for eigenvalue problems of a class of Schrödinger-type partial differential equations* at QCE 2021 IEEE Quantum Week, 2021.

- Poster contribution *Variational quantum algorithm for eigenvalue problems of a class of Schrödinger-type partial differential equations* at EQTC, 2021.

- Invited oral contribution *Introducción a la computación cuántica. Implementación de algoritmos cuánticos y aplicaciones* at SALMOREJOTECH, 2022.

- Oral contribution *Quantum Fourier analysis for multivariate functions and applications to a class of Schrödinger-type partial differential equations* at XXXVIII Biennial of Physics of the Spanish Royal Physics Society (R.S.E.F.), 2022.

- Poster contribution *Quantum and quantum-inspired algorithms for numerical analysis* at the first meeting of the International Quantum Tensor Network, 2022.

- Oral contribution *Global optimization of MPS in quantum-inspired numerical analysis* at Quantum Matter, 2023.

- Poster contribution *Global optimization of MPS in quantum-inspired numerical analysis* at the 8th Quantum Information Conference in Spain (ICE-8), 2023.

- Poster contribution *Global optimization of MPS in quantum-inspired numerical analysis* at the third meeting of the International Quantum Tensor Network, 2023.

- Poster contribution *SElf-Explaining Matrix-Product-State (SeeMPS) library* at the fourth meeting of the International Quantum Tensor Network, 2024.

Software

One of the main results of the thesis is the contributions to the SElf-Explaining Matrix-Product-State (SeeMPS) library. This is a Python-based library accelerated with Cython for implementing matrix product state algorithms, with a special focus on numerical analysis applications.

Other works

- Juan José Rodríguez-Aldavero, Paula García-Molina, Luca Tagliacozzo, Juan José García-Ripoll. Chebyshev approximation and composition of functions in matrix product states for quantum-inspired numerical analysis, *arXiv e-prints*, arXiv:2407.09609, 2024.
- Paula García-Molina, Ana Martin, Mikel Garcia de Andoin, Mikel Sanz. Mitigating noise in digital and digital-analog quantum computation, *Communications Physics*, 321, 2024.

Contents

Appendices

List of Tables

List of Figures

Chapter 1

Introduction

"The purpose of a storyteller is not to tell you how to think, but to give you questions to think upon."

— Brandon Sanderson, *The Way of Kings*

Partial differential equations (PDEs) constitute a relevant tool for the mathematical modeling of real-world problems. Notable examples include the Navier-Stokes equations, which govern fluid dynamics and have applications in engineering, such as optimizing the aerodynamics of Formula 1 cars; the SIR model in medicine used to predict the spread of infectious diseases; and the Black-Scholes equation in finance, which is fundamental in option pricing. In physics, PDEs are important for the study of quantum technologies, including the study of superconducting circuits [1, 2, 3] described by Hamiltonian PDEs, and the time evolution problem in levitodynamics [4] governed by the time-dependent Schrödinger equation. Solving these PDEs can be challenging due to their high dimensionality and fine resolution—number of points to correctly capture the problem—, leading to exponential memory and implementation costs.

The exponential advantage in information processing and the speedups offered for relevant operations, such as the Fourier transform, make quantum computers an interesting tool for solving numerical analysis problems. A particularly relevant class of quantum algorithms for solving PDEs [5, 6, 7, 8] is based on the Harrow-Hassidim-Lloyd (HHL) algorithm [9] for linear systems of equations. These methods have addressed problems like the Poisson [10] and the heat equation [11]. HHL-based techniques for PDEs, as well as other proposals [12, 13, 14, 15, 16] based on quantum amplitude estimation (QAE) [17], require fault-tolerant quantum computing. However, we find ourselves in the so-called

Noisy Intermediate-Scale Quantum (NISQ) [18] era, characterized by quantum computers with up to a few hundred physical qubits and no error correction [19], rendering the previous methods unfeasible. Hybrid quantum-classical algorithms [20] appear as an alternative to take advantage of present quantum devices. These algorithms combine classical and quantum computers, resulting in methods with lower hardware requirements and higher noise resilience [20, 21]. These promising characteristics led to the application of variational quantum algorithms [22] to quantum numerical analysis [23, 24, 25, 26, 24, 27, 28], solving a wide range of problems: the Feynman-Kac equation [29] and other stochastic PDEs [30], the Poisson equation [31, 32], the advection-diffusion equation [33], the heat equation [34], and fluid dynamics [35, 36], among others.

Quantum-inspired methods are classical algorithms that can achieve memory compression and speedups similar to those of quantum algorithms while avoiding the current limitations of quantum computers. These techniques may rely on tensor networks [37], an exponentially efficient formalism to represent quantum states. Tensor networks arise in the field of quantum many-body physics to overcome the *curse of dimensionality*, i.e., the exponential scaling of the number of parameters with the system size. This issue also poses a problem for resolving PDEs, motivating the use of tensor networks to tackle them. In quantum-inspired numerical analysis, functions and operators may be represented using one-dimensional tensor networks, known as matrix product states (MPS) and operators (MPOs). García-Ripoll [38] demonstrated the efficient encoding of highly differentiable multidimensional functions as MPS and the consequent speed up of numerical tasks like interpolation and solving PDEs. Later, Jobst and collaborators [39] formalized the error of the MPS representation of functions with a rapidly decaying Fourier spectrum. Similar tensorial approximations were developed in mathematics under the name of tensor trains (TTs), with their quantized version (QTTs) being equivalent to MPS. These representations have been successfully used to solve a variety of PDEs including the Hamilton-Jacobi-Bellman equations [40, 41, 42], high-dimensional nonlinear PDEs [43], the nonlinear Schrödinger equation [44], and turbulence problems [45]. Overall, these quantum-inspired methods may constitute powerful tools for addressing the computational complexity of PDEs, appearing as encouraging alternatives to both classical and quantum methods.

In this thesis, we propose leveraging quantum and quantum-inspired numerical analysis to offer potential scalability advantages while maintaining high accuracy for the solution of PDEs. Impeled by the high complexity of physically motivated PDEs, we construct encodings of functions and PDEs as quantum states and operators, acting as a basis for new, possibly more efficient methods to address these problems.

In the first part of the thesis, we focused on studying the application of quantum computers to solve PDEs. This resulted in a novel variational quantum algorithm [46] to solve static Schrödinger-type Hamiltonian PDEs. This algorithm relies on a highly efficient encoding of functions and operators in a quantum register based on Fourier techniques. It is combined with new suitable *Ansätze* [47, 48] for representing functions and their symmetries and various classical optimizers. This method is used to solve the Hamiltonian PDE of different superconducting qubits, leveraging the potential of quantum computers to model their very components. Despite the high accuracy of the created encoding, the noise and limited number of measurements of current quantum computers restrict the performance of this variational approach. This suggests the need for alternative techniques that could benefit from this quantum representation of functions.

Quantum-inspired techniques appear to be a suitable option for taking advantage of the quantum register encoding while avoiding errors arising from noise sources. By mapping the register's qubits to the tensors of an MPS, it is possible to construct efficient representations of functions [38]. This realization motivated the second part of the thesis, in which we created new quantum-inspired algorithms to resolve PDEs using a finite precision MPS framework. We used this binary encoding to develop novel global optimization algorithms to resolve Hamiltonian PDEs [49]. This work integrates the MPS-MPO representation of functions and operators with a set of basic algebraic operations and MPS truncation algorithms, resulting in an MPS-MPO finite precision algebra. This algebra enables working with MPS-MPO similarly to the elements of the matrix-vector algebra, providing a powerful tool for developing quantum-inspired algorithms for numerical analysis. The one- and two-dimensional harmonic oscillator Hamiltonian PDE—one of the quintessential equations of quantum mechanics—acted as a benchmark for different optimization algorithms, using a grid of up to $2^{30} \approx O(10^9)$ points. This work demonstrated that MPS asymptotically exhibit exponential advantages in memory and significant time savings compared to standard vector implementations for solving Hamiltonian PDEs.

Our previous work confirmed the advantages of using MPS for solving PDEs, thanks to their excellent memory savings. We then studied how this performance extends from static problems to time-dependent ones. Our focus is the field of levitodynamics, which presents challenging problems from a numerical perspective. More specifically, when a trapped particle is released from a tightly confined potential, its wavefunction may experience a rapid expansion, increasing in size two or more orders of magnitude in a brief period of time. This is a challenging problem, both because it requires a precise modelization with a very dense representation, and also because we need to solve the dynamics accurately. We addressed this problem with a new MPS pseudospectral method for solving time-dependent PDEs [50]. This work introduced a highly accurate technique to approximate functions of derivatives, which is combined with global evolution schemes independent of the locality of interactions to solve the particle's evolution successfully. MPS's error scaling and run times are comparable to FFT vector methods; however, MPS have a significant advantage in memory, overcoming vector limitations to facilitate larger discretizations and expansions. This result restated that the MPS-MPO finite precision algebra is an efficient framework for developing quantum-inspired numerical analysis techniques to solve PDEs effectively with high memory usage.

Finally, the SElf-Explaining Matrix Product State (SeeMPS) [51] library includes the MPS algebra and algorithms developed in this thesis and constitutes a valuable tool for addressing numerical analysis problems from a quantum-inspired perspective. We have developed around 12 algorithms for the numerical approximation of differentiation, interpolation, and partial differential equations. These algorithms are based on an MPS-MPO finite precision algebra that enables the encoding of multidimensional problems and an accurate representation of operators. The SeeMPS library is a growing open-source project that has incorporated algorithms for the loading of functions and operators [52] based on Chebyshev [53, 54] and TT-cross interpolation [55]. The library is not only freely available, but it supports all the quantum-inspired papers developed in this thesis and is also described in this manuscript.

The structure of this thesis is as follows. The thesis starts by introducing the representation of functions and operators in a quantum register in chapter 2. Then, chapter 3 focuses on tensor networks, particularly MPS, and how they can be used to construct a finite precision algebra for developing quantum-inspired algorithms. Once the primary building blocks are presented, the following sections introduce the quantum and quantum-inspired algorithms for numerical analysis created in this thesis. Chapter 4 covers the work in Ref [46]: a variational quantum algorithm to solve static Schrödinger-type Hamiltonian PDEs applied to the simulation of superconducting circuits. Chapter 5 uses the finite precision algebra to create quantum-inspired algorithms for global optimization of MPS. These algorithms, proposed in Ref. [49], solve the harmonic oscillator's Hamiltonian PDE in one and two dimensions. The dramatic expansion of the particle's wavefunction in a levitodynamics setting motivates the creation of MPS algorithms for time-dependent PDEs in chapter 6, as described in Ref. [50]. The basic functioning of the SeeMPS library is described in chapter 7. Chapter 8 summarizes the main contributions of this work and points out future research directions. The contents of this thesis are outlined in Table 1.1.

		Section
Problems of study	Harmonic oscillator	4.1 and 5.3
	Superconducting qubits	4.1
	Levitodynamics	6.1
Differentiation	Finite differences†	2.2.2
	Spectral method*,†	2.2.2
	Hermite Distributed Approximating Functionals†	6.2
Interpolation	Finite differences†	2.3
	Fourier interpolation*,†	2.3
MPS algorithms	Canonical form	3.3
	Expected value	3.5
	MPO-MPS product	3.5
	MPS-MPS product	3.5
	Scalar multiplication	3.5
	Simplification	3.5
	Linear combination	3.5
Optimization	Variational quantum PDE solver*	4.4
	Gradient descent†	5.1.2
	Arnoldi iteration†	5.1.4
	Power iteration†	5.1.5
	DMRG	5.1.6
Evolution	Euler†	5.1.1
	Improved Euler†	5.1.1
	Runge-Kutta†	5.1.1
	Runge-Kutta-Fehlberg†	5.1.1
	Crank-Nicolson†	6.3.1
	Arnoldi iteration†	6.3.2
	Split-step†	6.3.3

Table 1.1: Main contents of the thesis. The table gathers the problems of study and the different quantum and quantum-inspired numerical analysis techniques used to address them. The symbol * indicates that the algorithm is quantum, while † stands for quantum-inspired. The grey color represents non-original algorithms.

Chapter 2

Quantum register representation of functions

In the domain of numerical analysis, a very relevant task is to represent and construct functions defined over continuous domains. Let us take a canonical problem, which is solving the heat transfer equation $\nabla^2 f(\mathbf{x}) = g(\mathbf{x})$, where $\mathbf{x} \in \Omega$ is a multidimensional variable that runs along some compact volume in $\Omega \subset \mathbb{R}^N$, with suitable boundary conditions. Since computers cannot deal with infinite-dimensional degrees of freedom, the numerical solution of such a problem must start by finding a suitable, finite-dimensional representation of the functions $f(\mathbf{x})$ and $g(\mathbf{x})$. Typical approaches include sampling the values of these functions over a regular or optimized multidimensional grid or expanding those functions in a basis of functions—e.g., Fourier modes, Hermite or Chebyshev polynomials, finite element polynomials—. Any of these embeddings transforms the problem into a discrete version that is amenable to treatment in a computer. However, the size of the embedding and the accuracy in the solution of the associated problem can pose severe challenges in many relevant situations.

In this scenario, quantum computers have been proposed as an alternative platform that can provide advantages in both encoding and solving numerical analysis problems. Starting with early results for coupled linear equations—the HHL algorithm [9]— it is clear that a central advantage of the use of quantum computers lies in leveraging the exponentially large Hilbert space, i.e., a quantum register with n qubits may potentially represent 2^n degrees of freedom associated with any of the previous function encodings. These techniques result in a highly efficient framework for solving numerical analysis problems in quantum computers (Chapter 4) that also extends to quantum-inspired settings (Chapters 5 and 6), more accurate and efficient in the NISQ era [18].

2.1 Function discretization

A quantum register can encode a function by dividing the position space into discrete intervals and mapping them to the register's states. The one-dimensional case serves as an example to understand this encoding, which is also valid for multiple dimensions (Section 2.4), inspired by the representation in Refs. [47, 48]. Let us consider a one-dimensional function $f(x)$ defined on the finite interval $[a, b)$ of size $L_x = |b - a|$. A 2^n-point discretization of such interval leads to

$$x_s^{(n)} = a + s\Delta x^{(n)}, \tag{2.1}$$

with $s \in \{0, 1, \dots, 2^n - 1\}$ and $\Delta x^{(n)} = \frac{L_x}{2^n}$. Each integer s corresponds to a quantum state $s = (s_0 s_1 \dots s_{2^n-1})$, leading to a binary representation of the coordinate values

$$x_s^{(n)} = x_{(s_0 s_1 \dots s_{2^n-1})}^{(n)} = a + \Delta x^{(n)} \sum_{k=0}^{n-1} s_k 2^{n-k-1}. \tag{2.2}$$

Then, a quantum register represents a function as a normalized quantum state,

$$\left| f^{(n)} \right\rangle = \frac{1}{\mathcal{N}_f^{1/2}} \sum_{s=0}^{2^n-1} f\left(x_s^{(n)}\right) |s\rangle\,, \tag{2.3}$$

where $\mathcal{N}_f$ is the normalization constant[1]. Equation (2.3) encodes a function in a quantum register as a linear superposition of basis quantum states, in which the amplitudes represent the value of the function at the coordinate $x_s^{(n)}$ associated with each state. This representation of functions in a quantum register requires only a logarithmically small number of qubits $n = O(\log_2(1/\Delta x))$, resulting in an exponential advantage compared to the classical approach. This advantage translates into further improvements in the representation errors, as discussed in section 2.5.

As an example of the encoding, Figure 2.1 represents a Gaussian function with zero mean $\mu = 0$ and standard deviation $\sigma = 1$,

$$f(x) = \frac{1}{\mathcal{N}_f^{1/2}} e^{-x^2/2}, \tag{2.4}$$

[1] The quantum computing formalism requires quantum states to be normalized. Tensor networks present an alternative platform for this quantum register representation that overcomes this limitation (Chapter 3).

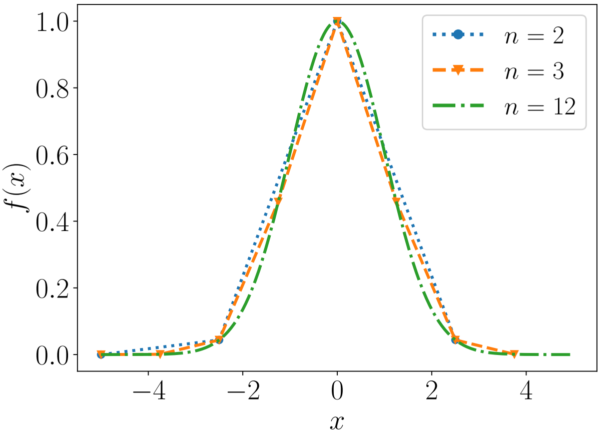


Figure 2.1: Encoding of a Gaussian function in a quantum register. Gaussian function (2.4) quantum register discretization (2.3) in a 2^n-point grid (2.1) with $L_x = 10$, where $n = 2, 3, 12$ is the number of qubits of the quantum register. The normalization constant is removed for the sake of visual comparison.

where $\mathcal{N}_f$ is the normalization constant. Let us discretize this function in $n = 2$ qubits, which corresponds to a 4-point discretization

$$f^{(n)} = [f(x_0), f(x_1), f(x_2), f(x_3)]^T. \tag{2.5}$$

The quantum register encoding stores each value in the amplitudes of the quantum register as

$$\left|f^{(n)}\right\rangle = \frac{1}{\mathcal{N}_f^{1/2}} \left(f(x_0)\left|0\right\rangle + f(x_1)\left|1\right\rangle + f(x_2)\left|2\right\rangle + f(x_3)\left|3\right\rangle\right) \tag{2.6}$$

$$= \frac{1}{\mathcal{N}_f^{1/2}} \left(f(x_{00})\left|00\right\rangle + f(x_{01})\left|01\right\rangle + f(x_{10})\left|10\right\rangle + f(x_{11})\left|11\right\rangle\right). \tag{2.7}$$

Adding a qubit to the discretization doubles the number of points (Figure 2.1), showing the exponential efficiency of the quantum register encoding.

This representation of functions in a quantum register is a particular case of a general embedding of classical data into a quantum framework, commonly called amplitude encoding. Amplitude encoding is a popular method in quantum machine learning [56, 57, 58, 59] and quantum numerical analysis [47, 48, 9, 60]. Among the many applications of this encoding of functions, chapter 4 shows how to solve Schrödinger-like static

Hamiltonian PDEs in ideal and NISQ [18] scenarios. Interestingly, this quantum representation can be leveraged in classical computers with the help of quantum-inspired methods. More precisely, as discussed through this thesis, the combination of a quantum register encoding with tensor networks allows exponential gains in memory and execution tasks for many numerical analysis algorithms—e.g., solution of static PDEs in chapter 5 and solution of time-dependent PDEs in chapter 6.

2.2 Operator discretization

The discretization of functions on a grid can induce different representations of operators—e.g., potentials and derivatives—as linear or non-linear transformations of the discretized information. In the particular case of a quantum register encoding, it is possible to represent those operators as linear transformations of the quantum register. Sometimes those transformations correspond to physically relevant operators—e.g., Hermitian operators that can generate some dynamical evolution or become part of a Hamiltonian—, but more often they will not, in which case one needs some approximation technique—see variational methods in chapter 4 —or use quantum-inspired methods that are not restricted to physical operations—e.g. tensor network methods in chapters 5 and 6.

2.2.1 Potential operators

In this thesis, we will study PDEs associated with the Hamiltonian of different quantum systems. The Hamiltonian $H = D(-i\partial_x) + V(x)$ consists of the kinetic and potential terms, respectively. The kinetic term corresponds to the differential operator of the PDE (Section 2.2.2), while the potential operator is a function of the position. Therefore, the application of the potential operator is equivalent to the multiplication by a continuous function $V(x)$ written as the diagonal operator

$$\hat{V}(x) = \sum_s V\left(x_s^{(n)}\right) |s\rangle\langle s|. \tag{2.8}$$

The fact that the operator is diagonal is very convenient since its application is equivalent to elementwise vector multiplication, which enables a better operation scaling for its tensor network implementation, as discussed in section 3.7.

2.2.2 Differential operators

To apply a differential operator, finding an appropriate method to implement the derivative of a discretized function is essential. One commonly used technique is the finite difference method, which can approximate differential operators for differentiable functions with an error that reduces polynomially in Δx. Alternatively, Fourier techniques can exponentially increase the accuracy of the approximation at the expense of greater complexity.

Finite difference method

This method uses different orders of the Taylor expansion of the function to approximate the derivatives with an algebraically decreasing order $O(\Delta x^p)$ that depends on the order p of the approximation. Two of the most common expressions are the centered finite difference approximation for the first and second derivative of $f(x)$,

$$\frac{\partial f(x)}{\partial x} = \frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x} + O(\Delta x^2), \tag{2.9}$$

$$\frac{\partial^2 f(x)}{\partial x^2} = \frac{f(x+\Delta x) - 2f(x) + f(x-\Delta x)}{\Delta x^2} + O(\Delta x^2). \tag{2.10}$$

The classical approach implements the displacements as shifts of vectors that encode the discretized functions. In the quantum representation, the displacement operators have a non-Hermitian representation that shifts the quantum register states. When working with open boundary conditions, this non-physical transformation is

$$\hat{\Sigma}^{+}|s\rangle = \begin{cases} |s+1\rangle & s < 2^n - 1, \\ 0 & \text{else,} \end{cases} \qquad \hat{\Sigma}^{-} = \left(\hat{\Sigma}^{+}\right)^{\dagger}, \tag{2.11}$$

which performs a displacement over the quantum register states. This operation is unitary when considering periodic boundary conditions, leading to the upper and lower diagonal matrices

$$\hat{\Sigma}^{+} = \begin{pmatrix} 0 & & & 1 \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{pmatrix}, \quad \hat{\Sigma}^{-} = \begin{pmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ 1 & & & 0 \end{pmatrix}. \tag{2.12}$$

Since the operations are unitary, they are amenable to its application on a quantum register. Open boundary conditions correspond to non-unitary operators and cannot be applied in physical quantum computers.

However, they are still suitable for the quantum-inspired representation in chapters 5 and 6. Using a linear combination of displacement operators (2.11), the differential operator implements the first and second derivative of the function encoded in the quantum register as [38]

$$|\partial_x f^{(n)}\rangle \simeq \frac{1}{2\Delta x^{(n)}} \left(\hat{\Sigma}^{+} - \hat{\Sigma}^{-}\right) \left|f^{(n)}\right\rangle, \tag{2.13}$$

$$|\partial_x^2 f^{(n)}\rangle \simeq \frac{1}{\left(\Delta x^{(n)}\right)^2} \left(\hat{\Sigma}^{+} - 2\mathbb{I} + \hat{\Sigma}^{-}\right) \left|f^{(n)}\right\rangle. \tag{2.14}$$

Spectral method

Fourier analysis techniques provide a highly efficient encoding with an exponential increase in precision (Section 2.5) compared to the finite difference method. These tools are suitable for periodic functions—naturally, this includes functions that vanish at all orders of their derivatives towards the boundaries of the function's domain. For the spectral encoding to succeed, these functions must be bandwidth-limited, meaning their Fourier transform $\tilde{f}(p) = [\hat{\mathcal{F}} f](p)$ should be negligible outside a moderate size volume in momentum or frequency space $[-L_p/2, L_p/2)$. According to the Nyquist-Shannon sampling theorem [61, 62], bandwidth-limited functions can be efficiently sampled over a finite grid with a spacing $\Delta x < 2\pi/L_p$. The discretized Fourier transform (DFT) substitutes its continuous version in this situation. The DFT of $f(x_s)$ discretized on a grid of N points is given by

$$\tilde{f}(p_l) = \frac{1}{\sqrt{N}} \sum_{s=0}^{N-1} e^{-2\pi i l s/N} f(x_s), \quad l = \{-N/2, \dots, N/2-1\}. \tag{2.15}$$

The inverse DFT is given by

$$f(x_s) = \frac{1}{\sqrt{N}} \sum_{l=-N/2}^{N/2-1} e^{2\pi i l s/N} \tilde{f}(p_l), \quad s = \{0, \dots, N-1\}, \tag{2.16}$$

where we conveniently reorder the elements of the sum[2]. This definition is valid for N even, which is satisfied for an n-qubit quantum register encoding with $N = 2^n$. Let us note that the original domain of variable x will be referred to as position space, and the Fourier transformed domain of variable p will be indistinctly called frequency or momentum space.

[2] Note that the Fourier transform is periodic as $e^{2\pi i l s/N} = e^{2\pi i (l+N)s/N}$, and hence the indices can be translated, mapping the highest frequency components to the corresponding negative frequencies.

The encoding in Eq. (2.16) is the basis for Fourier interpolation of a discretized function over its definition domain. This definition enables the approximation of arbitrary differential operators $G(\partial_x)$. Thus, in general

$$G(\partial_x)f(x) = \frac{1}{\sqrt{N}} \sum_{l=-N/2}^{N/2-1} G(ip_l)e^{ip_l x}\tilde{f}(p_l). \tag{2.17}$$

This expression only involves two DFTs (a direct and inverse one) and an elementwise multiplication of the function by complex coefficients. With the help of the fast Fourier transform algorithm (FFT), this computation may be performed in $O((2\log_2 N + 1)N)$ steps in a computer.

The quantum Fourier transform (QFT) $\hat{\mathcal{F}}$ allows us to represent differential operators $G(\partial_x)$ as operators acting on a quantum register. The QFT is a unitary operator that transforms an n-qubit quantum state as[3]

$$|r\rangle \overset{\hat{\mathcal{F}}}{\longmapsto} \frac{1}{\sqrt{2^n}} \sum_{s=0}^{2^n-1} e^{i2\pi rs/2^n} |s\rangle . \tag{2.18}$$

When acting on a linear superposition representing a function, $\left|f^{(n)}\right\rangle$, the QFT produces a new quantum state that encodes the inverse DFT of the function,

$$\begin{aligned} \left|\tilde{f}^{(n)}\right\rangle = \sum_s \tilde{f}^{(n)}\left(p_s^{(n)}\right)|s\rangle := \hat{\mathcal{F}}\left|f^{(n)}\right\rangle \\ = \frac{1}{\sqrt{2^n}} \sum_{r,s=0}^{2^n-1} e^{i2\pi sr/2^n} f\left(x_r^{(n)}\right)|s\rangle . \end{aligned} \tag{2.19}$$

Note that, due to the encoding of non-negative numbers in the quantum register, the negative frequencies from the DFT are now stored in the largest integer states of the quantum register (what is known as complement-2 representation in computer science),

$$p_s = \frac{2\pi}{\Delta x^{(n)} 2^n} \times \begin{cases} s & \text{for } 0 \leqslant s < 2^{n-1}, \\ s - 2^n & \text{otherwise.} \end{cases} \tag{2.20}$$

Interestingly, this same problem arises when using the FFT to implement a DFT in a classical computer, and it is something to be considered when interpreting Eq. 2.17.

The number of gates of both the QFT and inverse QFT circuits grows logarithmically with the discretization grid $O(n^2)$ (Figure 4.5), resulting

[3] Note that the sign convention is opposite to the DFT.

in an exponential advantage over the classical fast Fourier transform with $O(n2^n)$ operations. The smaller error of the spectral method compared to the finite difference—up to doubly exponential in the number of qubits (Section 2.5)—together with the exponential advantage of the quantum implementation, motivates the use of this technique [38] for the application of differential operators of any order as diagonal operators in Fourier space using

$$\hat{D}(-i\partial_x) := \hat{\mathcal{F}}^{-1} \sum_{\{s\}} D\left(p_s^{(n)}\right) |s\rangle\langle s| \, \hat{\mathcal{F}}. \tag{2.21}$$

The QFT is indeed one of the most advantageous quantum routines when compared to their classical counterpart, and it has many applications, from Shor's algorithm [63] for prime numbers factorization to quantum machine learning [64] and the solution of linear systems of equations [9].

2.3 Interpolation

Some computations require us to provide an estimate for values on all points of the continuous domain on which a function is defined. An interpolation scheme is an algorithm that offers such an estimate starting from the values of a discretized function on the original grid. This thesis relies on two types of interpolation formulas, depending on whether finite difference or spectral methods approximate derivatives.

The first and simplest method uses a piecewise linear (or polynomial) interpolation of the function between points of the discretization grid. Given an n-qubit representation of a function $f\left(x_s^{(n)}\right)$ with a discretization step $\Delta x_s^{(n)}$, the finite difference interpolant approximates the new points of the $(n+1)$-qubit grid by the Taylor expansion

$$f\left(x_s^{(n)} + \varepsilon\right) = f\left(x_s^{(n)}\right) + \varepsilon \partial_x f\left(x_s^{(n)}\right) + O\left(\left(\Delta x^{(n)}\right)^2\right), \tag{2.22}$$

where $\varepsilon \in [-\Delta x^{(n)}, \Delta x^{(n)}]$. Substituting $\varepsilon = \Delta x/2$ in Eq. (2.22), the second-order finite difference interpolation formula is given by

$$f\left(x_s^{(n)} + \Delta x^{(n)}/2\right) \approx f\left(x_s^{(n)}\right) + \frac{f\left(x_s^{(n)} + \Delta x^{(n)}\right) - f\left(x_s^{(n)}\right)}{2}. \tag{2.23}$$

This operation can be applied in the quantum register representation by replacing the displacements of the function with the displacement operators (2.11). Higher-order finite difference approximations include more displacements of the quantum register. These techniques easily translate to a quantum-inspired setting due to the representation of displacement operators as matrix product operators (MPOs).

Fourier interpolation [65, 38] is a second method with more advantageous error scalings—up to doubly exponentially in the number of qubits—for bandwidth-limited functions that can be resampled according to the Nyquist-Shannon theorem [61, 62]. Given a sufficiently small spacing $\Delta x^{(n)} \leqslant 2\pi/L_p$, this theorem ensures that any bandwidth-limited function $f(x)$ can be interpolated from a discretization $\left|f^{(n)}\right\rangle$, up to exponentially small errors. It states that a function with a maximum bandwidth $p_{\max}$ can be completely determined by its samples taken at a rate $p_l = 2p_{\max}$. If the sampling is made with a smaller frequency rate, the loss of information due to aliasing does not permit proper reconstruction of the function. For a function with domain sizes L_x and L_p in position and momentum space, respectively, their spaces need to satisfy $\Delta x^{(n)} \leqslant 2\pi/L_p$ and $\Delta p^{(n)} \leqslant 2\pi/L_x$.

Given a function (Figure 2.2(a)), the basic idea of Fourier interpolation is to transform the function to Fourier space (Figure 2.2(b)) and add new points as zeros (zero padding) in momentum space outside the bandwidth-limited interval (Figure 2.2(c)), since $\tilde{f}(p_l) \approx 0$ for $|p_l| > p_{\max}$. Then, the application of the inverse Fourier transform leads to the interpolated function in position space (Figure 2.2(d)). Algorithm 1 shows the pseudocode to implement Fourier interpolation using the FFT (see Appendix A for a step-by-step explanation and a Python implementation of Fourier interpolation).

Algorithm 1 Fourier interpolation pseudocode.

```
function FourierInterpolation($f$,$M$)
    Interpolate a function $f$ to $M$ points
    $N$ = len($f$)                              ▷ Original number of elements
    $F$ = fft_transform($f$)                    ▷ Perform Fourier transform
    $F_{\text{padded}}$ = pad_with_zeros($F$,$M$-$N$)      ▷ Add $M - N$ points
    $f_{\text{int}}$ = ifft_transform($F_{\text{padded}}$)   ▷ Undo shift and transform to position space
    return sqrt($M/N$)*$f_{\text{int}}$         ▷ Return rescaled interpolated function
end function
```

Fourier interpolation constitutes a unitary operation in the quantum register representation, unlike the finite difference approximation, where the displacement operators are only unitary for periodic boundary condi-

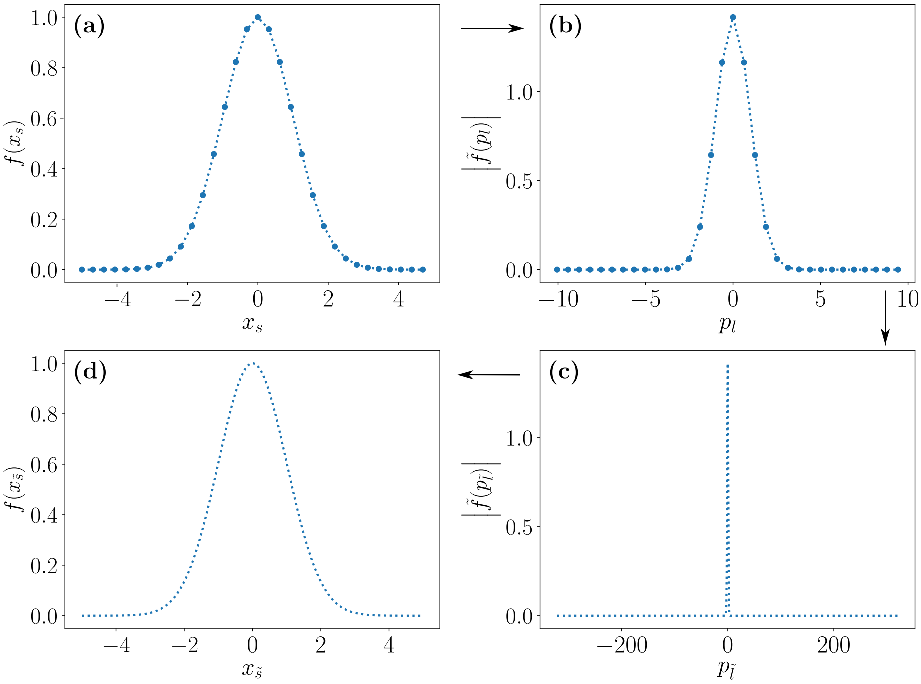


Figure 2.2: Fourier interpolation of a Gaussian function. (a) 5-qubit representation of $f(x)$. (b) Fourier transform $\tilde{f}(k)$. (c) Zero-padded Fourier transform $\tilde{f}_{\text{zero-padded}}(p)$ with $M - N = 2^{10} - 2^5 = 992$ added zeros. (d) 10-qubit interpolated function $f_{\text{int}}(x)$.

tions. The QFT (2.18) and its inverse enable the creation of a quantum circuit to reconstruct the original function from the discretized state in momentum space. Section 4.3.4 includes the detailed implementation of the Fourier interpolation quantum circuit. The Nyquist-Shannon theorem indicates that the minimum number of qubits necessary for an accurate sampling is

$$n_{\min} \leqslant \log_2\left(\frac{L_p L_x}{2\pi}\right). \tag{2.24}$$

2.4 Dimensionality

The encoding techniques discussed in sections 2.1 to 2.3 can be generalized to work with multidimensional functions. Let us consider a d dimensional function $f(\mathbf{x}) = f(x_1, x_2, \dots, x_d)$ with a definition interval $x_i \in [a_i, b_i)$ of size $L_{x_i} = |b_i - a_i|$ for each dimension $i = (1, \dots, d)$. The encoding uses n_i qubits to represent each coordinate over 2^{n_i} points as

$$x_{i,s_i}^{(n_i)} = a_i + s_i \Delta x_i^{(n_i)}. \tag{2.25}$$

The integer $s_i \in \{0, 1, \ldots, 2^{n_i} - 1\}$ labels the grid coordinate $x_{i,s_i}^{(n_i)}$ of the i-th dimension. Then, a set of labels $\mathbf{s} = (s_1, s_2, \ldots, s_d)$ is the d-dimensional state associated to the coordinate $\mathbf{x_s}$. The normalized quantum state

$$\left| f^{(N)} \right\rangle = \frac{1}{\mathcal{N}_f^{1/2}} \sum_{\{s_i\}} f(\mathbf{x_s}) \left| \mathbf{s} \right\rangle, \tag{2.26}$$

represents a multidimensional function, where $\mathcal{N}_f$ is a normalization factor, and the number of qubits of the register is the sum of the registers for each coordinate $N = \sum_i n_i$.

Analogously, multidimensional operators in function space become transformations on the quantum register encoding each coordinate. For the concrete case of the QFT, the spectral method resorts to its multidimensional version $\hat{\boldsymbol{\mathcal{F}}}$, which transforms the state $|\mathbf{r}\rangle$ as

$$|\mathbf{r}\rangle \overset{\hat{\boldsymbol{\mathcal{F}}}}{\longmapsto} \frac{1}{\sqrt{2^N}} \sum_{\{s_i\}} e^{i2\pi r \sum_i s_i / 2^N} \left| \mathbf{s} \right\rangle, \tag{2.27}$$

extending the one-dimensional QFT to the d dimensions of $f(\mathbf{x_s})$ by applying the corresponding quantum circuits on the n_i qubits encoding each dimension x_i. Similar reasonings enable generalizing other operators—e.g., potentials $V(\mathbf{x})$, derivatives ∇,∇^2—and algorithms—e.g., integration, interpolation, etc.—to the multidimensional scenario. It is worth pointing out that when applying the differential operator, the QFT is only applied on the corresponding n_i qubits of the register, with a cost that scales linearly with the dimensionality for all tensorial addition or multiplication of operators. For the finite difference method, the displacement operators are defined for each register, $\Sigma_i^{\pm}$, acting only on the n_i qubits of the corresponding dimension.

2.5 Approximation errors

The interpolants in section 2.3 approximate the exact continuous function $f(x)$ up to an error that depends on the concrete technique. For the more straightforward finite difference interpolation, the number of elements of the Taylor expansion determines the polynomial scaling of the error with the spatial discretization. In the case of the Fourier interpolation, the number of nodes of the infinite Fourier series and the use of the discrete Fourier transform to approximate it dictate its accuracy. This section reviews some results of the finite difference method and Fourier analysis to evidence the errors of the function representation techniques that will be used in this thesis. These errors are independent of their application in classical, quantum, and quantum-inspired numerical analysis.

2.5.1 Finite difference errors

The finite difference method relies on the Taylor expansion of a function to provide different order approximations of derivatives. This method is suitable for continuous functions $f(x)$ that are continuously differentiable up to the order required by the finite difference scheme. For example, to approximate the second derivative of $f(x)$, it must be at least twice differentiable in the region of interest.

The simplest finite difference scheme arises from the definition of the derivative of a smooth function $f(x)$, with $x \in \mathbb{R}$,

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}. \tag{2.28}$$

The Taylor series

$$f(x+h) = \sum_{p=0}^{\infty} \frac{h^p}{p!} \left. \frac{d^p f(x)}{dx^p} \right| x^p, \tag{2.29}$$

provides the error of this approximation

$$f'(x) = \frac{f(x+h) - f(x)}{h} + O(h). \tag{2.30}$$

This finite difference scheme is known as forward finite difference, and its error is related to the supremum of the second derivative as [66]

$$\left| \frac{f(x+h) - f(x)}{h} - f'(x) \right| \leq Ch, \quad C = \sup_{y \in [x, x+h_0]} \frac{|f''(y)|}{2}, \tag{2.31}$$

with $h \leq h_0$.

The combination of two Taylor expansions

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2} f''(x) + O(h^3), \tag{2.32}$$

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2} f''(x) + O(h^3), \tag{2.33}$$

leads to the more accurate centered finite difference scheme

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + O(h^2), \tag{2.34}$$

whose continuous version is bounded by [66]

$$\left| \frac{f(x+h) - f(x-h)}{2h} - f'(x) \right| \leq Ch^2, \tag{2.35}$$
$$C = \sup_{y \in [x-h_0, x+h_0]} \frac{|f^{(3)}(y)|}{6}.$$

This expression improves the truncation error from linear in h to a quadratic order $O(h^2)$. This finite difference approximation of the derivative can be extended to an arbitrary order p by considering more terms of the Taylor expansion, assuming that the function is p times differentiable.

It is also possible to obtain an approximation of the second derivative $f''(x)$ by adding (2.32) and (2.33). This leads to the second-order centered finite difference

$$f''(x) = \frac{f(x+h) + f(x-h) - 2f(x)}{2h^2} + O(h^2). \tag{2.36}$$

Similarly to Eq. (2.31), Eq. (2.36) is bounded by [66]

$$\left| \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} - f'(x) \right| \leq Ch^2, \tag{2.37}$$
$$C = \sup_{y \in [x-h_0, x+h_0]} \frac{|f^{(4)}(y)|}{12}.$$

The most standard method is the second-order centered finite difference scheme for approximating the first and second derivatives. We will use these formulas to approximate differential operators in quantum-inspired scenarios in sections 5 and 6. In practical applications, the infinitesimal step h is substituted by the discretization step Δx, and the function $f(x_s)$ is defined over a discretized grid with $x_s = a + s\Delta x, s = 0, \dots, N-1$.

2.5.2 Fourier series errors and spectral content

The finite difference method's accuracy is limited to a polynomial scaling in the step size Δx, leading to an error of order p, $O(2^{-pn})$, decaying exponentially with the number of qubits n. Refs. [67, 68] show how, using Fourier analysis, it is possible to obtain an error that decays up to doubly exponentially in the number of qubits.

The Fourier series expands a function $f(x)$ as a sum of orthogonal square-integrable basis functions $L^2_{[0,2\pi]}$ [67], i.e., the Fourier modes $\phi_l(x) = \frac{1}{\sqrt{2\pi}} e^{ilx}$ with $k \in \mathbb{Z}$, as

$$f(x) = \sum_{l=-\infty}^{\infty} \hat{f}_l \phi_l. \tag{2.38}$$

The values $\hat{f}_l,\ l \in \mathbb{Z}$ are the Fourier coefficients

$$\hat{f}_l = \frac{1}{\sqrt{2\pi}} \int_0^{2\pi} f(x) e^{-ilx} dx. \tag{2.39}$$

In practical applications, the truncation of the infinite number of terms of the Fourier series leads to the truncated Fourier series

$$P_N f(x) = \sum_{l=-N/2}^{N/2} \hat{f}_l \phi_l. \tag{2.40}$$

Since this approximation removes terms from the original exact series, it introduces an error given by

$$\|f(x) - P_N f(x)\|^2 = \sum_{|l|=N/2+1}^{\infty} \left|\hat{f}_l\right|^2, \tag{2.41}$$

which depends on how fast the Fourier coefficients (2.39) decay to zero, which they always do according to the Riemann-Lebesgue lemma. For an m times continuously differentiable $f(x)$ in $x \in [0, 2\pi]$ with $m \geqslant 1$, and if $f^{(j)}(x)$ is periodic for all $j \leqslant m-2$, then $\hat{f}_l = O(1/l^m)$ [68]. This leads to a truncation error for $f \in H^m(0, 2\pi)$ of order [68]

$$\|f(x) - P_N f(x)\| = O(N^{-m}), \tag{2.42}$$

where H^m is the Sobolev space, i.e., the space of functions whose first $m-1$ derivatives are periodic. For an analytic function $f(x) \in C^\infty$ and periodic with all its derivatives on $[0, 2\pi]$, the decay of the lth Fourier coefficient is faster than any negative power of l [68]. More precisely, the error decays exponentially with N,

$$\|f(x) - P_N f(x)\| = O\left(e^{-rN}\right), \tag{2.43}$$

for $f(x)$ periodic with period 2π and analytic in a strip of radius $r > 0$ centered around the real axis $|\mathrm{Im} z < r|$ [68].

Gibbs phenomenon

Many functions show discontinuities in their definition or their derivatives, resulting in the appearance of the Gibbs phenomenon when using the Fourier series for their approximation. This causes oscillations around such discontinuities due to the necessity of arbitrarily large frequencies. Let us imagine a piecewise continuously differentiable periodic function $f(x)$, $x \in [0, 2\pi]$ with a jump discontinuity at $x = x_0$. Its truncated Fourier series [68] is given by

$$\begin{aligned} P_N f(x) &= \frac{1}{2\pi} \int_0^{2\pi} \left[\sum_{l=-N/2}^{N/2} e^{-il(x-y)} \right] f(y) dy \\ &= \frac{1}{2\pi} \int_0^{2\pi} D_N(x-y) f(y) dy, \end{aligned} \tag{2.44}$$

where $D_N(\xi)$ is the Dirichlet kernel

$$D_N(\xi) = 1 + 2 \sum_{k=1}^{N/2} \cos(k\xi) \,. \tag{2.45}$$

Approximating the Fourier series around the discontinuity as $N \to \infty$ leads to [68]

$$\begin{aligned} P_N f(x) \simeq & \frac{1}{2} \left[f(x_0^+) + f(x_0^-) \right] \\ & + \frac{1}{2\pi} \left[f(x_0^+) - f(x_0^-) \right] \int_0^{x-x_0} D_N(y) dy. \end{aligned} \tag{2.46}$$

The second term of the equation shows that the Dirichlet kernel modifies the jump of the function by a factor $(0.089489872236\ldots)$, which is around 9%. This also modifies the interpolated function since using the trapezoidal quadrature rule to relate Eq. (2.44) with it leads to

$$I_N f(x) = \frac{1}{N} \sum_{s=0}^{N-1} D_N(x - x_s) f(x_s). \tag{2.47}$$

Luckily, this effect is only present around discontinuities, and it averages out and decays algebraically as $N \to \infty$. Therefore, it does not limit the computations with $P_N f(x)$ or $I_N f(x)$ in the approximations of integrals and other observables.

2.5.3 Fourier interpolation errors

In numerical applications, the discrete Fourier series replaces the continuous version, whose expression is given by

$$\tilde{f}_l = \frac{1}{\sqrt{N}} \sum_{s=0}^{N-1} f(x_s) e^{-ilx_s}, \tag{2.48}$$

for $x_s = 2\pi s/N, s = 0, \ldots, N-1, x \in [0, 2\pi)$ and $-N/2 \leqslant l \leqslant N/2 - 1$. This leads to the Fourier interpolant $I_N f(x)$ of degree $N/2$

$$I_N f(x) := \sqrt{\frac{2\pi}{N}} \sum_{l=-N/2}^{N/2-1} \tilde{f}_l \phi_l(x) = \frac{1}{\sqrt{N}} \sum_{l=-N/2}^{N/2-1} \tilde{f}_l e^{ilx}, \tag{2.49}$$

that coincides with the original function in the original lattice values, i.e., $I_N f(x_s) = f(x_s)$. The difference of the exact $\hat{f}_l$ and discrete $\tilde{f}_l$ Fourier coefficients is quantified by the aliasing term R_N,

$$R_N := \tilde{f}_l - \hat{f}_l = \sum_{m=-\infty, m\neq 0}^{\infty} \hat{f}_{l\pm mN}. \tag{2.50}$$

Since the Fourier coefficients are orthogonal, the Fourier series truncation and the aliasing onto the lattice commute. Then, the difference between the original function and the interpolated one is

$$\begin{aligned} &\|f(x) - I_N f(x)\|^2_{H^m(0,2\pi)} \\ &\qquad \leq \|f(x) - P_N f(x)\|^2_{H^m(0,2\pi)} + \|R_N f(x)\|^2_{H^m(0,2\pi)}, \end{aligned} \tag{2.51}$$

where $H^m(0, 2\pi)$ is the Sobolev space of functions $L^2(0, 2\pi)$ such that all the distributional derivatives of u of order up to m can be represented by functions in $L^2(0, 2\pi)$ [68]. This shows that the interpolation error $\|f(x) - I_N f(x)\|$ behaves asymptotically like the truncation error [69, 68], and then, for analytical functions, the interpolation error decreases exponentially with the number of points of the lattice N and doubly exponentially with the number of qubits n, since $N = 2^n$.

2.5.4 Fourier differentiation errors

Fourier differentiation relies on Fourier interpolation since it uses a discretized version of the derivative obtained from differentiating the interpolated function $(D_N f)(x) := \partial_x I_N f(x)$. However, aliasing prevents interpolation and differentiation from commuting [67], as opposed to the truncation case. Thus, the resulting errors may differ. The errors associated to this discretized differentiation are given by

$$\|f'(x) - \mathcal{D}_N f(x)\| = O\left(N^{1-m}\right). \tag{2.52}$$

for a differentiable function $f \in H^m(0, 2\pi)$ from a Sobolev space $H^m(0, 2\pi)$ that supports $m \geqslant 1$ derivatives [68]. This error decays exponentially with N [70] for an analytic function.

In practical applications, physically motivated PDEs have solutions that are analytical functions and, hence, arbitrarily differentiable. Thus, the quantum Fourier numerical analysis techniques constitute a valuable framework for the highly efficient representation of many functions of interest. The error may decay up to exponentially with the points of the lattice N and doubly exponential with the number of qubits n, i.e., $O(e^{-rN}) \sim O(e^{-r2^n})$, with some problem-dependent constant r. In addition, decreasing the Gibbs oscillation up to an arbitrarily small value is possible by enlarging the interval size L until the value of $f(x)$ is negligible, for which the function and higher derivatives are close to periodicity.

Chapter 3

Tensor networks

Tensor networks represent quantum states as a net of interconnected tensors. This representation originated in the field of quantum many-body physics, as it can potentially reduce memory requirements exponentially, addressing the *curse of dimensionality*. The memory compression of tensor networks has motivated the extension of this formalism to other fields, such as machine learning or numerical analysis.

This chapter introduces tensor network foundations, explicitly focusing on their application in developing quantum-inspired algorithms. The discussion centers on one-dimensional tensor networks, namely matrix product states (MPS) and operators (MPOs), which set the basis for one of the central tools of this thesis, the MPS-MPO finite precision algebra introduced in section 3.5. This algebra combines basic MPS-MPO operations with truncation algorithms, providing a suitable framework to create tensor network quantum-inspired algorithms. By extending the quantum register encoding of functions and operators in chapter 2 to an MPS-MPO representation, this algebra allows us to tackle numerical analysis problems. Examples include the solution of Hamiltonian PDEs described in Chapter 5 and the evolution problem in Chapter 6. The practical implementation of this MPS-based algebra and the algorithms for numerical analysis uses the SeeMPS [51] Python library described in chapter 7.

3.1 Motivation

The Hilbert space of a composite quantum system grows exponentially with its number of elements, imposing prohibitive resources for representing moderate-size quantum models, i.e., a d-level N-particle system needs d^N coefficients to construct its quantum state exactly. For example, a two-level system with 80 particles would need $2^{80} \approx 10^{24}$ elements to write the quantum state— a quantity comparable to the number of stars in the Universe. Classical methods to simulate such systems often struggle with this exponential scaling, facing memory limitations for two-level systems with more than 30 particles [71], and even fewer for more complex scenarios. This issue is particularly important in areas of physics that deal with systems with large numbers of elements, such as quantum many-body systems in condensed matter physics [72] and quantum computing.

Tensor networks [37, 73] provide an alternative formalism for working with these systems, potentially enabling an exponential compression in the number of parameters of a quantum state. This reduction in scaling applies to states that verify the *area law* for the entanglement entropy—the entanglement entropy of such region is proportional to the size of the region's boundary rather than its volume. Consequently, these states are restricted to a region of the Hilbert space and hence constrained by locality [37], such as states with nearest or next-to-nearest interactions [37, 74, 75, 76, 77, 78]. There are many Hamiltonians in Nature of this type [37, 79], so in practice, many important problems do not need to access the entire Hilbert space. These states can be targeted by tensor networks, taking advantage of this locality constraint to reduce the information needed for their representation and enabling the study of quantum many-body problems based on the entanglement properties of the quantum states.

The concept of tensor network dates back to 1941 when Kramers and Wannier [80, 81] introduced transfer matrices to the Ising problem. Significant developments in the field [82, 83, 84, 85, 86] followed this work, which peaked with the creation of the density matrix renormalization group (DMRG) [87, 88]. The original proposal was reformulated as a variational algorithm to find the lowest eigenstate of some quantum-many body operator [89, 90, 91], unleashing the true potential of DMRG via the use of matrix product states (MPS). This algorithm performs a local optimization of the tensors in an MPS, sweeping various times over the whole system until the expected energy converges to the lowest value. This reformulation of DMRG opened the venue for many applications in fields such as condensed matter physics [92, 93] and computational chemistry [94, 95, 96]. Nowadays, DMRG is one of the most popular MPS algorithms. It has been extended to tackle different problems, such as finite-temperature [97, 98] periodic boundary conditions MPS [91], infinite MPS [99] or higher dimensional tensor networks [100, 101, 102]. Another important landmark was the time-evolving block decimation (TEBD) algorithm [103, 104, 97] to address the time evolution problem. TEBD uses the Suzuki-Trotter formula to approximate the exponential evolution by local operators. The time-dependent variational principle (TDVP) [105, 106, 107] leverages the tangent space to simulate the real- and imaginary-time dynamics of the system, and other proposals combine DMRG-like methods with Taylor, Padé, and Arnoldi approximations of the evolution operator [108, 109]. The evolution problem can be studied both in imaginary and real-time and more DMRG versions have been developed with this purpose [110, 111, 97]. This development of the area has led to many applications in the context of quantum many-body physics: the approximation of the low-energy properties of quantum Hamiltonians [103, 104, 97, 87, 88], the simulation of spin, bosonic and fermionic systems even in multiple dimensions [112, 113, 114, 115, 101, 116, 117, 118, 119, 120, 121, 122], and the study of quantum phases of matter [123, 124, 125, 124, 126], among others.

A common characteristic of previous applications is that the system's entanglement grows slowly with the problem size or simulation time, enabling an efficient tensor network representation. However, tensor networks have been successfully applied to problems with unknown entanglement structures. Tensor networks, whose one-dimensional representation was rediscovered as tensor trains [127] and quantized tensor trains [128] in the area of mathematics, have applications in diverse fields, such as machine learning [129, 130, 131, 132, 133, 134, 135, 136] and numerical analysis [109, 44, 137, 38, 49], demonstrating their potential to yield effective heuristic results. The success of these applications illustrates the capability of tensor networks to address the challenges posed by the exponential scaling beyond quantum many-body physics, motivating the development of quantum-inspired methods. This thesis explores this domain, introducing a flexible and powerful framework for the creation of quantum-inspired algorithms (Section 3.5) to solve static and time-dependent PDEs in chapters 5 and 6, respectively.

3.2 Tensor Networks

Let us define a tensor as a multidimensional array of complex numbers with respect to a given basis. The tensor indices represent its dimensions and allow access to its components. The total number of indices is the rank of the tensor. Many everyday mathematical objects are tensors of a certain rank; for example, a scalar is a rank-0 tensor x, a vector is a rank-1 tensor v_α, and a matrix is a rank-2 tensor $A_{\alpha\beta}$.

Tensor contraction, i.e., summing over all values of the repeated tensor indices, is the basis of tensor network operations. A simple instance of tensor contraction is matrix multiplication. Let $A_{\alpha\beta}$, $B_{\gamma\delta}$ be two matrices with dimensions, $N_A \times M$, $M \times N_B$, respectively. Then, the matrix product can be written as

$$C_{\alpha\delta} = \sum_{\beta=1}^{M} A_{\alpha\beta} B_{\beta\delta}, \tag{3.1}$$

where $\beta = 1, \ldots, M$. The number of values M of the contracted index determines the cost of the contractions. Similarly, contractions support more operations like computing the trace and the expectation value. Other significant tensor operations include the permutation of indices and reshaping.

As the number of tensors increases, the mathematical representation of tensor contraction as a sum over indices becomes cumbersome. The tensor network community uses a diagrammatic notation to simplify this representation, enabling a quick and intuitive understanding of these operations. This notation is a modification of the Penrose graphical notation or tensor diagram notation [138] and is nowadays widely employed in different fields besides tensor networks, such as category theory, computing science, and quantum algorithms. Figure 3.1(a) shows a variety of tensors in this diagrammatic representation, in which tensors are shapes, with *legs* representing the tensor indices. In this notation, matrix multiplication (3.1) is depicted as in Figure 3.1(b), where connections among tensors represent contractions of the given indices. The Kronecker delta δ_{ij}, which represents a contraction of a tensor index with the identity matrix, completes this diagrammatic notation. It does not modify the tensor, and it is therefore represented as a straight line that goes from the index i to j.

The wavefunction of a quantum many-body system of N d-level particles is [139, 37]

$$|\psi\rangle = \sum_{i_1,\ldots,i_N} C_{i_1,\ldots,i_N} |i_1\rangle \otimes \cdots \otimes |i_N\rangle, \tag{3.2}$$

where $i_k = 0, ..., d-1$, $k = 1, \ldots, N$. The wave function's coefficients $C_{i_1,\ldots,i_N}$ can be understood as a rank-N tensor. By decomposing this tensor in a network of smaller contracted tensors (Figure 3.2)—a tensor network—it is possible to reduce the exponential scaling of the number of parameters with the number of particles to a polynomial one, $O(\text{poly}(N)\text{poly}(\chi))$ [37], where χ is the bond dimension, i.e., the rank of

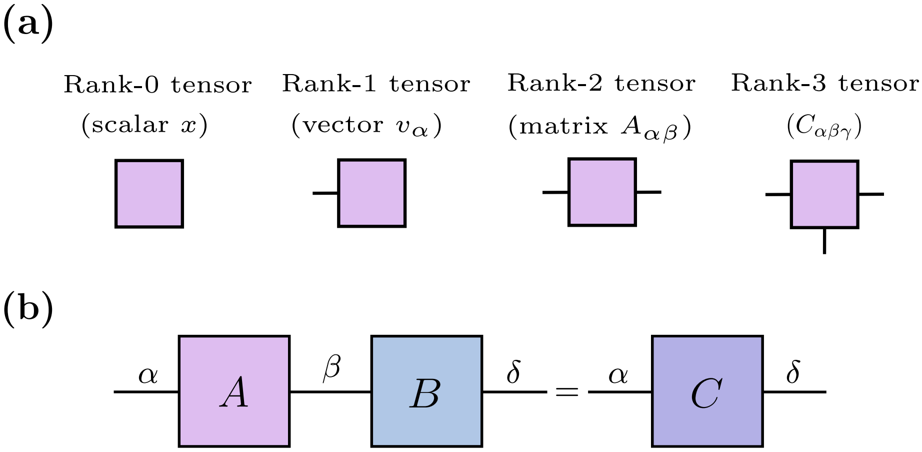


Figure 3.1: Diagrammatic representation of tensors. (a) Diagrammatic representation of rank-0, -1, -2, and -3 tensors, where each tensor is a shape with *legs* representing the indices. (b) Diagrammatic representation of matrix multiplication (3.1).

the contracted indices of the tensors. These indices are called virtual or bond indices, and they represent the entanglement among the elements of the system and, hence, the quantum correlations. The bond dimension, the number of tensors, and the network structure bound the entanglement entropy of the quantum state following an area law [37, 140] and targeting the corner of relevant states of the Hilbert space, hence determining the possible applications of the tensor network. Figure 3.3 illustrates different tensor networks. Matrix product states (MPS) reproduce a one-dimensional lattice model, and projected entangled pair states (PEPS) [141] represent two-dimensional lattices. More complex applications resort to tree tensor networks (TTN) or the multi-scale entanglement renormalization ansatz (MERA). The tensor network diagrammatic representation is helpful in practical applications since it facilitates working with tensor contractions involving multiple indices for networks with numerous tensors and complex connections.

An interesting property of tensor networks is their gauge freedom [72], meaning that different tensors can give rise to the same state for a given network structure. Thus, the quantum state is invariant under the introduction of a pair of matrices $XX^{-1} = \mathbb{I}$ between two contracting indices of the tensor network (Figure 3.2), enabling the search for the optimal size tensors that minimize resources.

Besides choosing the most appropriate gauge, the order of the con-

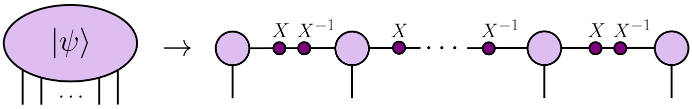


Figure 3.2: General quantum state (3.2) as an MPS. The number of indices of the original tensor is the rank of the tensor, and each index corresponds to the physical index of a tensor of the tensor network. The matrix X accounts for the gauge freedom of the tensor network, meaning that the contraction of different tensors can represent the same quantum state.

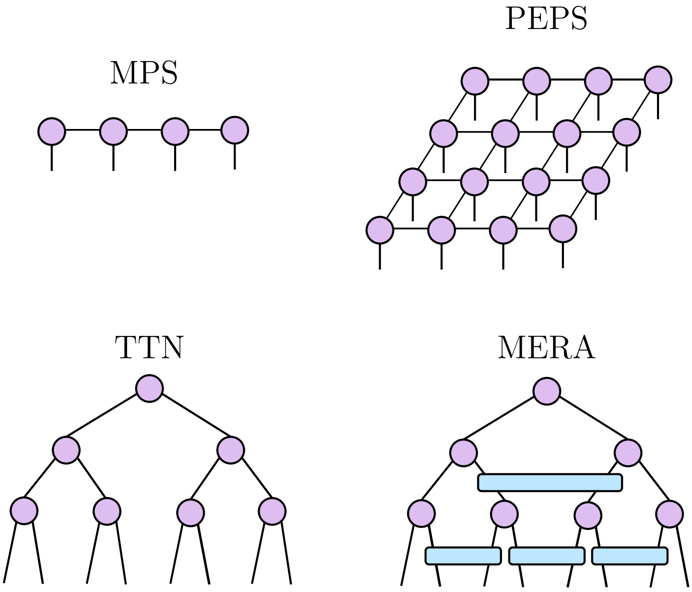


Figure 3.3: Principal types of tensor networks. The figure depicts matrix product states (MPS), projected entangled pair states (PEPS), tree tensor network (TTN), and multi-scale entanglement renormalization ansatz (MERA).

tractions plays a determinant role in the performance of tensor network algorithms since it affects the number of operations and, consequently, its cost. For example, Figure 3.4 shows two different ways of contracting a three-tensor network to obtain the same result. Order (a) involves two $O(d\chi^3)$ contractions, while in order (b), the first contraction is $O(d\chi^4)$ followed by a $O(\chi^4)$ contraction. In most cases, order (a) is more efficient, as it involves lower-order contractions as normally $\chi \gg d$. Finding

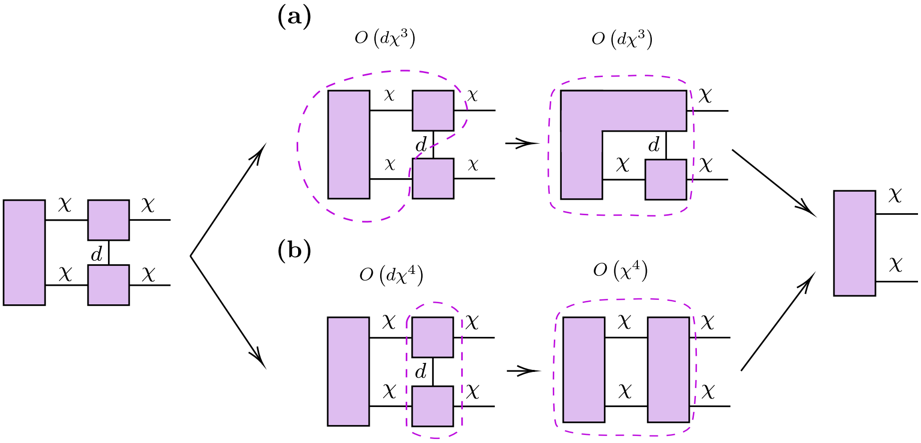


Figure 3.4: Contraction of a three-tensor network. Order (a) leads to lower-order contractions than order (b).

the optimum contraction path is key to minimizing the cost of tensor network algorithms, especially as the number of tensors and indices increases. Greedy search algorithms are often employed to approximate such paths for complex contractions. However, many practical implementations usually involve the subsequent application of the same contraction patterns, allowing the exact determination of the optimum path in advance and thus avoiding the additional overhead of search algorithms.

3.3 Matrix product states

Matrix product states (MPS) are one of the most extended tensor networks due to their simplicity and versatility. They are one-dimensional tensor networks formed by rank-3 tensors representing the elements of the system, as depicted in Figure 3.5. In quantum applications, the free indices $\{i_k\}_{k=1,\ldots,N}$ are called the physical indices and correspond to the physical degrees of freedom of the Hilbert space of a one-dimensional d-level system. The bond indices α_k and β_k bound the entanglement entropy of the quantum system as $S \leq \log \chi$, where χ is the maximum bond dimension. Then, for $\chi = 1$, there are no correlations among the connected tensors and $S = 0$, representing a product state. The entropy saturates to $S = \log \chi$ for the maximally entangled state. The general dependence of MPS entropy is $S = O(\log \chi)$, meaning that MPS verify a one-dimensional area law independent of the number of sites of the

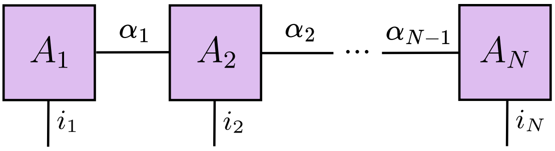


Figure 3.5: N-site MPS with open boundary conditions. The sites in the extremes of the MPS—sites one and N—are represented by just two indices, as the extreme bond indices have bond dimension 1.

tensor network [37].

To represent a general quantum state (3.2) as an MPS, let us decompose the tensor $C_{i_1,...,i_N}$ into N rank-3 tensors, following the procedure in Ref. [37]. The first step is to apply the Schmidt decomposition to the subspaces of the first index and the remaining $N-1$ indices, which allows us to rewrite the total system as a bipartite system

$$|\psi\rangle = \sum_{\alpha_1=1}^{\min(d,\chi)} \lambda_{\alpha_1}^{[1]} |\tau_{\alpha_1}^{[1]}\rangle \otimes |\tau_{\alpha_1}^{[2...N]}\rangle\,, \tag{3.3}$$

where the Schmidt coefficients are $\lambda_{\alpha_1}^{[1]}$, and $|\tau_{\alpha_1}^{[1]}\rangle\,, |\tau_{\alpha_1}^{[2...N]}\rangle$ are the corresponding left and right Schmidt vectors. Then, the left Schmidt vector is rewritten in the local basis $|i_1\rangle$

$$|\psi\rangle = \sum_{i_1=1}^{d} \sum_{\alpha_1=1}^{\min(d,\chi)} \Gamma_{\alpha_1}^{[1]i_1} \lambda_{\alpha_1}^{[1]} |i_1\rangle \otimes |\tau_{\alpha_1}^{[2...N]}\rangle\,, \tag{3.4}$$

where $\Gamma_{\alpha_1}^{[1]i_1}$ corresponds to the change of basis $|\tau_{\alpha_1}^{[1]}\rangle = \sum_{i_1} \Gamma_{\alpha_1}^{[1]i_1} |i_1\rangle$. The Schmidt decomposition iteratively acts on the subsequent subsystems until each site's tensor is obtained in the local basis for each particle (Figure 3.6). This leads to the quantum state representation

$$|\psi\rangle = \sum_{\{i\}} \sum_{\{\alpha\}} (\Gamma_{\alpha_1}^{[1]i_1} \lambda_{\alpha_1}^{[1]} \Gamma_{\alpha_1\alpha_2}^{[2]i_2} \lambda_{\alpha_2}^{[2]} ... \lambda_{\alpha_{N-1}}^{[N-1]} \Gamma_{\alpha_{N-1}}^{[N]i_N}) |i_1\rangle \otimes |i_2\rangle \otimes ... \otimes |i_N\rangle\,, \tag{3.5}$$

where the sum over each index in $\{i\}$ and $\{\alpha\}$ runs up to their respective allowed values. Grouping the tensors as $A_{\alpha_{k-1},\alpha_k}^{i_k} = \Gamma_{\alpha_{k-1}\alpha_k}^{[k]i_k} \lambda_{\alpha_k}^{[k]}$ results in an MPS with open boundary conditions representing the general quantum state in Eq. (3.2),

$$|\psi\rangle = \sum_{\{i\}} \sum_{\{\alpha\}} (A_{\alpha_1}^{i_1} A_{\alpha_1,\alpha_2}^{i_2} \dots A_{\alpha_{N-1}}^{i_N}) |i_1\rangle \otimes |i_2\rangle \otimes ... \otimes |i_N\rangle\,. \tag{3.6}$$

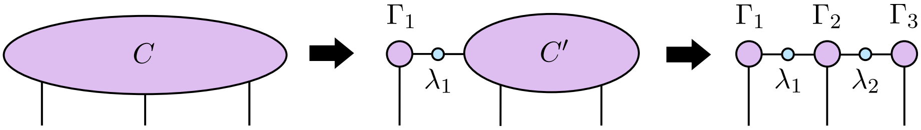


Figure 3.6: Rewriting a quantum state as an MPS. Iterative application of the SVD to decompose a tensor as an MPS for a 3-site example.

The decomposition of the MPS in terms of the Schmidt coefficients facilitates obtaining the entanglement entropy of any bipartition. For a density matrix ρ_μ with eigenvalues $\{p_\mu\}$, the associated entropy is $S(\{p_\mu\}) = -\sum_\mu p_\mu \log(p_\mu)$. The eigenvalues of the density matrices of the subsystems are defined as $p_\mu = |\lambda_\mu|^2$, thus the decay of the Schmidt coefficients determine the entanglement entropy of the system.

One possible Schmidt decomposition can be obtained through the singular value decomposition (SVD) [142]. Given an $m \times n$ matrix C, the SVD factorizes it as

$$C = USV^\dagger. \tag{3.7}$$

In its compact version, U and $V^\dagger$ are, respectively, an $m \times \max(m, n)$ complex unitary matrix whose columns are the eigenvectors of $CC^\dagger$ (left-singular vectors) and a $\max(m, n) \times n$ complex unitary matrix whose columns are the eigenvectors of $C^\dagger C$ (right-singular vectors). S is a rectangular diagonal matrix with positive real numbers on the diagonal, which correspond to the singular values of C.

The first step to obtaining the Schmidt decomposition via the SVD is to reshape the rank-N tensor $C_{i_1,\ldots,i_N}$ as a matrix $C = C_{i_1,j}$, $j = \{i_2 \ldots, i_N\}$, which represents the coefficients of the bipartite system

$$|\psi\rangle_{AB} = \sum_{i_1,j} C_{i,j} \, |i_1\rangle_A \, |j\rangle_B \,, \tag{3.8}$$

where $\{|i_1\rangle_A\}$, $i_1 = 0, \ldots, d_A - 1$, and $\{|j\rangle_B\}$, $j = 0, \ldots, d_B - 1$, are the basis of the subsystems with Hilbert spaces $\mathcal{H}_A$ and $\mathcal{H}_B$, respectively. Then, applying the SVD to the state in Eq. (3.8) and reorganizing terms leads to

$$|\psi\rangle_{AB} = \sum_{\alpha=1}^{\chi} \lambda_\alpha \sum_{i_1}^{d_A} U_{i_1,\alpha} \, |i_1\rangle_A \sum_{j}^{d_B} V^\dagger_{\alpha,j} \, |j\rangle_B = \sum_{\alpha=1}^{\chi} \lambda_\alpha \, |\alpha\rangle_A \, |\alpha\rangle_B \,, \tag{3.9}$$

where the singular values of C, $S = \lambda_\alpha \mathbb{I}$, correspond to the Schmidt coefficients, and the basis elements of each space are

$$|\alpha\rangle_A = \sum_{i_1}^{d_A} U_{i,\alpha} |i_1\rangle_A \,, \quad |\alpha\rangle_B = \sum_{j}^{d_B} V^\dagger_{\alpha,j} |j\rangle_B \,. \tag{3.10}$$

This first SVD of C produces the first tensor $A_1 = U^{i_1}_{\alpha_1}$ of the MPS, which is a left partial isometry, i.e., $(A^{i_1}_{1_{m,l}})^\dagger A^{i_1}_{1_{m,j}} = \delta_{j,l}$, as depicted in Figure 3.7(a). The iterative application of the SVD on the remaining system leads to the complete tensor representation in the so-called left canonical form, in which all tensors except A_N, i.e., the last tensor, are left partial isometries. In the canonical form [79], each bond index corresponds to the labeling of Schmidt vectors in the Schmidt decomposition of the quantum state across that index [37], storing all the information of the MPS in a single tensor. The canonical form simplifies some MPS operations, like the calculation of the norm and the expectation value of local operators (Figure 3.7(c)-(d)), which involve contracting the MPS with its conjugate transpose. In this case, all contractions result in the identity, except for the Nth tensor, reducing the computation to the contraction of this tensor and its conjugate transpose for the norm, and similarly, by applying the local operator $\hat{O}_N$ in the middle of the tensors, for the expectation value of a local operator. The canonical form largely decreases the order of the contraction, from $O(N\chi^3 d)$ to $O(\chi^2 d)$ for the norm and similarly for the expectation value. In addition, the canonical form is also more stable and less prone to numerical errors arising from the contractions.

The canonical form is not unique, and the tensor network gauge freedom enables us to obtain the most appropriate one for each operation. For instance, it is possible to obtain the right canonical form, in which all tensors except the first one are right partial isometries (Figure 3.7(b)), or mixed forms with left and right partial isometries with respect to a concrete center of the MPS. Given a general MPS, it is possible to obtain the canonical form with respect to a desired site by iteratively performing the SVD on two contracted sites to obtain the needed left and right partial isometries, as depicted in Figure 3.8.

The Schmidt decomposition provides a means to approximate a quantum system by removing the Schmidt coefficients with the least weight in the states, resulting in a new quantum state with an error arising from the truncation of such coefficients. In practice, only the non-zero coefficients are considered, or a set $\lambda_\alpha > \lambda_{\text{min}}$ with a minimum value λ_{min} as a threshold. This reduces the bond dimension of the tensors to the number of elements retained, leading to a number of order $O(Nd\chi^2)$ parameters.

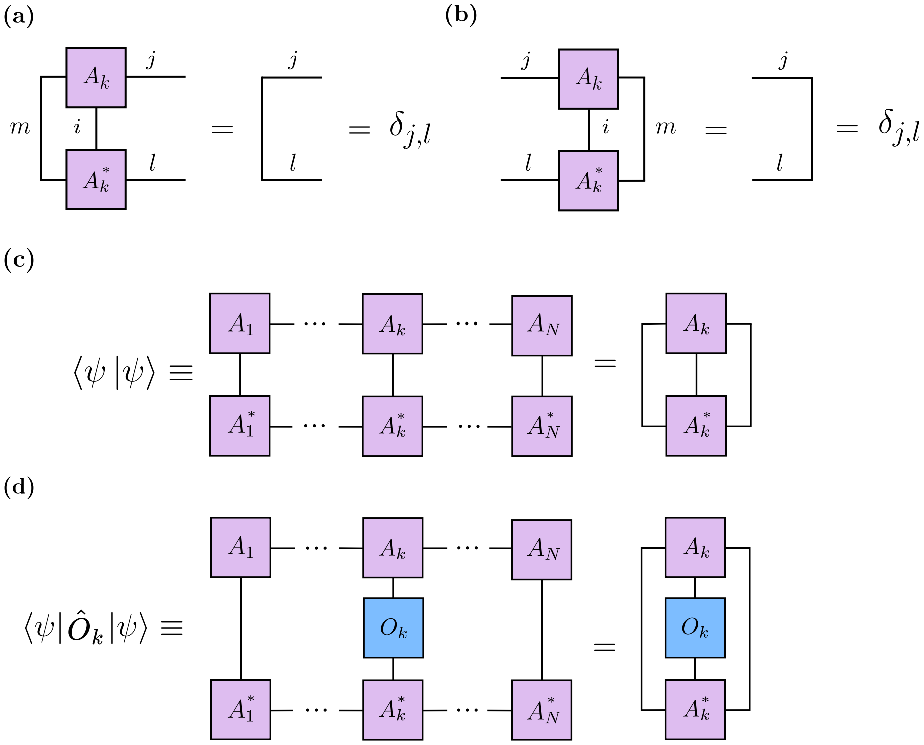


Figure 3.7: Canonical form for MPS. (a) Left partial isometry. (b) Right partial isometry. (c) Scalar product of an MPS in canonical form. (d) Expectation value of local operator $\hat{O}_k$ for an MPS in canonical form with respect to site k.

This scaling achieves exponential memory compression compared to the standard vector representation if the bond dimension χ does not grow exponentially with the system size. The SVD is thus a useful tool for MPS truncation, allowing control of the tensors' size. In practice, highly accurate results can be obtained with relatively low bond dimensions, as demonstrated in the solution of static and time-dependent PDEs in chapters 5 and 6.

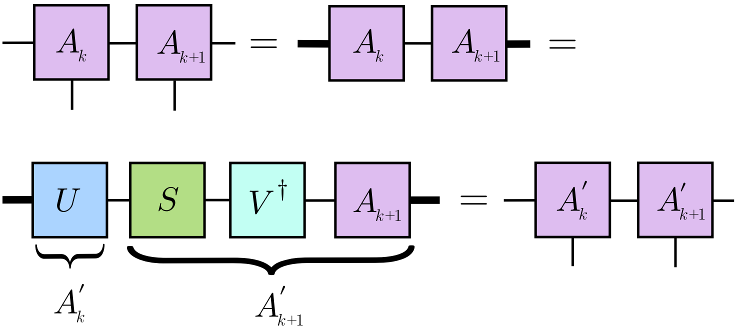


Figure 3.8: SVD for MPS canonical form. The SVD was applied on two MPS sites to obtain the tensors to construct the canonical form.

3.4 Matrix product operators

The mathematical representation of a general operator $\hat{O}$ in the basis $\{|i_1, \ldots, i_N\rangle\}$ is

$$\hat{O} = \sum_{i_k, i'_k} O^{i_1,\ldots,i_N}_{i'_1,\ldots,i'_N} |i'_1, \ldots, i'_N\rangle\langle i_1, \ldots, i_N|, \tag{3.11}$$

Similarly to a quantum state (3.2), it is possible to rewrite a quantum operator as a net of one-dimensional tensors, known as matrix product operator (MPO). The expression of an MPO with open boundary conditions is [143]

$$\hat{O} = \sum_{i_k, i'_k} \sum_{\beta_k} W^{i_1,i'_1}_{\beta_1}[1] W^{i_2,i'_2}_{\beta_1,\beta_2}[2] \ldots W^{i_N,i'_N}_{\beta_{N-1}}[N] |i'_1, \ldots, i'_N\rangle\langle i_1, \ldots, i_N|. \tag{3.12}$$

where we have used that $O^{i_1,\ldots,i_N}_{i'_1,\ldots,i'_N} = \sum_{\beta_k} W^{i_1,i'_1}_{\beta_1}[1] W^{i_2,i'_2}_{\beta_1,\beta_2}[2] \ldots W^{i_N,i'_N}_{\beta_N}[N]$. MPOs are formed by rank-4 tensors, each with two bond and two physical indices.

The MPS-MPO contraction is equivalent to applying a quantum operator on a quantum state. The MPS and MPO are contracted through the physical indices, resulting in a new MPS where the bond dimension is the product of the bond dimensions of the original MPS and the MPO. Figure 3.9(d) shows this operation, which results in a new MPS with bond dimension χ^2. This quadratic growth of the bond dimension leads to an exponential scaling as successive operations accumulate, making it necessary to introduce bond dimension truncation algorithms to avoid excessive computational costs.

3.5 Finite precision algebra

The combination of MPS and MPO with a set of basic operations constitutes an algebra that operates similarly to matrix-vector operations. This MPS-MPO algebra—which we originally introduced in Ref. [49]—is one of the primary bases of this thesis since it enables the extension of algorithms initially designed for a vector algebra to the MPS-MPO representation. Therefore, this algebra constitutes a fundamental tool for creating quantum-inspired algorithms in chapters 5 and 6.

Figure 3.9 depicts the basic operations of the MPS-MPO algebra. It consists of three linear operations: (i) the scalar product $\langle\psi_A|\psi_B\rangle$ (Figure 3.9(a)), (ii) the matrix-vector multiplication as MPO-MPS contraction (Figure 3.9(c)), and (iii) the scalar multiplication, performed by multiplying the MPS/MPO tensors by the corresponding scalar weights. These operations combine to create new operations, like the computation of expectation values in Figure 3.9(b). In addition to these standard linear operations, we include a nonlinear one, the elementwise vector multiplication, i.e., the multiplication of two MPS (Figure 3.9(d)). This operation is commonly performed in numerical analysis, in which potential operators are diagonal and, hence, can be applied to a function as an elementwise multiplication of their vector elements. Even though the canonical form is not necessary to perform these operations, its use is advisable to reduce the cost of some of them, like the computation of the norm and the expectation product of quasilocal operators, and to increase the numerical stability.

The described algebra is equivalent to the vector algebra since the operations are exact, i.e., only limited by finite computer precision. However, applying an MPO onto an MPS, $\hat{O}\,|\psi\rangle$, or computing the expectation value $\langle\psi|\hat{O}\,|\psi\rangle$, are tasks that, when implemented naively, lead to a polynomial increase in the tensors sizes. The consecutive application of exact MPO and MPS operations scales exponentially with χ, rendering them intractable in practice due to the order of operations $O(\chi^4)$ and the memory requirements. Then, even though it is possible to perform numerically exact operations within the tensor network formalism, it is necessary to truncate their bond dimension to keep the size of the problem manageable for current computers. Conventional variational methods can reduce the bond dimension's growth. These methods define the problems of approximating a state or a linear combination of states as optimization tasks, searching for the best approximating MPS with a given bond dimension or a maximum error.

Let us introduce a variational truncation algorithm [144] to approxi-

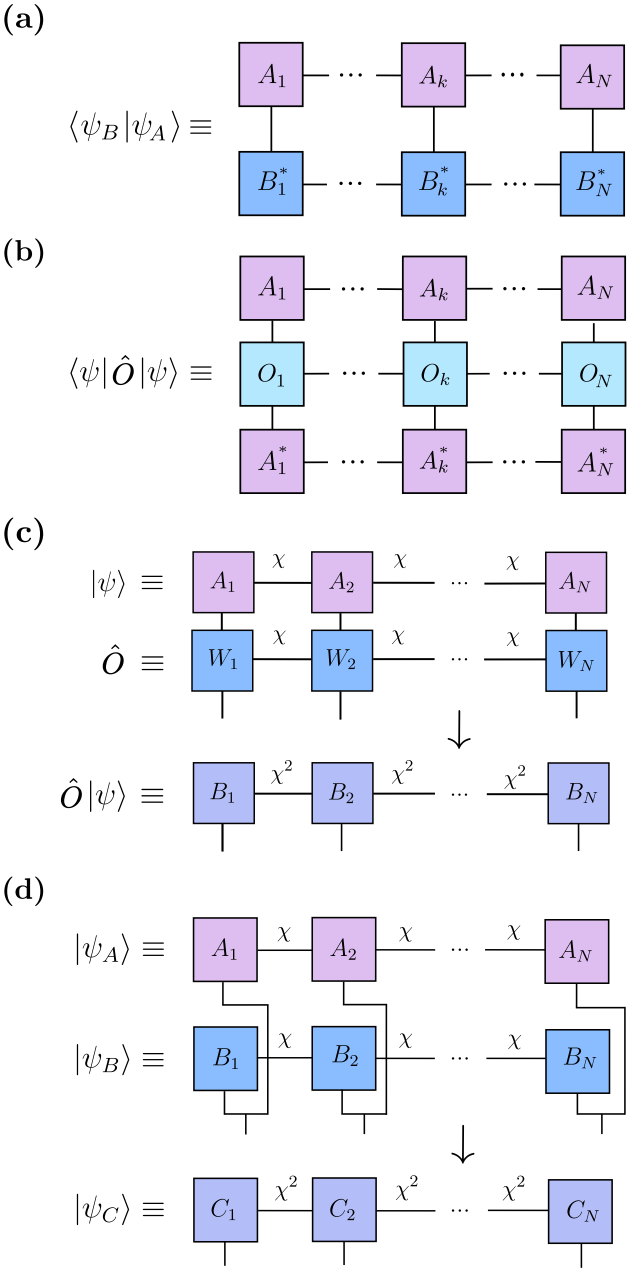
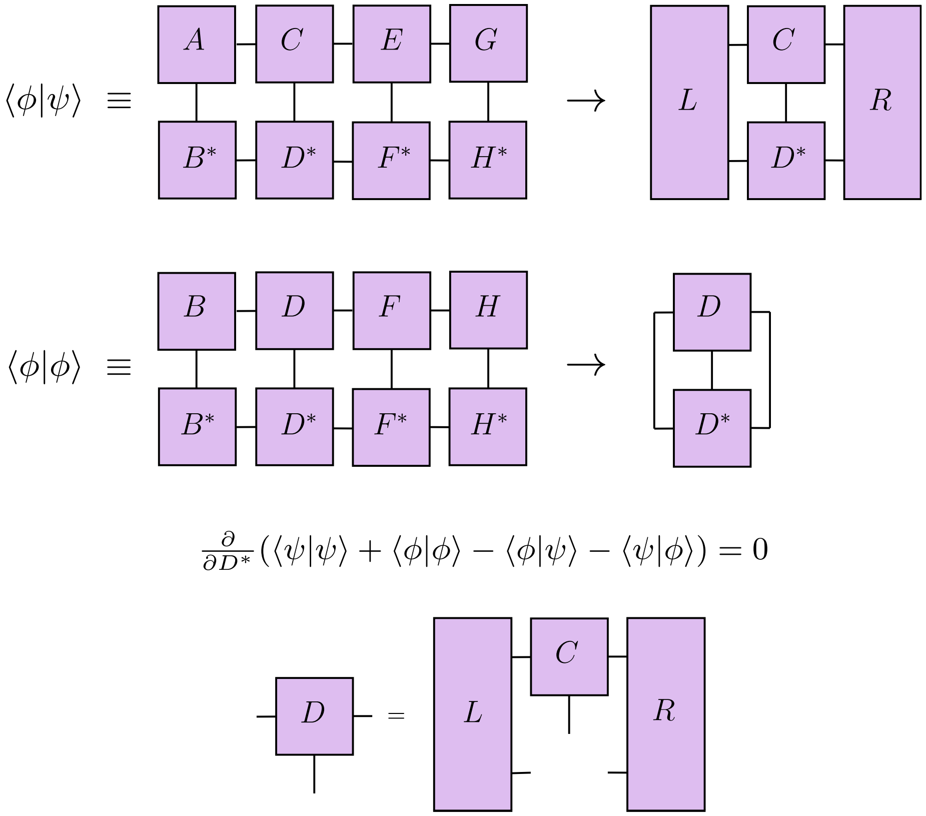


Figure 3.9: MPS-MPO finite precision algebra operations. (a) Scalar product of two MPS. (b) Expectation value of an MPO for an MPS state. (c) Application of a quantum operator $\hat{O}$ on a quantum state $|\psi\rangle$, $\hat{O}|\psi\rangle$, as an MPO-MPS contraction. (d) Contraction of two MPS resulting in an MPS.



$$\frac{\partial}{\partial D^*}\left(\langle\psi|\psi\rangle + \langle\phi|\phi\rangle - \langle\phi|\psi\rangle - \langle\psi|\phi\rangle\right) = 0$$

Figure 3.10: MPS one-site variational truncation algorithm. Diagrammatic representation of one step of the algorithm to minimize Eq. (3.13) with respect to a single tensor.

mate an MPS quantum state $|\psi\rangle$ with bond dimension χ_ψ by projecting it in the subspace of MPS with bond dimension χ_ϕ, MPS_{χ_ϕ}, such that $\chi_\phi < \chi_\psi$. This algorithm targets the minimization of the distance

$$\text{argmin}_{\phi\in\text{MPS}_{\chi_\phi}} d(\psi,\phi) = \|\psi-\phi\|^2 = \langle\psi|\psi\rangle + \langle\phi|\phi\rangle - 2\Re(\langle\psi|\phi\rangle). \quad (3.13)$$

This expression constitutes a bilinear function with respect to any tensor in $|\phi\rangle$, resulting in a functional that can be efficiently optimized via an iterative algorithm.

Figure 3.10 depicts one step of the one-site MPS truncation algorithm. Let us start by writing the MPS in canonical form, which is maintained through the algorithm's iterations to enhance its performance. Then, in each step, the algorithm performs a local optimization of the MPS, variationally optimizing the tensor parameters while keeping the rest fixed. The new optimized site D is the one that minimizes the distance in Eq. (3.13)

$$\frac{\partial}{\partial D^i_{\alpha\beta}} d(\psi, \phi) = \frac{\partial}{\partial (D^*)^i_{\alpha\beta}} d(\psi, \phi) = 0. \tag{3.14}$$

To obtain D, it is necessary to compute the partial derivatives

$$\frac{\partial}{\partial (D^*)^i_{\alpha\beta}} \langle\psi|\phi\rangle = U^i_{\alpha\beta}, \quad \frac{\partial}{\partial (D^*)^i_{\alpha\beta}} \langle\phi|\phi\rangle = D^i_{\alpha\beta}, \tag{3.15}$$

where U is constructed from the contraction of the left environment L, the right environment R and C. This results in the new expression of the minimized tensor $D^i_{\alpha\beta} = U^i_{\alpha\beta}$. The canonical form plays a key role in this algorithm, simplifying the computation of the norm of the new state and reducing numerical instabilities. This optimization is repeated on all sites until a sweep of the entire MPS is completed. Subsequently, the algorithm assesses the error in the MPS distance (3.13), and sweeps are repeated until convergence or until the maximum number of sweeps is reached.

An improved version with a two-site local optimization enables a more stable and efficient implementation of variational truncation algorithms, adapting the bond dimension at each step using the SVD. This implementation is included as part of the SeeMPS [51] library.

The logic behind this variational scheme extends to the approximation of the linear combination of MPS states by modifying the minimization condition to

$$\mathrm{argmin}_{\phi\in\mathrm{MPS}} \left\| \phi - \sum_{l=1}^{L} \alpha_l \psi_l \right\|^2, \tag{3.16}$$

where now the solution is a weighted linear combination of the solution for each state

$$D^i_{\alpha,\beta} = \sum_{n=1}^{L} \alpha_n U^{(n)i}_{\alpha\beta}. \tag{3.17}$$

While the exact linear combination of MPS is possible via the direct sum of their tensors, it results in a new MPS with an increased bond dimension equal to the sum of the tensors of the original states. The variational algorithm in Eq. (3.16) provides an effective method to approximate linear combinations of MPS—and MPO when rewritten as an MPS—, constituting an efficient tool for performing sums and subtractions within the finite precision MPS-MPO algebra.

Similar truncation strategies have been developed in the context of DMRG [108], but a crucial difference is in that the optimizations described above the states under consideration do not share any tensors with the target states, $\hat{O}\,|\psi\rangle$, $|\psi\rangle$ or $|\psi_l\rangle$. Consequently, by using L matrix product states of bond-dimension χ, it becomes feasible to implement an optimization that in DMRG would require environments of size $L \times \chi$ and operations that scale as L^2 times worse, in general [109].

This truncation algorithm completes the MPS-MPO algebra, allowing practical applications by reducing the MPS bond dimension and enabling an efficient performance of sums and subtractions of MPS. Even though the MPS-MPO operations are exact up to machine precision, the truncation of these constructs introduces an error, hence *finite precision*. However, for applications with a suitable entanglement structure, such as the ones in chapters 5 and 6, the error from replacing the MPS and linear combinations with smaller bond dimension approximations is acceptable in favor of higher memory efficiency

3.6 Representation of functions as MPS

While tensor networks were originally developed in the area of quantum many-body physics to represent quantum states, they can also approximate other tensor structures, such as linear operators [38]. The quantum register representation of functions (2.3) directly translates to this formalism by mapping the physical dimension of the tensors to the qubits of the quantum register (Figure 3.11(a)). Then, each site encodes information corresponding to a specific order of significance of the binary encoding of the coordinates (2.2), where the leftmost site encodes the most significant digit, hence the information corresponding to the coarsest grid. The expression of the MPS representing a function is then derived accordingly as

$$|f^{(n)}\rangle = \sum_{\{s\}}\sum_{\{\alpha\}}(A^{s_1}_{\alpha_1}A^{s_2}_{\alpha_1,\alpha_2}\dots A^{s_n}_{\alpha_{n-1}})\,|s_1\rangle\otimes|s_2\rangle\otimes\ldots\otimes|s_n\rangle\,, \tag{3.18}$$

where s denotes the physical indices instead of i to match the quantum register notation. Each tensor $A^{s_k}_{\alpha_{k-1},\alpha_k} \in \mathbb{C}^{2\times\chi_{k-1}\times\chi_k}$ has a bounded size, with dimensions χ_{k-1}, χ_k that depend on the entanglement content. Provided the bond dimension is kept under bounds, the whole representation requires only polynomial many resources $\mathcal{O}(n2\chi^2)$, as opposed to the exponential number of resources of the most common vector representation of functions.

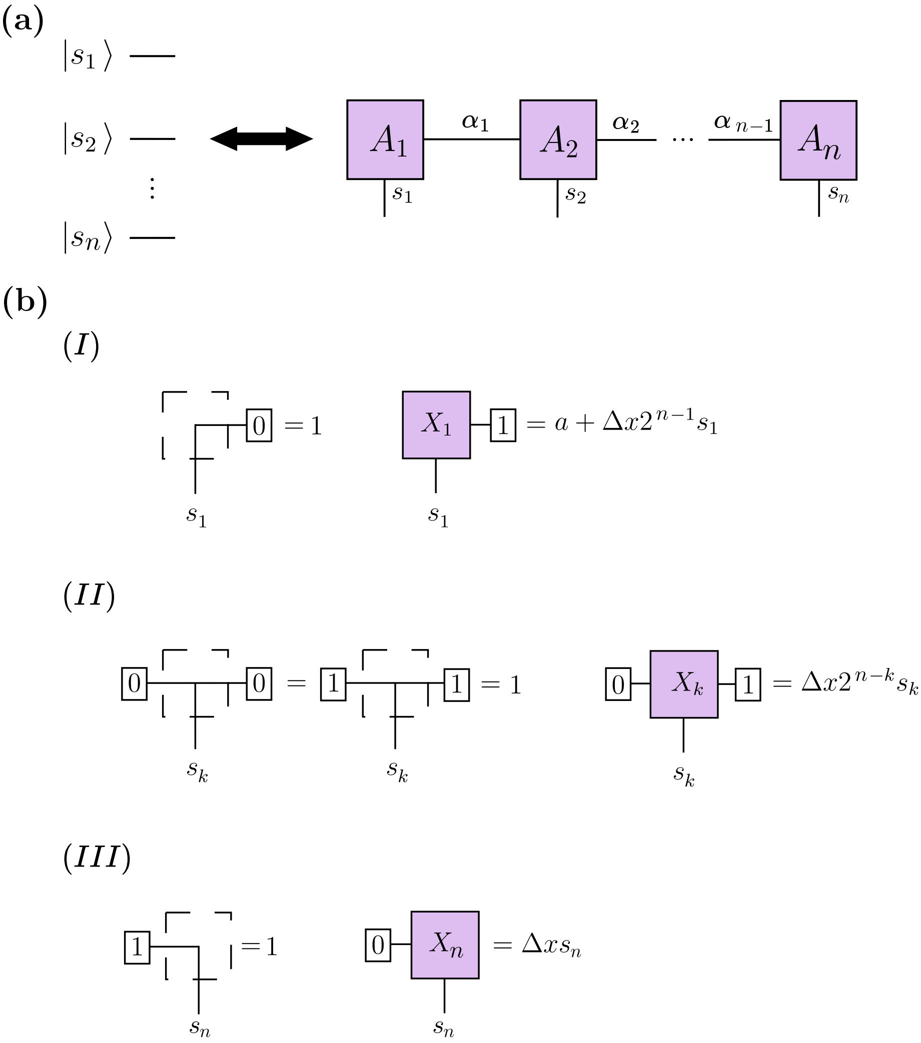


Figure 3.11: MPS representation of functions. (a) Equivalence of a n-qubit quantum register and a n-site MPS. (b) MPS elements for the 1D $f(x) = x$ function with $\chi = 2$. The figure depicts the non-zero tensors' element of the MPS. Middle tensors (II) are rank-3 tensors, while the first (I) and last (III) tensors have rank-2. If the tensor equals a delta function, it is represented as a straight line.

To ease the understanding of this encoding of functions as MPS, let us consider the representation of one of the most basic functions, $f(x) = x$, which can be written as an MPS with bond dimension $\chi = 2$. Figure 3.11(b) depicts the non-zero components of the MPS, showing that a change in the bond index from 0 to 1 corresponds to adding the binary component of the coordinate associated with the given site. For instance, let us construct the coordinate x_s with $s = (s_1, s_2, s_3)$,

$$x_s = \begin{pmatrix} 1 & a + \Delta x 2^2 s_1 \end{pmatrix} \begin{pmatrix} 1 & \Delta x 2 s_2 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \Delta x 2^0 s_3 \\ 1 \end{pmatrix} = a + \Delta x \sum_{k=1}^{3} s_k 2^{n-k}, \tag{3.19}$$

which recovers Eq. (2.2) for $n = 3$. This function is the basis for understanding the MPS representation, which relies on a binary encoding of the coordinates in the physical indices. This is because the function $f(x) = x$ is equivalent to the position space grid for the general definition of functions.

When using MPS to represent functions, the error arising from the truncation of the singular values adds to the common errors in numerical analysis, such as truncation or round-off errors. Understanding this MPS truncation's effect is crucial to choosing a suitable truncation tolerance for function representation tailored to each problem. If the tolerance is too small, the benefit from MPS compared to tensors will be minimal, but excessively large tolerances can result in significant errors. Figure 3.12(a) shows a normalized Gaussian function with varying levels of error truncation tolerance, defined by the minimization distance (3.13). This formulation ensures that the truncation criteria for the SVD and the truncation algorithms in Section 3.5 are identical. Other criteria can be chosen, such as absolute or relative error of the singular values. The results show that the higher the tolerance, the worse the function approximation. Higher truncation of the singular values leads to a higher error in the approximation of the original vector function $f(x)$ by the MPS function $f(x)_{\mathrm{MPS}}$, $\|f(x) - f(x)_{\mathrm{MPS}}\|$, as depicted in Figure 3.12(b). The Gaussian function has a bounded bond dimension (Figure 3.12(c)); as the number of qubits increases, the bond dimension remains constant for a fixed tolerance. This implies that there is a maximum bond dimension necessary to encode this Gaussian function as an MPS, and hence a limit to the memory requirements to encode it up to a certain precision, allowing a memory gain with respect to the vector representation, as discussed in Chapter 5.

The efficient representation of functions as MPS is key for quantum-inspired numerical analysis. The MPS encoding of the position space

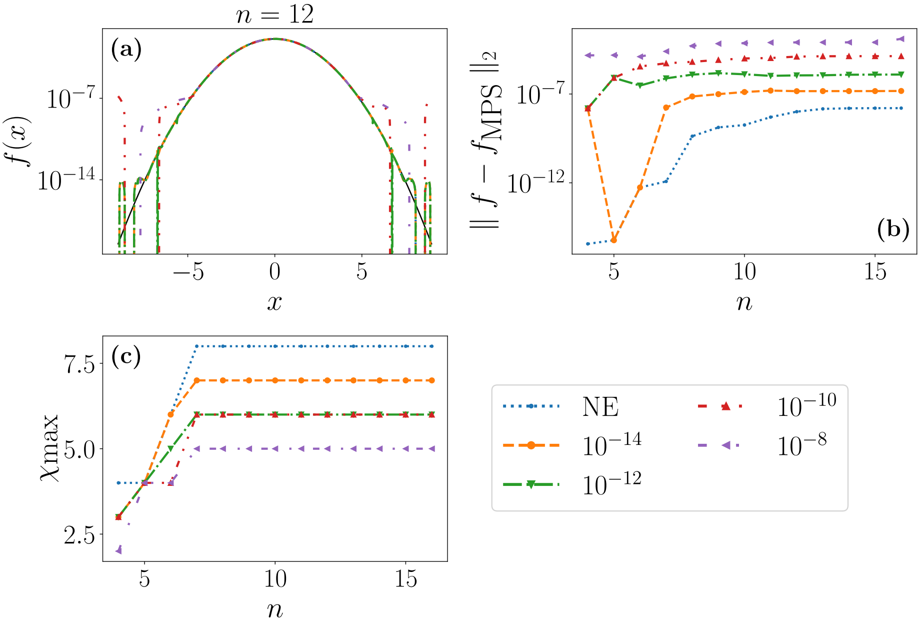


Figure 3.12: MPS representation of Gaussian function. (a) Gaussian representation with different levels of truncation tolerance (3.13), where NE stands for numerically exact. (b) Norm-2 error $\|f - f_{\text{MPS}}\|_2$ scaling with the number of qubits n, i.e., sites. (c) Maximum bond dimension χ_{max} scaling with the number of qubits n. The tolerance for the simplification algorithms' convergence is NE.

coordinates (Figure 3.11) enables the exact construction of functions, such as the exponential e^x, which is a product MPS, or the sine and cosine functions, which are linear combinations of exponentials and can be computed via the linear combination truncation algorithm in Section 3.5 or the exact direct sum and later simplification. Similarly, it is possible to encode the momentum space coordinate p (2.20) as an MPS with bond dimension $\chi = 2$ to extend this MPS-MPO algebra to spectral applications. There exists no general recipe for constructing functions as MPS. The most straightforward approach involves the direct construction of the MPS from the vector representation using the SVD, but this limits the maximum number of sites to the capacity of the vector encoding. To overcome this limitation, interpolation techniques, such as Chebyshev series [145, 52] or TT-cross interpolation [146, 147], offer an MPS approximation of the desired function.

To express multidimensional functions as MPS we assign a set of MPS sites to each dimension, similar to the quantum register case. In the MPS formalism, the map of the coordinates to the sites determines the entanglement of the resulting MPS [38], playing a role in the bond dimension of the represented function and the associated computational requirements. The straightforward order (A) distributes the sites sequentially, first by coordinate, then by significance. An alternative encoding—order (B)—involves sorting sites by significance first, concentrating the largest scales in neighboring sites in the MPS. For example, for a two-dimensional function with three qubits per dimension, the resulting orders would be

$$\text{Order A:}|s_1 s_2\rangle \quad \rightarrow \quad |s_1^1\rangle|s_1^2\rangle|s_1^3\rangle|s_2^1\rangle|s_2^2\rangle|s_2^3\rangle, \tag{3.20}$$

$$\text{Order B:}|s_1 s_2\rangle \quad \rightarrow \quad |s_1^1\rangle|s_2^1\rangle|s_1^2\rangle|s_2^2\rangle|s_1^3\rangle|s_2^3\rangle. \tag{3.21}$$

Although not all functions are appropriate for an MPS representation due to bond dimension requirements, tensor networks have been successfully applied for a variety of numerical analysis problems, such as the solution of PDEs [44, 45, 38, 46] or interpolation [146, 147, 38]. García-Ripoll [38] showed that for a Grover-Rudolph-like state [48] representing a highly differentiable function, adding new qubits entails a vanishingly small amount of entanglement. Jobst [39] extended this to functions with fast-decaying Fourier coefficients, establishing a theoretical bound for the bond dimension.

3.7 Representation of operators in an MPS-MPO framework

The MPS-MPO framework can also represent numerical analysis operators in function space. As in the quantum register representation in chapter 2, we distinguish two types of linear operators: differential and potential operators.

An MPO representation for differential operators can be recovered in many different ways. Two of them are based on the spectral techniques and finite-difference approximations discussed in section 2.2. Let us focus on approximating the second order derivative ∂_x^2 since this differential operator is present in the static and time-dependent PDEs studied in chapters 5 and 6.

The spectral method approximates the differential operator ∂_x^2 as

$$\partial_x^2 f(x) = \frac{1}{\sqrt{N}} \sum_{l=-N/2}^{N/2-1} p^2 e^{ip_l x} \tilde{f}(p_l). \tag{3.22}$$

While the Fourier space operator p can be represented as an MPO with bond dimension $\chi = 2$, a naive construction of the QFT's MPO from its quantum register representation leads to a bond dimension linear in its number of sites n. This bond dimension's scaling, together with the successive MPO-MPS contractions and simplifications, renders the method costly. A careful consideration of the bit reversal operation results in a more efficient implementation, as was shown in Ref. [148]. The spectral method could be extended for general differential operators $G(\partial_x)$ as long as they can be efficiently represented in the MPS-MPO framework in momentum space.

The finite difference representation of the differential operator ∂_x^2 relies on the displacement operators $\Sigma^{\pm}$

$$|\partial_x^2 f^{(n)}\rangle \simeq \frac{1}{\left(\Delta x^{(n)}\right)^2} \left(\hat{\Sigma}^{+} - 2\mathbb{I} + \hat{\Sigma}^{-}\right) |f^{(n)}\rangle \,. \tag{3.23}$$

This operation can be efficiently written as an MPO of weighted displacements with fixed bond dimension $\chi = 3$, hence reducing the cost in the approximation of derivatives with respect to the spectral method. However, the finite difference method's accuracy scales polynomially with the discretization size, leading to an error significantly larger than spectral techniques. In chapter 6, we extend the Hermite Distributed Approximated Functionals (HDAFs) [149] to an MPS-MPO framework. This results in a new method to approximate functions of derivatives with pseudospectral accuracy and a reduced cost similar to the finite difference approximation.

In the position basis, the multiplication by a continuous potential function $V(x)$ is a diagonal operator

$$\hat{V}(x) := \sum_{s=0}^{2^n-1} V\left(x_s^{(n)}\right) |s\rangle\langle s| \,. \tag{3.24}$$

This operator is a diagonal matrix in function space, and hence, it could be reduced to a vector, substituting the matrix-vector multiplication with an elementwise vector multiplication. When represented as an MPS following the encoding in Figure 3.11(b), the operator's application on the function would be performed as the contraction of two MPS as depicted in Figure 3.9(d), hence reducing its cost from $O(Nd^2\chi^4)$ to $O(Nd\chi^4)$. The MPS representation of diagonal operators enables the use of the Chebyshev and TT-cross interpolation techniques for its construction.

Chapter 4

Quantum numerical analysis for static PDEs

Numerical analysis is a powerful tool that, via the solution of linear equations, ordinary differential equations (ODEs), and partial differential equations (PDEs), assists progress both in basic science—e.g., design of photonic materials, understanding of elasticity, modeling black holes—and engineering. Despite significant advances in the theoretical understanding of PDEs, solving them remains challenging. For instance, the design of superconducting quantum computers depends on accurately modeling superconducting qubits, couplers, and measurement devices, which involve multidimensional, nonlinear Hamiltonian PDEs. These equations are particularly difficult to solve due to the increasing complexity associated with both the number of variables—which drives up memory requirements—and the inherent difficulty of the equations themselves. Indeed, specialized libraries [150, 151, 152], which rely on state-of-the-art sparse matrix diagonalization techniques, restrict the complexity of the superconducting circuits due to memory limitations.

Quantum computers' potential to achieve exponential advantages in both data encoding—i.e., memory—and computational performance offers the possibility to enhance solving Hamiltonian PDEs and similar numerical analysis problems. This results in the field of quantum numerical analysis, which uses quantum computers to tackle mathematical problems such as linear systems of equations [153, 154, 155] and partial differential equations [156, 157]. Quantum algorithms for numerical analysis traditionally focus on a fault-tolerant scenario, with proposals [5, 7, 8] based on the Harrow-Hassidim-Lloyd (HHL) [9] algorithm for linear systems of equations, or using quantum amplitude estimation (QAE) [17] techniques [12], among others. However, since quantum computers have not yet reached this stage, the scientific community has researched hybrid quantum numerical analysis algorithms [24, 23, 25, 26, 27, 28, 31, 32, 33, 29, 30, 34, 35, 36] that combine the power of (imperfect) NISQ [18] devices with classical computers. This includes, for instance, variational algorithms and approximate techniques, such as error mitigation and iterative phase estimation, to compensate for the imperfections. These proposals motivated us to create a new quantum algorithm for numerical analysis: a variational quantum algorithm for solving static Schrödinger-type Hamiltonian PDEs (4.1) [46].

This chapter aims to develop a variational quantum algorithm for solving static Schrödinger-type Hamiltonian PDEs (4.1) to address the simulation of superconducting circuits. This method was originally proposed in Ref. [46] and constitutes the first work of this thesis. The quantum Fourier analysis techniques in chapter 2 are the basis for the creation of this algorithm. These are combined with new suitable variational ansätze (Section 4.3) for the representation of functions and their symmetries as quantum circuits and global and gradient-based classical optimizers. The results showcase the high-precision of quantum Fourier analysis techniques for representing functions and operators in a quantum register, achieving low infidelities, even up to $O(10^{-4})$ under the effect of noise sources for a small number of qubits. However, the impact of noise sources and the limited number of measurements of current quantum computers restrain the performance of the Fourier techniques. This opens new avenues for using more efficient quantum-inspired algorithms to take advantage of this encoding in chapter 5.

4.1 Application: superconducting qubits

Just as bits are classical computers' fundamental building blocks, qubits play the same role in quantum computers. This technology is still in development, and one of the principal bets for its implementation relies on superconducting qubits. Superconducting qubits are quantum circuits that exhibit quantum phenomena such as energy quantization, reversible unitary evolution, quantum superposition, and entanglement [3]. The main components of superconducting circuits are capacitors, inductors, and Josephson junctions [158], leading to various qubit designs such as the charge qubit [159], the transmon qubit [1], and the three Josephson junction flux qubit [2]. Superconducting qubits require extremely low temperatures—typically on the order of millikelvin—to operate effectively. These ultra-low temperatures suppress decoherence and thermal fluctuations, preserving the fragile quantum states essential for quantum computation [3].

The Kirchoff analysis of a superconducting electrical circuit results in a set of ordinary differential equations for intensity, voltage or flux variables, with an associated Lagrangian and Hamiltonian due to their conservative nature. The quantization of the circuit—justified because of the superconductor's coherent nature and low operation temperature—leads to a quantum version of the Hamiltonian. The dynamics and static properties of the circuit can now be described with a set of static or time-dependent PDEs for the collective wavefunction. The time-independent Schrödinger equation gives the energy levels of the qubit

$$Hf(\mathbf{x}) = Ef(\mathbf{x}), \tag{4.1}$$

where E is the energy associated to the state $f(\mathbf{x})$ for the Hamiltonian operator

$$H = D(-i\nabla) + V(\mathbf{x}). \tag{4.2}$$

The complexity of the eigenvalue problem in Eq. (4.1) grows exponentially with the circuit size. When addressing these equations with traditional numerical analysis techniques—e.g., finite differences, finite elements, or spectral methods—the problem's size grows exponentially, too fast for ordinary computers and standard diagonalization techniques [150, 151, 152].

To address this challenge, we propose a new variational quantum algorithm: the variational quantum PDE solver [46], specifically designed for solving static Schrödinger-type Hamiltonian PDEs, as described in section 4.4. This algorithm encodes the PDE into a parameterized quantum circuit, with a classical optimizer variationally modifying the circuit's parameters to minimize the Hamiltonian energy, represented as a cost function. The encoding relies on the quantum register representation of functions and operators in chapter 2, combined with new quantum ansätze to represent functions and their symmetries.

As a benchmark for the method, we solve the Hamiltonian PDE of three of the most studied superconducting circuits: the LC-resonator, the transmon, and the flux qubit. Note that these benchmarks are all one-dimensional problems, but, as discussed in section 2.4, the quantum register representation, and consequently, the variational quantum PDE solver, can be generalized to higher-dimensional problems.

Let us first analyze the simplest quantum circuit: the LC-resonator (Figure 4.1(a)). It consists of a superconducting resonator—parallel inductor L and capacitor C—coupled to a constant voltage source. In the absence of charge bias, its Hamiltonian PDE in terms of the phase variable φ is

$$H = -E_C \partial_\varphi^2 + \frac{1}{2} E_L \varphi^2, \tag{4.3}$$

where $E_C = 2e^2/C$ is the capacitive energy with e the elementary charge, and $E_L = \Phi_0^2/(8L\pi)$ is the inductive energy with Φ_0 the flux quanta. This is equivalent to a quantum harmonic oscillator with frequency $\omega = 1/\sqrt{LC}$ and mass $m = \hbar^2/(8E_C)$, whose Hamiltonian is given by

$$H = -\frac{\hbar^2}{2m} \partial_x^2 + \frac{1}{2} m\omega^2 x^2. \tag{4.4}$$

Therefore, the LC resonator has exact solutions

$$f_n(x) = \left(\frac{\beta^2}{\pi}\right)^{1/4} \frac{1}{\sqrt{2^n n!}} e^{-\beta^2 x^2/2} H_n(\beta x), \tag{4.5}$$

where $\beta = \sqrt{m\omega/\hbar}$ and H_n is the Hermite polynomial of order n. For $\hbar = \omega = m = 1$, the exact ground state of the one-dimensional quantum harmonic oscillator is

$$f_0(x) = \frac{1}{\pi^{1/4}} e^{-x^2/2}, \tag{4.6}$$

which is a trivial Gaussian (Figure 4.2(a)). The corresponding eigenenergies are given by $E_n = (n + \frac{1}{2})$, with ground state energy $E_0 = 0.5$.

The existence of an analytic solution, together with its simplicity and many applications—molecular vibrations [160], quantum optics [161], etc—besides superconducting quantum circuits, make the quantum harmonic oscillator equation a convenient choice to calibrate the accuracy of the Fourier encoding and the variational solution of the PDE. From now on, we will use the more general quantum harmonic oscillator formulation to study this example.

The study of the harmonic oscillator serves as a basis for the transmon qubit [1], a simple nonlinear version of the LC circuit. As depicted in Figure 4.1(b), the transmon qubit combines a superconducting island coupled to a ground plane via a Josephson junction with an externally controlled voltage source [3]. The Hamiltonian PDE of a superconducting transmon qubit [1] without charge offset is

$$\left[-4E_C \partial_\varphi^2 - E_J \cos(\varphi) - E\right] f(\varphi) = 0, \tag{4.7}$$

where the phase variable is periodic over the interval $\varphi \in [-\pi, \pi)$. The model's parameters are the Josephson energy E_J and the capacitive energy E_C. The transmon operates in the regime $E_J/E_C \gtrsim 50$ [162]. The

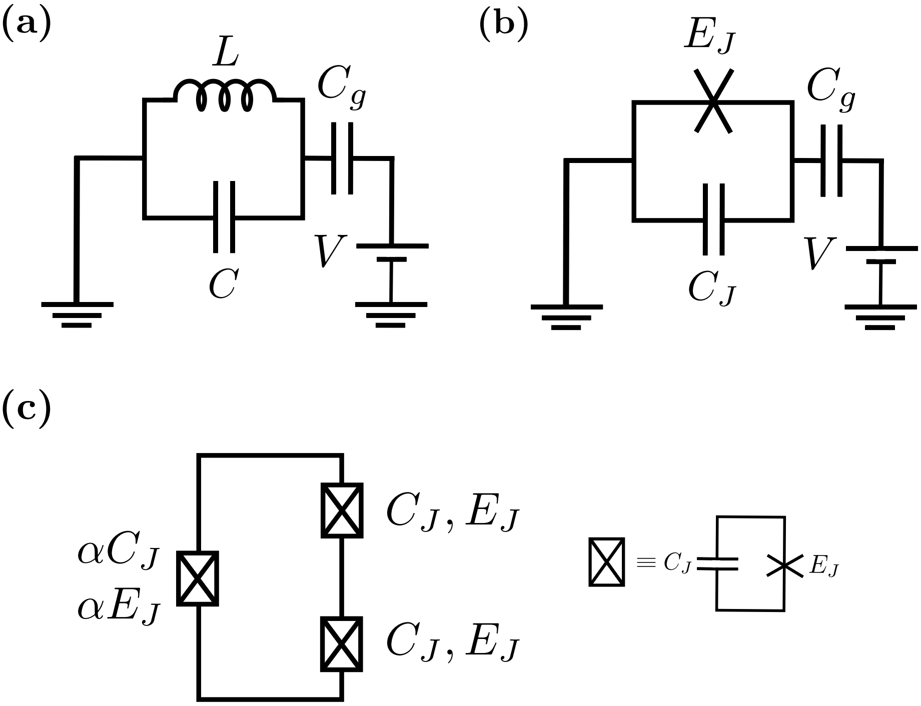


Figure 4.1: Superconducting circuits. (a) LC-resonator with an inductor L in parallel with a capacitor C, gate capacitor C_g and voltage source V. (b) Transmon qubit with junction's energy E_J, junction's capacitance C_J, coupling capacitance C_g and voltage source V. (c) Three Josephson-junction flux-qubit with two equal junctions with E_J, C_J and one modulable junction with $\alpha E_J, \alpha C_J$, $\alpha \sim 0.7 - 0.8$.

eigenfunctions of the transmon qubit PDE are the Mathieu functions, i.e., the analytical solutions of Mathieu's differential equation, given by

$$\frac{d^2}{dx^2} y(x;q) + (a - 2q\cos 2x) y(x;q) = 0. \tag{4.8}$$

Thus, the eigensolutions and eigenergies for a transmon without charge offset are [3]

$$\psi_n(\varphi) = \frac{1}{\sqrt{2}} me_{k(n,0)}\left(\frac{\varphi + \pi}{2}; \frac{E_J}{2E_C}\right), \tag{4.9}$$

$$E_n = \mathcal{M}_A(r,q), \tag{4.10}$$

where $\mathcal{M}_A(r,q)$ is the Mathieu characteristic function for even functions and $me_{\nu+2k}(z;q)$ are the Floquet functions, with $\nu + 2k$ being the characteristic exponent of the function after a period π. The index k is a function expressed as (valid for all values except $2n_g$) [3]

$$k(n, n_g) = -\text{round}(n_g) + \frac{1}{4}(-1)^{\text{floor}(n_g)}[-1 + (-1)^n(1 + 2n)]. \tag{4.11}$$

The functions round(x) and floor(x) correspond to the nearest integer to the real number x and the integer immediately below x, respectively. In our benchmarks $n_g = 0$, since there is no charge offset.

In the large limit of E_J/E_C, it is possible to expand the problem around $\varphi = 0$,

$$H = -4E_C\partial_\varphi^2 + \frac{1}{2}E_J\varphi^2, \tag{4.12}$$

which turns the transmon qubit Hamiltonian into a harmonic oscillator with

$$\frac{\hbar^2}{2m} \sim 4E_C, \quad \frac{1}{2}E_J \sim \frac{1}{2}m\omega^2, \tag{4.13}$$

and effective frequency $\hbar\omega = \sqrt{8E_C E_J}$. Hence, the transmon ground state (Figure 4.2(b)) can be approximated by a Gaussian function in some operational regimes, with

$$\psi(\varphi) \propto \exp\left[-\frac{1}{2}\left(\frac{\varphi}{a_0}\right)^2\right], \quad a_0^4 = \frac{8E_C}{E_J}. \tag{4.14}$$

This Gaussian approximation fails to capture the nonlinear contributions and does not consider the periodicity of the function $f(\varphi)$. Due to this periodicity, the definition interval of the position space is fixed to 2π.

Let us introduce a third superconducting circuit: the three-junction flux qubit. This consists of a superconducting loop with three Josephson junctions, with a lumped-element model, as shown in Figure 4.1(c) [2]. This is a more complex model whose ground state is not a Gaussian and cannot be approximated by one, hence constituting a more challenging problem for our method. The one-dimensional reduction of the three-junction flux qubit leads to the Hamiltonian PDE [163]

$$\left[-\frac{E_c}{\frac{1}{2}+\alpha}\partial_\varphi^2 - E_J\left[2\cos(\varphi) - \alpha\cos(2\varphi)\right] - E\right] f(\varphi) = 0, \tag{4.15}$$

which has no analytical solution. The small junction size takes typical values $\alpha \sim 0.7 - 0.8$. In this case, the inductive potential $V(\varphi) = -E_J(2\cos(\varphi) - \alpha\cos(2\varphi))$ creates two minima, leading to a ground state delocalized between those minima (Figure 4.2(c)). The gap between the

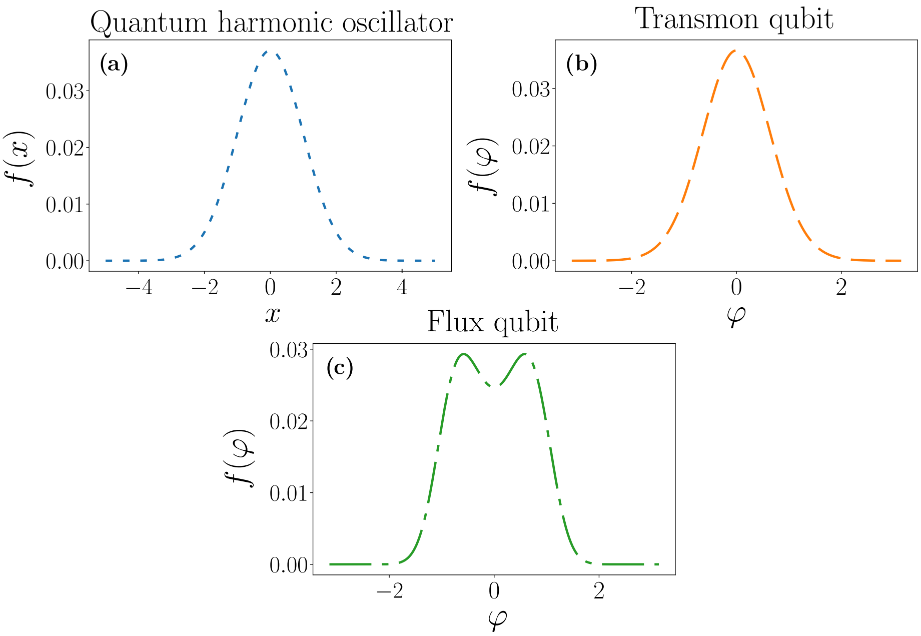


Figure 4.2: Ground state of Hamiltonian benchmark problems. (a) The quantum harmonic oscillator. (b) The transmon qubit. (c) The flux qubit.

model's ground and first excited states is smaller than that of the transmon, thus requiring better estimates of the energy functional to distinguish the ground from the excited states.

The ground states of the circuit models and their associated PDEs—quantum harmonic oscillator, transmon qubit, and flux qubit—are simple, infinitely differentiable functions. As sketched in Figure 4.2, these ground states can also be formulated as real, symmetric, and even wavefunctions. These properties, together with the fact that these solutions also vanish to zero towards the limit of the interval, make them appropriate for the Fourier representation techniques in chapter 2 and the circuit ansätze in section 4.3.

4.2 Quantum numerical analysis

Since the inception of quantum computing, researchers have investigated their application to solve complex mathematical problems. One of these applications was the encoding and manipulation of discretized functions as states of the quantum register [47, 48]. These new techniques opened the possibility of using quantum computers to assist in numerical analysis tasks. There are many applications of quantum numerical analysis, such as the solution of linear systems of equations [9], matrix diagonalization [164], and interpolation [165]. Due to their ubiquitous nature in science and engineering, a promising application of quantum computers has been the solution of differential equations. One of the first quantum speed-ups in this field was proposed by Kacewicz, solving initial-value problems for ordinary differential equations (ODEs) using quantum amplitude estimation (QAE) [17] as a subroutine within a classical method [12]. This proposal has led to exciting new quantum algorithms based on QAE for the Navier-Stokes PDEs [13, 14] and the Burger's equation [15], and an improved version based on Chebyshev points tested for the heat and convection-diffusion equations [16]. Leyton and Osborne also developed an algorithm for gate-based quantum computers to solve nonlinear ordinary differential equations using a quantum implementation of the Euler method [166].

The creation of the Harrow-Hassidim-Lloyd (HHL) algorithm [9] for solving linear systems of equations proportionated a potential tool for developing quantum algorithms for PDEs. Notably, the HHL algorithm is a primitive that can implement many traditional methods to solve systems of coupled differential equations. Berry [5] led the development of these methods, transporting the quantum speedups of the HHL algorithm to the solution of ODEs, combining it with the Euler method and quantum simulation. This approach was upgraded, improving its precision [7] and extending it to PDEs [8]. This technique can be used together with the finite-element method [6] and spectral methods [8], and is suitable for a variety of linear problems: the Poisson equation [10], the heat equation [11] and the wave equation [167, 168], which is successfully simulated in Ref. [169]. These ideas, when combined with the Carleman [170] or the quantum nonlinear Schrödinger linearization [171], also extend to weakly nonlinear differential equations. Additionally, alternatives based on hardware-efficient Taylor expansions [172], Runge-Kutta methods on quantum annealers [173], and the intrinsic dynamics provided by continuous-variable quantum computers [174] also stand out.

The previous methods are fault-tolerant quantum algorithms. An alternative approach that might be more suitable for current imperfect quantum computers would be to adopt a model of hybrid quantum-classical computation, where the quantum computer assists with tasks such as encoding the solution to a complex problem, and the classical computer provides an optimization algorithm that brings this encoding closer to the sought state. This approach, known as "variational quantum algorithms" [175, 22] (VQAs in short), constitutes the basis for many applications of quantum computers in domains such as condensed matter and quantum chemistry [175, 176, 177, 178], quantum optimization [179, 180, 181, 182, 183, 184], and quantum machine learning [185, 186, 58, 187, 188, 189, 190, 191, 192, 193, 194, 195]. What follows is a presentation of VQAs that provides a background focusing on applications to numerical analysis. See Ref. [22] for more information on this topic. VQAs combined with the variational encodings from section 4.3 are a central tool for our study of a solver for Hamiltonian PDEs (4.1), discussed later in Section 4.4.

Let us suppose a time-independent Hamiltonian $\hat{H}$ with a non-degenerate spectrum bounded from below, with greatest lower bound E_0. The expectation value of $\hat{H}$ and a normalized quantum state $|\psi\rangle$ verifies

$$\begin{aligned} \langle\psi|\,\hat{H}\,|\psi\rangle &= \sum_n |c_n|^2 E_n = |c_0|^2 E_0 + \sum_{n>0} |c_n|^2 E_n \\ &= E_0 + \sum_{n>0} |c_n|^2 (E_n - E_0) \quad \rightarrow \quad \langle\psi|\,\hat{H}\,|\psi\rangle \geq E_0, \end{aligned} \tag{4.16}$$

where $|\psi\rangle$ is expressed in the basis of eigenstates of $\hat{H}$, $\{|\varphi_n\rangle\}$, as $|\psi\rangle = \sum_n c_n|\varphi_n\rangle$. To minimize the separation between $\langle\psi|\,\hat{H}\,|\psi\rangle$ and E_0, we aim to minimize the coefficients $|c_n|^2$. Since the form is non-negative, this process effectively reduces the distance between $|\psi\rangle$ and the ground state.

Then, by allowing $|\psi\rangle = |\psi(\alpha)\rangle$, i.e., for the state to be a parameterized ansatz, the variational method constitutes a systematic approach to find the approximate eigenenergies and eigenvalues of $\hat{H}$ [196]. This method is particularly effective for low-energy states, and the choice of ansatz is crucial to correctly targeting the solution's subspace of the Hilbert space. For the state to converge to the ground state, $c_0 = \langle\varphi_0|\psi\rangle \neq 0$, i.e., the state $|\psi\rangle$ must have a nonzero superposition with the ground state of $\hat{H}$.

The variational method can be utilized in quantum computing by employing variational trial states created within a quantum computer. Variational quantum algorithms (VQAs) [22] are algorithms that use a family of parameterized quantum circuits to optimize a cost function over the set of states that can be created by those circuits. When the cost function is the Hamiltonian of some problem, the variational quantum eigensolver (VQE) [175] provides, by virtue of the variational principle, an estimate of that Hamiltonian's ground state energy. Figure 4.3 depicts

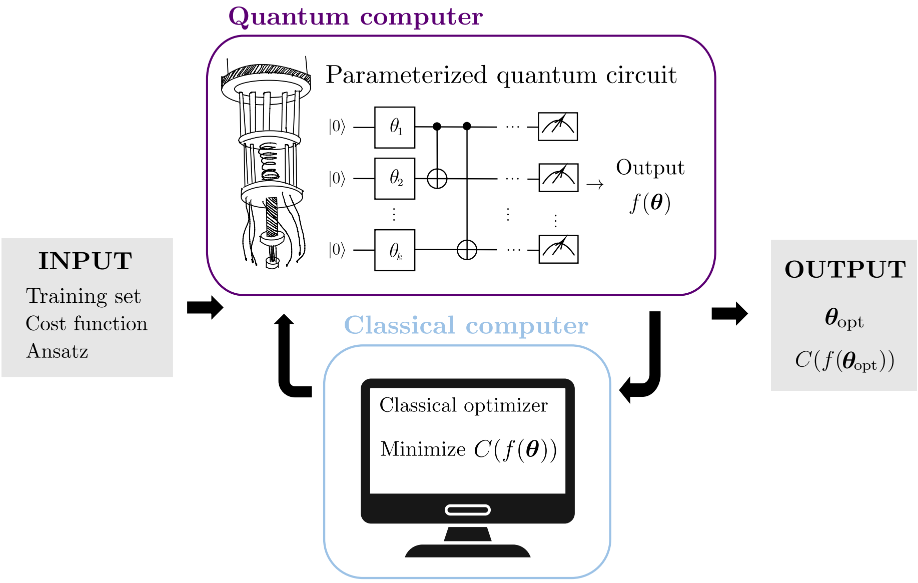


Figure 4.3: Variational quantum eigensolver scheme. The quantum computer encodes the problem ansatz and computes the values for the cost function. In contrast, the classical computer performs the optimization, which minimizes the cost function value obtained from the quantum computer measurement.

the basic working scheme of VQAs. The hybrid quantum algorithm operates by tuning the parameterized quantum circuit towards the solution of a complex problem, whereby the tuning's responsibility is typically the role of a classical optimization algorithm executed in a conventional computer. The main elements of VQAs are listed below.

- **Variational quantum circuit.** A quantum circuit $\left|f_{\boldsymbol{\theta}}^{(n)}\right\rangle$ parameterized by the angles of the rotational quantum gates $\boldsymbol{\theta} = (\theta_1, ..., \theta_n)$. It is also known as circuit ansatz, and its structure, number of parameters, and qubits depend on the concrete problem to solve. A non-parameterized circuit can precede it to initialize the circuit to a desired state.

- **Cost function.** The minimization objective function of the problem $C(\boldsymbol{\theta})$. Its minimum is either the problem's solution or the quantum state that best approximates it. The cost function is usually some metric whose value is computed from the measurements of the quantum circuit.

- **Classical optimizer.** The process of classical optimization entails minimizing the cost function by adjusting circuit parameters until convergence. The update rule for generating new parameters during each iteration depends on the optimizer chosen. This election is optimal for each application.

Among classical optimizers, one may distinguish between gradient-based and gradient-free optimizers. Gradient-based optimizers—which rely on gradient computations to determine the direction and magnitude of parameter updates—are a common choice in VQAs due to their efficiency and fast convergence for smooth landscapes. When choosing an optimizer, it is essential to consider the number of quantum circuit evaluations, leading to options such as Simultaneous Perturbation Stochastic Approximation (SPSA) [197], which only needs two evaluations per step. More complex methods, such as adaptative moment estimation (Adam) [198], may result in more precise solutions at the cost of more evaluations. The gradient computation can also be improved using the parameter shift rule [199, 200], which allows for an analytical gradient. State-of-the-art optimization techniques, such as automatic differentiation [201], are also considered, aiming to maximize the cost-accuracy relation of the implementations.

VQAs generated significant interest in the early stages of NISQ computation, as an algorithm that can be cheaply implemented in hybrid quantum computers and could potentially provide an advantage over classical alternatives—e.g., thanks to the exploring and sampling capabilities of the quantum circuit in an exponentially large Hilbert space. One of their primary applications is searching for ground and excited states of quantum and even classical Hamiltonians, which are crucial in physics and chemistry. VQE [175] plays a very relevant role, addressing the challenge of minimizing the energy $E(\boldsymbol{\theta}) = \langle \psi(\boldsymbol{\theta}) | \hat{H} | \psi(\boldsymbol{\theta}) \rangle$ of a given system Hamiltonian H. In most applications, the Hamiltonian is a linear combination of Pauli gates and the energy is a linear combination of its expectation values, which can be efficiently evaluated in a quantum computer with polynomial resources [22]. In practice, these Hamiltonians are usually sparse, leading to a polynomial cost with the system size for the cost function [22]. This scheme can be extended to other problems, such as the computation of excited states [202, 203, 204]. VQE, and VQAs in general, have been very popular to tackle problems in condensed matter and quantum chemistry [176, 177, 178], in which dealing with many-body systems with classical techniques becomes rapidly unfeasible.

Despite the considerable effort in developing VQAs, these algorithms face significant challenges, which are one of the leading research focuses in this field. These challenges can be summarized in three key concepts [22]: (i) trainability, (ii) efficiency, and (iii) accuracy. Trainability accounts for the ease of the ansatz to fit the data. To maximize trainability, choosing a suitable ansatz for each problem and a good set of initial parameters for the training is critical, thereby avoiding the algorithms getting stuck in local minima. These strategies may help elude barren plateaus [205]—areas with flat gradients in the search space—, preventing the algorithms from finding the true minima of the cost function. Gradient-free optimizers, though generally less efficient than gradient-based optimizers, do not suffer from barren plateaus. Another issue is efficiency, which is related to the number of measurements of the quantum circuit. The more measurements made, the closer the results are to the actual expectation value of the operators, according to the law of large numbers. However, access to NISQ devices is limited, leading to a limited number of measurements, which may be insufficient. Finally, we are also battling the accuracy obtained with these techniques. As mentioned, hardware noise is an issue to consider. However, for some applications, VQAs have shown noise resilience [20, 21] since the change experienced in the parameter landscape can still lead to an optimum set of parameters if the barren plateaus are not considerably increased. It is also possible to resort to error mitigation techniques [206, 207, 208] to reduce the effect of noise in these algorithms.

While developing VQAs presents challenges, the exploration of methods to address them, along with the potential enhancements with existing quantum devices, has led to an increasing interest in its use for quantum numerical analysis algorithms for the NISQ era. As in standard VQAs, a variational quantum circuit encodes the solution to a complex problem (an ODE or PDE in our case) and the circuit's parameters are adjusted through a learning process that optimizes a loss function. In some cases, the variational form encodes the complete function [24, 23]. In contrast, in others, the variational circuit acts as a quantum neural network that, given the right coordinates, outputs a prediction for the function under study [25, 26]. These VQAs may work with both nonunitary [24, 27, 28] and nonlinear differential equations [23, 209]. Some applications include the solution of the Poisson equation [31, 32], the advection-diffusion equation [33], the Feynman-Kac equation [29] and other stochastic PDEs [30], the heat equation [34], and fluid dynamics [35, 36]. Other approaches tackle the problem from a machine learning perspective, combining quantum algorithms with machine learning techniques [210, 211].

In this context, we create a novel VQE to solve Hamiltonian PDEs: the variational quantum PDE solver [46]. As described before, the algorithm's definition requires us to introduce the problem encoding, variational circuits, cost function, and the hybrid quantum-classical optimization stages. As opposed to many VQAs for PDEs [23, 34, 210, 31, 29, 209], which rely on the finite difference approximation, our proposal uses spectral methods reducing the error in operator representation and enabling efficient encoding via the quantum Fourier transform. This encoding also accurately approximates denser grids without increasing the number of optimization parameters via Fourier interpolation.

Standard VQAs for PDEs typically rely on general hardware-efficient ansätze [25, 32, 30, 29, 36], which are not optimized for function encoding and require the exploration of the complete Hilbert space. Other alternatives have been proposed, such as MPS-based variational circuits for better function representation [23, 209], but the function's entanglement still limits them. Inspired by the work of Zalka [47] and Grover and Rudolph [48]—focused on discretizing non-negative probability distributions in quantum registers—, we have developed a more suitable ansatz for function representation that is exact up to discretization. This ansatz is further enhanced by a novel quantum circuit that encodes the symmetries of the functions, decreasing the scaling of parameters. Finally, we integrate the quantum problem encoding with different classical optimizers to effectively solve a range of Hamiltonian PDEs in superconducting circuits, leveraging quantum computers to model their very components.

4.3 Quantum circuits for numerical analysis

This section introduces a series of quantum circuits that can be combined with VQEs to solve static PDEs. Mapping continuous functions to a quantum register state (Chapter 2) is the first step in solving numerical analysis problems in a quantum computer. However, in practical applications, the functions must be encoded as a set of quantum gates. These gates construct parameterized quantum circuits acting as variational ansätze for encoding the functions. Choosing the right ansatz is essential for the success of the problem, and factors like problem constraints and the nature of the solutions determine it. Thus, it is reasonable to design an ansatz suitable for function representation with an adequate balance of expressibility and trainability.

The R_Y ansatz [179], also known as real amplitudes ansatz, which is vastly used in VQE applications [176], allows testing the performance of the quantum register representation for a general scope ansatz. This circuit, suitable for representing real states, searches for them in the entire Hilbert space. Then, the solution of the PDE is reachable within the search space of the R_Y ansatz as long as it is real. A new ansatz tailored to the representation of functions—inspired by the one proposed in Refs. [47, 48]—has the potential to take advantage of the Fourier encoding of functions in a quantum register. Moreover, including the symmetries as a set of gates added to the quantum circuit decreases the number of parameters necessary to represent a function and, hence, the cost of adding extra gates, their implementation, associated errors, and optimization difficulty. Finally, the Fourier interpolation quantum circuit can refine the function's representation by combining the QFT (2.18) with the addition of auxiliary qubits. These circuit ansätze fulfill the requirements for encoding the ground states of the benchmark PDEs in section 4.1.

4.3.1 R_Y ansatz

The R_Y ansazt is a general multi-purpose ansatz that pertains to the class of hardware efficient ansätze. This quantum circuit originates in chemistry applications [176]. Still, its large expressibility and chaotic nature, which allows a parameter search through the whole Hilbert space, make it suitable for more tasks, such as machine learning and optimization problems [212, 213, 214], and the solution of PDEs [36]. The R_Y ansatz's reachability motivates its use for encoding functions in quantum numerical analysis applications.

The R_Y ansatz combines layers of local, real-valued parameterized R_Y rotations with controlled-NOT (CNOT) gates, leading to the circuit depicted in Figure 4.4(a). This ansatz results in a parameterized quantum state

$$\left|f_{\boldsymbol{\theta}}^{(n)}\right\rangle = \prod_{q=0}^{n-1} R_Y^q\left(\theta_q^{\text{depth}+1}\right) \prod_{d=1}^{\text{depth}} \left[\prod_{c=0}^{n-1}\prod_{c<t} \text{CNOT}_{c,t} \prod_{q=0}^{n-1} R_Y^q\left(\theta_q^d\right)\right] |0\rangle^{\otimes n}, \tag{4.17}$$

where the parameters are the rotation angles of the $R_Y(\theta) = \exp(-i\theta\sigma^y/2)$ gates. In this ansatz, the number of parameters scales linearly with the number of qubits, preventing its cost from exploding with the circuit size. However, when initialized with random parameters, the likelihood of a nonzero gradient along any reasonable direction to a certain precision decreases exponentially as the number of qubits increases, leading to trainability issues due to barren plateaus in the parameters landscape [215].

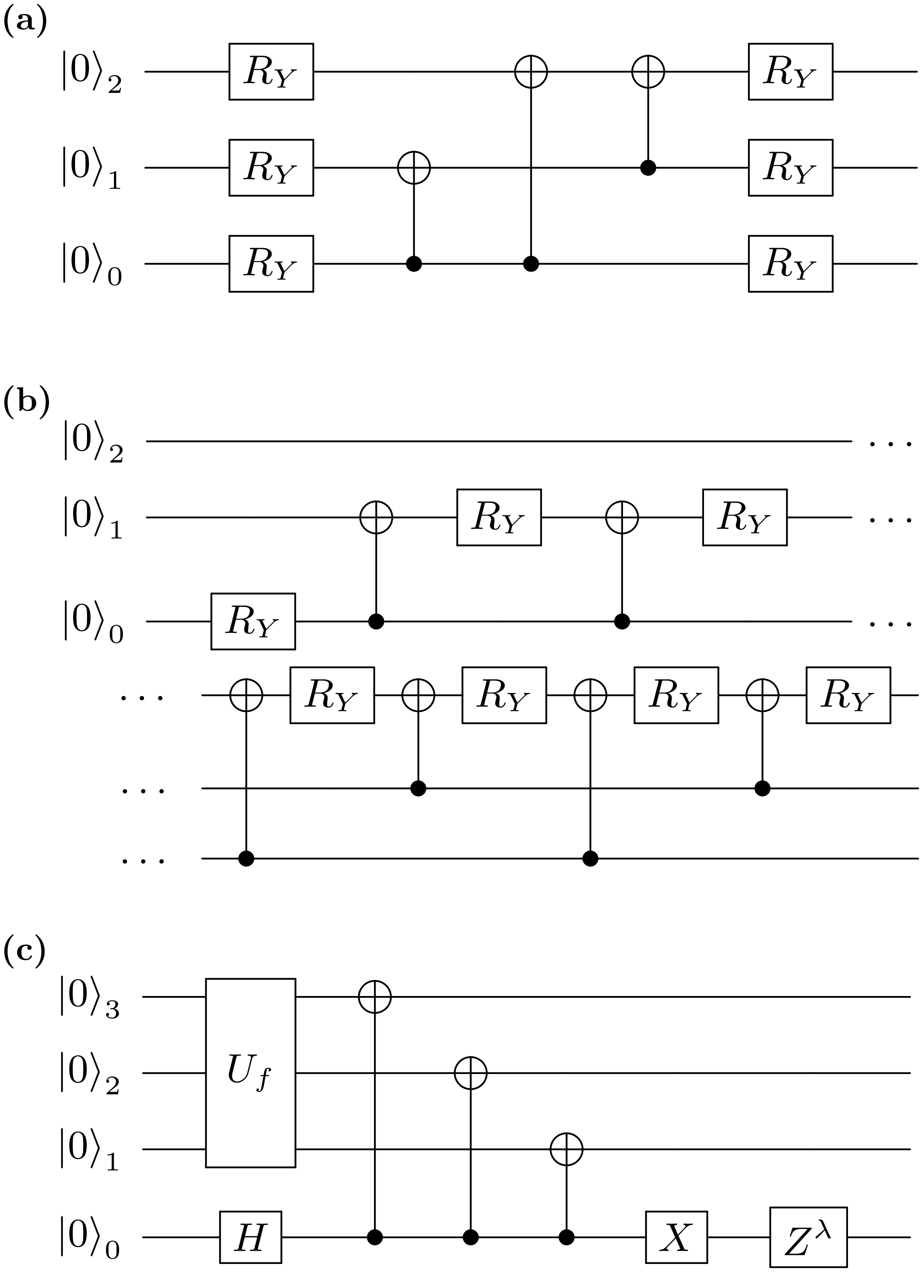


Figure 4.4: Variational representation of one-dimensional smooth functions. (a) R_Y ansatz of depth one, with full entanglement over three qubits. (b) ZGR ansatz to represent a function $f(x)$ with three qubits. (c) Ansatz to represent an (anti)symmetric function $g(x) = (\operatorname{sgn} x)^\lambda f(|x|)$, where U_f encodes $f(x > 0)$.

4.3.2 Zalka-Grover-Rudolph (ZGR) ansatz

The proposals of Zalka [47] and Grover and Rudolph [48] to discretize non-negative probability distributions in a quantum register motivate a better-suited ansatz for the representation of functions, exact up to the discretization. In the original proposal, conditional rotations of the least significant qubits based on the state of all previous qubits approximate such probability distributions,

$$\left|f_{\boldsymbol{\theta}}^{(n)}\right\rangle = \prod_{i=n-1}^{1} \prod_{z_i=0}^{2^{i-1}-1} \exp[i\theta_i(z_i)\sigma_i^y \left|z\right\rangle\!\left\langle z\right|)] \left|0\right\rangle^{\otimes n}, \tag{4.18}$$

where z_i is an integer constructed from the i first qubits $z = s_0 s_1 \cdots s_{i-1}$, and $\theta_i(z_i)$ is a set of angles chosen to represent the given function. This process constructively obtains the desired state, where the state of a qubit with low significance i is determined by controlled rotations defined by the more significant qubits $j \in \{0, 1, \ldots, i-1\}$.

These ideas lead to the Zalka-Grover-Rudolph (ZGR) ansatz, where these conditional rotations are combinations of σ^y rotations and CNOT gates (see Figure 4.4(b)). A slightly more efficient version, where a similar number of angles parametrizes rotations, is given by

$$\left|f_{\boldsymbol{\theta}}^{(n)}\right\rangle = \prod_{i=n-1}^{1} \prod_{z_i=0}^{2^{i-1}-1} e^{i\theta_{iz_i}\sigma^y} \mathrm{CNOT}_{z_i \text{ XOR } (z_i-1),i} e^{i\theta_{00}\sigma_0^y} \left|0\right\rangle^{\otimes n}. \tag{4.19}$$

The most significant bit active in z_i XOR $(z_i - 1)$ acts as control qubit for the CNOT gate with target qubit i. This ansatz has an exponential scaling of the number of parameters with the number of qubits, but still constitutes a valuable choice as (i) it is more accurate than the R_Y ansatz since it is tailored to represent highly differentiable smooth functions, (ii) in practice it gets incredibly close to the exact construct, but does not require additional NOT gates, and (iii) it illustrates the required ordering of conditional operations, from high to low significant qubits.

4.3.3 Symmetrization of variational circuits

The in-circuit symmetrization of the function significantly reduces the parameter scaling of the previous ansätze. This technique enables the construction of the negative part of the function—$f(x)$, $x < 0$— by reflecting the function encoded for $x > 0$

Given a function with reflection symmetry, $f(x) = (-1)^\lambda f(-x)$, with $\lambda = 0, 1$, and a symmetric discretization of the space

$$x_s^{(n)} = -\frac{L_x}{2} + \left(s + \frac{1}{2}\right) \Delta x, \tag{4.20}$$

its quantum circuit encoding satisfies

$$\langle 1 s_1 \dots s_{n-1} | f^\lambda \rangle = (-1)^\lambda \langle 0 \bar{s}_1 \dots \bar{s}_{n-1} | f^\lambda \rangle. \tag{4.21}$$

Equation (4.21) shows a relation between positive coordinates $x_s > 0$ and negative coordinates $x_s < 0$ that can be embedded into the variational circuit as shown in Figure 4.4(c). First, a Hadamard gate acts on the most significant qubit, constructing a quantum superposition where the most significant qubit encodes the values of the function for $x_s > 0$ ($s_0 = 1$) and $x_s < 0$ ($s_0 = 0$). Then, an ansatz encodes the positive values coordinates in the $n - 1$ least significant qubits. Finally, a set of CNOT gates reverses the function's orientation in the negative coordinates (4.21) and sets the right sign for the encoded state. Using this circuit, it is possible to encode an n-qubit function with an $(n-1)$-qubit ansatz, thus halving the number of parameters.

There are other proposals of symmetrization methods, such as the one in Ref. [216]. In this work, a projector operator creates a symmetry-adapted VQE. However, the nonunitarity of the projection operator requires classical postprocessing. In this section's approach, the symmetrization is obtained by adding one qubit to the circuit and a small number of single- and two-qubit gates, constituting a fully coherent approach that does not require postprocessing.

4.3.4 Fourier interpolation quantum circuit

This interpolation circuit computes the Fourier interpolation (Section 2.3) of the discretized function represented as a quantum state $\left| f^{(n)} \right\rangle$. The integration of the Fourier interpolation circuit with the previous ansätze (Section 4.3) leads to up to doubly exponentially decaying representation error in the number of qubits of the approximation, allowing an increase in the number of points for the function representation without increasing the number of parameters for the optimization. This helps to avoid potential issues with trainability and reduces the cost of obtaining the result.

Figure 4.5(a) represents the quantum circuit to interpolate a function $|f^{(n)}\rangle$ encoded in an n-qubit quantum register. This circuit is the quantum version of Fourier interpolation described in section 2.3 and appendix A. The first step of the circuit is applying the QFT (2.18) on such register, leading to the Fourier space function $|\tilde{f}^{(n)}\rangle$. Then, the padding algorithm adds $2^{m+n}-2^n$ zeros on the momentum space function by first adding an m-qubit register, and hence $2^{m+n}-2^n$ points to the representation of the function. The operation U_{sym} correctly orders the zeros in the state, considering the anomalous encoding of the momenta (2.20). It consists of a series of CNOT gates with the most significant qubit of the original register as control and the new qubits of the auxiliary register as targets. This operation maps the original discretization with 2^n points to the intervals $s \in [0, 2^{n-1}) \cup [2^{n+m}-2^{n-1}, 2^{n+m})$, so the new points are added in the middle states. Finally, the inverse QFT acting on the whole $(n+m)$-qubit register transforms the function back to position space, resulting in the interpolation function codified in the circuit's amplitudes. This algorithm can be written as

$$\left|f^{(n+m)}\right\rangle = \hat{\mathcal{F}}^{-1}\hat{U}_{\text{sym}}\left(|0\rangle^{\otimes m} \otimes \hat{\mathcal{F}}\left|f^{(n)}\right\rangle\right) =: \hat{U}^{n,m}_{\text{int}}\left|f^{(n)}\right\rangle. \tag{4.22}$$

Quantum Fourier interpolation is exponentially more efficient than its classical version. From the point of view of operations, the QFT uses $O((n+m)^2)$ quantum gates, which is exponentially less than the $O((n+m)2^{n+m})$ operations required by the fast Fourier transform. Incidentally, this advantage comes from the exponential compression provided by the quantum register, where a vector that takes N elements in a classical register requires only $\log_2(N)$ qubits to be represented.

Fourier interpolation also extends to momentum space as

$$\tilde{f}(p) \propto \sum_{s=0}^{2^n-1} e^{ipx_s}\langle s|f^{(n)}\rangle,\ p \in [-L_p/2, L_p/2). \tag{4.23}$$

The operation applies the QFT on the registers encoding the original n-qubit function and the m auxiliary qubits (Figure 4.5(b)) as

$$\left|\tilde{f}^{(n+m)}\right\rangle = \hat{\mathcal{F}}_{n+m}\left[|0\rangle^{\otimes m} \otimes \left|f^{(n)}\right\rangle\right]. \tag{4.24}$$

In this case, the bounds for the momentum space remain unchanged, while the position space grid is enlarged from $x \in [a,b)$ to $x \in [a, a+2^m L_x)$, setting all those extra points to the same value that the function takes at the boundary.

The QFT circuit is the basis of quantum Fourier interpolation. The circuit is a combination of Hadamard gates and two-qubit conditional

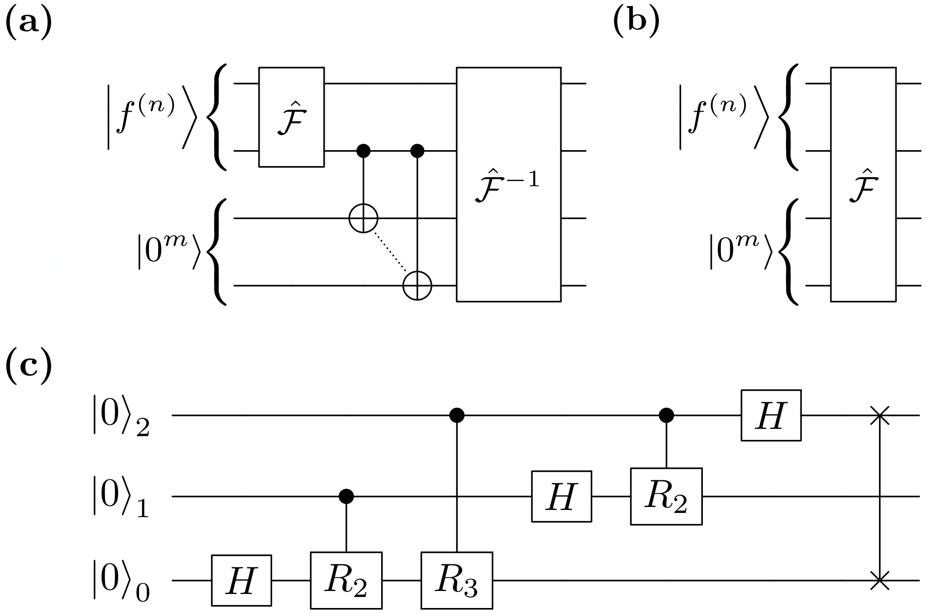


Figure 4.5: Quantum Fourier circuits. (a) Algorithm to recreate a finer interpolation in position space $|f^{(n+m)}\rangle$ adding m qubits to a previous discretization $\left|f^{(n)}\right\rangle$. (b) Algorithm to recreate a momentum space discretization with $n+m$ qubits from $\left|f^{(n)}\right\rangle$. (c) 3-qubit QFT, where the control phase rotation CR_k has a rotation angle $\theta_k = 2\pi/2^k$.

rotations (Figure 4.5(c))

$$\begin{aligned}\mathrm{QFT}(|s_0 s_1 \ldots s_{n-1}\rangle) = \frac{1}{\sqrt{2^n}} &\left(|0\rangle + e^{2\pi i\,[0.s_{n-1}]}|1\rangle\right) \\ &\otimes \left(|0\rangle + e^{2\pi i\,[0.s_{n-2}s_{n-1}]}|1\rangle\right) \\ &\otimes \cdots \otimes \left(|0\rangle + e^{2\pi i\,[0.s_0 s_1 \ldots s_{n-1}]}|1\rangle\right).\end{aligned} \tag{4.25}$$

The previous expression uses the fractional binary notation $[0.s_0 \ldots s_{n-1}] = \sum_{k=0}^{n-1} s_k 2^{-k-1}$.

4.4 Variational quantum PDE solver

As we showed in Ref. [46], the quantum register representation in chapter 2 and the variational circuit encodings from section 4.3 may be combined with conventional VQE algorithms to implement a solver for static

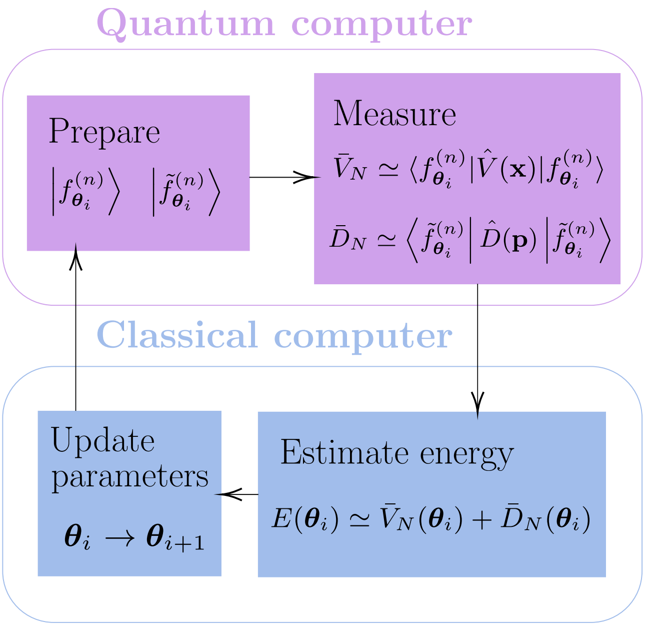

Quantum computer
Prepare
$\left|f^{(n)}_{\boldsymbol{\theta}_i}\right\rangle$ $\left|\tilde{f}^{(n)}_{\boldsymbol{\theta}_i}\right\rangle$
Measure
$\bar{V}_N \simeq \langle f^{(n)}_{\boldsymbol{\theta}_i}|\hat{V}(\mathbf{x})|f^{(n)}_{\boldsymbol{\theta}_i}\rangle$
$\bar{D}_N \simeq \left\langle \tilde{f}^{(n)}_{\boldsymbol{\theta}_i}\right|\hat{D}(\mathbf{p})\left|\tilde{f}^{(n)}_{\boldsymbol{\theta}_i}\right\rangle$
Classical computer
Update parameters
$\boldsymbol{\theta}_i \rightarrow \boldsymbol{\theta}_{i+1}$
Estimate energy
$E(\boldsymbol{\theta}_i) \simeq \bar{V}_N(\boldsymbol{\theta}_i) + \bar{D}_N(\boldsymbol{\theta}_i)$

Figure 4.6: Variational quantum PDE solver. A quantum computer initializes two quantum circuits in the states $\left|f^{(n)}_{\boldsymbol{\theta}_i}\right\rangle$ and $\left|\tilde{f}^{(n)}_{\boldsymbol{\theta}_i}\right\rangle$. The values of the measurements from the quantum computer estimate the expectation values of the $\hat{V}(\mathbf{x})$ and $\hat{D}(\mathbf{p})$, thereby approximating the energy functional $E[\boldsymbol{\theta}]$. A classical computer uses these estimates to iteratively update the parameters of the variational quantum circuit until convergence.

Schrödinger-type Hamiltonian PDEs (4.1). Rewriting the Hamiltonian PDE (4.1) as a lower-bounded Hamiltonian operator

$$H = D(-i\nabla) + V(\mathbf{x}) \geqslant E_0, \tag{4.26}$$

leads to a variational representation of the problem, where E_0 is the minimum energy, i.e., the ground state energy. This reformulation allows the transformation of solving a PDE into a problem that admits a variational formalism, being reduced to seeking the ground state E_0. The result of this variational search is equivalent to the solution of the diagonalization problem of H to find the minimum eigenvalue. While the variational quantum PDE solver is suitable for the multidimensional case, we will restrict to one-dimensional PDEs for the sake of simplicity.

The general scheme of the algorithm is depicted in Figure 4.6 and it comprises the following elements: (i) a map between states $\left|f^{(n)}\right\rangle$

of a quantum register with n qubits and bandwidth-limited continuous functions $f(x)$; (ii) the realization that given $\left|f^{(n)}\right\rangle$, the QFT creates a state $\left|\tilde{f}^{(n)}\right\rangle = \hat{\mathcal{F}}\left|f^{(n)}\right\rangle$ that encodes the classical Fourier transform $\tilde{f} = \mathcal{F}f$ of the encoded function $f(x)$; (iii) a quantum algorithm that uses $\left|f^{(n)}\right\rangle$ and $\left|\tilde{f}^{(n)}\right\rangle$ and suitable representations of position and momentum operators $\hat{x}$ and $\hat{p}$, to estimate the energy functional $E[f] = (f, Hf)$ with polynomial resources as

$$E[f] = \left\langle \tilde{f}^{(n)}\right| \hat{D}(p) \left|\tilde{f}^{(n)}\right\rangle + \left\langle f^{(n)}\right| \hat{V}(x) \left|f^{(n)}\right\rangle; \tag{4.27}$$

(iv) a variational quantum circuit $W(\boldsymbol{\theta})$, with $\boldsymbol{\theta} \in \mathbb{R}^k$, which creates parameterized states of a quantum register with n qubits $\left|f_{\boldsymbol{\theta}}^{(n)}\right\rangle = W(\boldsymbol{\theta})\left|00\ldots0\right\rangle$ (Section 4.3); and (v) a classical optimization algorithm, such as constrained optimization by linear approximation (COBYLA), simultaneous perturbation stochastic approximation (SPSA), or adaptative moment estimation (Adam), that finds the parameters $\boldsymbol{\theta}$ that minimize the cost function $E(\boldsymbol{\theta})$. Ingredients (i) and (iv) map the variational ansatz $W(\boldsymbol{\theta})$ to the set of bandwidth limited functions $f_{\boldsymbol{\theta}}(x)$ and their energies $E[f_{\boldsymbol{\theta}}] := E(\boldsymbol{\theta})$ for the given PDE. Quantum algorithms (ii) and (iii) estimate the energy $E(\boldsymbol{\theta})$ for each set of parameters. Finally, the classical algorithm (v) interacts with the quantum algorithms (i)-(iv) to obtain the best approximation $f_{\mathrm{argmin}_{\boldsymbol{\theta}} E(\boldsymbol{\theta})}$ of a solution f_0 to Eq. (4.1),

$$f_0 \leq \mathrm{argmin}_f \langle f|\hat{H}|f\rangle \leq f_{\mathrm{argmin}_{\boldsymbol{\theta}} E(\boldsymbol{\theta})}. \tag{4.28}$$

A quantum representation $\hat{H}^{(n)} = \hat{D}\left(p^{(n)}\right) + \hat{V}\left(x^{(n)}\right)$ of the Hamiltonian differential operator (4.2) enables solving the Hamiltonian PDE (4.1) using the map between the states of a quantum register and the space of bandwidth-limited differentiable functions. As introduced in section 2.2, the position $V(\hat{x})$ and differential $D(\hat{p})$ operators are given by

$$\hat{V}\left(x^{(n)}\right) := \sum_s V\left(x_s^{(n)}\right)|s\rangle\langle s|, \tag{4.29}$$

$$\hat{D}\left(p^{(n)}\right) := \hat{\mathcal{F}}^{-1}\sum_s D\left(p_s^{(n)}\right)|s\rangle\langle s|\,\hat{\mathcal{F}}, \tag{4.30}$$

using the spectral method to approximate the differential operator. This reformulation of the problem allows transforming the search for the stationary solutions of the PDE (4.1) into the search for the eigenstates of the $\hat{H}^{(n)}$ operator, admitting a variational formalism.

Then, a dense variational family of unitary operators $W(\boldsymbol{\theta})$ acting on n qubits defines a continuous family of trial states $\left|f_{\boldsymbol{\theta}}^{(n)}\right\rangle := W(\boldsymbol{\theta})\left|0\right\rangle^{\otimes n}$ (Section 4.3), which a classical optimizer updates by the minimization of a cost function

$$E(\boldsymbol{\theta}) := \left\langle f_{\boldsymbol{\theta}}^{(n)}\right| \hat{H}^{(n)} \left|f_{\boldsymbol{\theta}}^{(n)}\right\rangle . \tag{4.31}$$

The optimum parameters $\boldsymbol{\theta}_0 :=$ argmin $E(\boldsymbol{\theta})$ lead to the continuous function associated with the quantum state $\left|f_{\boldsymbol{\theta}_0}^{(n)}\right\rangle$ that approximates the solution.

The variational quantum PDE solver obtains these optimum parameters by combining a classical optimizer to minimize $E(\boldsymbol{\theta})$ with a quantum computer or simulator that provides an approximate $E(\boldsymbol{\theta})$. Thus, in practice, for a finite number of measurements, it is unfeasible to obtain $E(\boldsymbol{\theta})$, but a randomized estimator $\bar{E}_M$ that results from a finite set M of measurements. This estimator is the "energy" obtained as the sum of the expectation values of the potential $\bar{V}_M$ and differential $\bar{D}_M$ operators,

$$E(\boldsymbol{\theta}) := \bar{E}_M + O\left(\frac{1}{\sqrt{M}}\right) \quad \text{with } \bar{E}_M := \bar{V}_M + \bar{D}_M. \tag{4.32}$$

The expectation values are computed separately in two quantum circuits encoding $\left|f_{\boldsymbol{\theta}}^{(n)}\right\rangle$ and $\left|\tilde{f}_{\boldsymbol{\theta}}^{(n)}\right\rangle := \hat{\mathcal{F}}\left|f_{\boldsymbol{\theta}}^{(n)}\right\rangle$ and measuring those states in the computational basis. This result is probabilistic, with errors that decrease with the number of measurements according to the law of large numbers, $\Delta\hat{V}/\sqrt{M}$ and $\Delta\hat{D}/\sqrt{M}$ for the potential and differential operators, respectively. Better algorithms could be constructed by using amplitude estimation over approximate implementations of the unitary operator $\exp(-i\hat{H}\Delta t)$, but this requires an infrastructure and a precision of gates that is not presently available.

Because of the association of continuous, infinitely differentiable functions $f(x)$ with quantum states $\left|f^{(n)}\right\rangle$, Eq. (4.31) gives us the exact value of the functional $E[f^{(n)}] = (f^{(n)}, \hat{H}f^{(n)})$. This means that the variational algorithm finds a strict upper bound on the exact eigenvalue E_0 and the errors of the method can only be due to the expressive power of the variational ansatz and the capacity of $f^{(n)}(x_s)$ to approximate the solution $f(x)$.

Similarly to other VQAs, the variational quantum PDE solver is also suitable for approximating excited states. A possible approach is to modify the cost function to minimize the overlap with the previously found eigenstates [202]. Finally, Ref. [217] implements a related method for solving the harmonic oscillator equation and engineering Gaussian states in quantum circuits.

4.5 Numerical solution of Hamiltonian PDEs

In this section, we evaluate the effectiveness of the variational quantum PDE solver for static equations discussed in section 4.4. We focus on a subset of problems—specifically, the superconducting circuits in section 4.1—to study its performance and suitability for NISQ devices. Our study begins with an analysis under ideal conditions (Section 4.5.2), i.e., error-free quantum circuits, but considering limitations arising from the finite number of measurements. We then extend the study to real-world scenarios by examining the algorithm's behavior under noise sources present in current devices (Section 4.5.3).

The figures of merit in section 4.5.1 assess the method's accuracy in computing the ground state and energy of the quantum harmonic oscillator, transmon, and flux qubit PDEs. Although this approach could be extended to excited states, we focus on the ground state as it provides the simplest and most direct benchmark. The variational ansätze in section 4.3—the ZGR and R_Y ansätze, with depths 1, and 1 and 2, respectively—encode the problem for quantum circuits from 2 to 6 qubits. The code for the simulations is based on IBM's Python library QISKIT [218], using three classical optimizers for the minimization: COBYLA, a gradient-free method; SPSA, a stochastic optimizer with a numerical gradient; and Adam, which we combine with an analytic estimate of the gradient [199, 200].

4.5.1 Figures of merit

This benchmark studies the algorithms' performance based on the accuracy of the energy evaluation and the determination of the solution. More figures of merit, such as time or cost function evaluations, could be considered. However, the goal is to test both the trainability and accuracy of the proposed techniques, constituting a proof of concept of the theoretical bounds for precision (Section 2.5) and the capability of quantum computers to tackle these problems. Thus, the proposed metrics are sufficient for this purpose, and improvements in the application to minimize the time and cost of the algorithms can lead to further studies.

First, the fidelity $F[\psi_1, \psi_2] := |\langle\psi_1|\psi_2\rangle|^2$ between two quantum states $|\psi_1\rangle, |\psi_2\rangle$—which is restricted to the values $F \in [0, 1]$ since quantum states are normalized—determines the error in the approximation of the solution. The first option is to consider the fidelity between the variational states produced by the optimization $W(\boldsymbol{\theta}_{\text{opt}})\,|0\rangle$, with the discretized solution using the same number of qubits $\left|f^{(n)}\right\rangle$,

$$F^{(n)} := \left|\left\langle f^{(n)}\middle|W(\boldsymbol{\theta}_{\text{opt}})\middle|0\right\rangle^{\otimes n}\right|^2. \tag{4.33}$$

Even though this equation leads to an arbitrarily small error in the approximation of the discretized solution, it gives no information regarding the continuous function that is the true solution to the problem, $f(x)$. In order to provide a more uniform metric for all grids, we introduce an estimate of the L_2 error of the function on the continuum limit, associating to each discretization a continuum limit via Fourier interpolation. We call this figure of merit the continuous fidelity. For a sufficiently large number of points for its discretization, one must consider $\lim_{m\to\infty} f^{(n+m)} \approx f(x)$, and compare it with the Fourier interpolated (4.22) approximation of the resulting approximation,

$$F^{\infty} := \lim_{m\to\infty}\left|\left\langle f^{(n+m)}\middle|U_{\text{int}}^{n,m}W(\boldsymbol{\theta}_{\text{opt}})\middle|0\right\rangle^{\otimes n}\right|^2. \tag{4.34}$$

This determines the extent to which information from the continuous exact solution can be recovered from its discretized approximation resulting from the optimization.

Figure 4.7 shows the infidelities $1 - F^{(n)}$ and $1 - F^{\infty}$ obtained with the ZGR ansatz and the best optimization method, as well as the best numerical approximation using a finite grid with 2^n points. The fidelity computed over the discretized state overestimates our knowledge of the solution, with $F^{(n)} \geqslant F^{\infty}$, especially for small grids, in which the number of initial points is too small for the interpolation to obtain a high fidelity. Since we implicitly work with continuous functions unambiguously defined over the grid points, the continuous fidelity offers a better representation of the information recovered. This is because Fourier interpolation accurately approximates the values outside the original grid as it is mathematically consistent with using the Fourier transform for approximating the derivatives. Thus, the simulations' plots will depict the median of the infidelities $1 - F^{\infty}$ over 100 repetitions of each simulation with different initial states or trajectories for our later figures, and the error bars will be the standard deviation around the mean. Moreover, F^{∞} uses $m = 12 - n$ extra qubits, which already gives a good converged measure.

We choose the relative error ε in the computation of the energy to gauge how well the method estimates the properties of the solution to

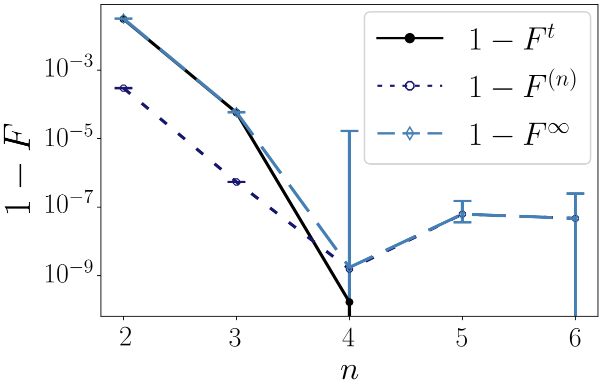


Figure 4.7: Comparison of the infidelity figures of merit for the harmonic oscillator for the numerical limit. The figure depicts the median and the standard deviation around the mean of 100 repetitions of the simulations using the ZGR ansatz. The theoretical infidelity $1 - F^t$ is the infidelity of the theoretical 2^n-point wavefunction interpolated up to 2^{12} points and the theoretical 2^{12}-point wavefunction. The n-qubit infidelity $1 - F^{(n)}$ (4.33) is the infidelity of the n-qubit function obtained from the optimization and the 2^n-point theoretical function. The continuous infidelity $1 - F^{\infty}$ (4.34) is the infidelity of the n-qubit function interpolated up to 2^{12} points and the 2^{12}-point theoretical function

the PDE. This error is given by

$$\varepsilon = \left| \frac{E_{t_n} - E_{\text{opt}}}{E_1 - E_0} \right|, \tag{4.35}$$

where E_{t_n} is the theoretical energy obtained over a grid with 2^n points, E_{opt} is the optimal energy derived by the algorithm, and $E_1 - E_0$ is the energy difference between the lowest and first excited solutions of the PDE (4.1). This figure of merit is dimensionless and fits the natural energy scales of the problem, allowing us to compare different PDEs. In idealized applications, ε and $1 - F^{\infty}$ should be proportional to each other, but this is not always true in real-world quantum computers, as the simulations demonstrate.

4.5.2 Ideal simulations

This section studies the performance of the variational quantum PDE solver for approximating the solution of three Hamiltonian PDEs—quantum harmonic oscillator, transmon, and flux qubit—in an ideal quantum circuit with no errors other than the quantum fluctuations in the measurement.

Harmonic oscillator

First, the algorithm tackles the most straightforward problem, the one-dimensional quantum harmonic oscillator model from section 4.1. In this case, for a position space symmetrically defined around the origin, the Nyquist-Shannon theorem determines the optimum interval for each n, $L_x \approx \sqrt{2\pi 2^n}$, maximizing the accuracy in both position and momentum space.

Figure 4.8 shows the simulation results for the three ansätze and optimizers considered. Figures 4.8(a)-(c) depict the continuous infidelity[1] $1 - F^{\infty}$ (4.34) of the optimal state, including the lowest infidelity $1 - F_t$ obtained with the best approximation on the same grid, that we use as a reference. Figures 4.8(d)-(f) show the relative error in the estimation of the energy (4.35) for those solutions. We conclude that the best optimizer is ADAM, closely followed by SPSA. Both methods overcome COBYLA due to their tolerance to the intrinsic uncertainty in estimating the energy. However, Adam's accuracy is higher than that of SPSA, even for the same ansatz. This is because SPSA uses a stochastic estimate of the cost function's gradient with errors amplified by small denominators, while our Adam implementation uses an analytic estimate based on the parameter-shift rule [199, 200].

Figure 4.9 shows the best result according to the infidelity for each ansatz, concluding that the ZGR ansatz is the best variational ansatz for this problem with a limited number of measurements. This ansatz is designed to represent continuous functions [47, 48], where every rotation builds on the previous ones in a smooth, easily differentiable fashion, with no loss of information. The ZGR ansatz limits the Hilbert space for the parameter search and hence targets quantum states more suitable for encoding this type of function. The R_Y ansatz is a general purpose one, therefore ideal for many applications, but more complicated to optimize for a concrete implementation. In this ansatz, the search space is less restricted, and the influence of different qubits and layers is more inefficiently transported by the layers of entangling unitaries, resulting in the vanishing of gradients [215]. Moreover, as the number of qubits increases, the accuracy decreases. This occurs due to the increasing number of parameters with the number of qubits, decreasing the trainability

[1] We use the notation $F = F^{\infty}$.

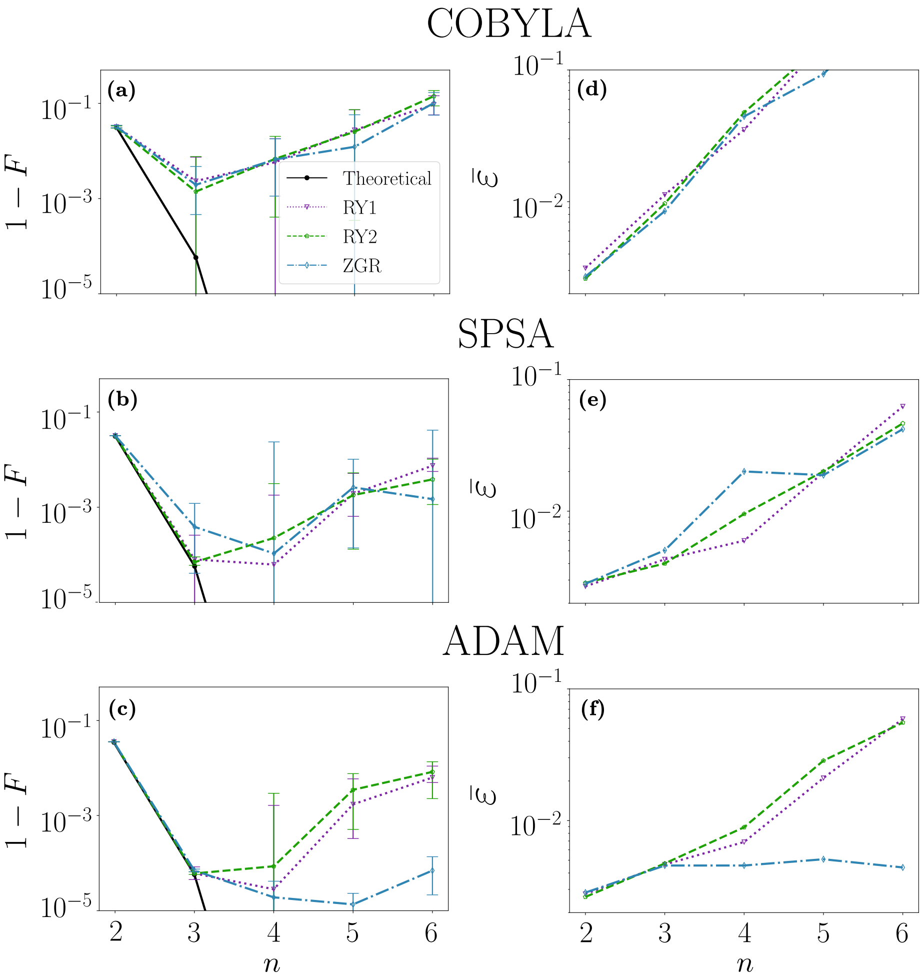


Figure 4.8: Results of the simulations for two, three, four, five, and six qubits with 8192 evaluations for the harmonic oscillator. The simulations compare the ZGR and the R_Y ansätze with depths 1 (RY1) and 2 (RY2) and the COBYLA, SPSA, and ADAM optimizers. (a) Continuous infidelity (4.34) with $n + m = 12$. (b) Rescaled energy $\bar{\varepsilon}$ (4.35).

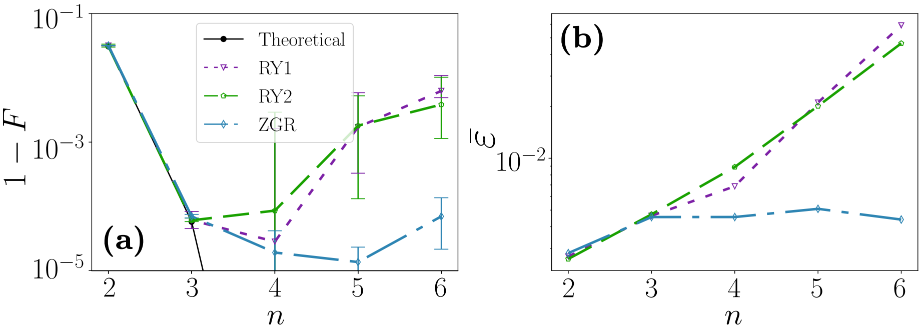


Figure 4.9: Lowest infidelity results of the simulations for two, three, four, five, and six qubits with 8192 evaluations for the harmonic oscillator. The figure represents the best optimizer for each ansatz—ZGR and the R_Y ansätze with depths 1 (RY1) and 2 (RY2). (a) Continuous infidelity (4.34) with $n+m=12$. (b) Rescaled energy $\bar{\varepsilon}$ (4.35).

of the ansatz due to the barren plateaus and the trapping in local minima, showing a more chaotic nature for the R_Y ansatz.

Figure 4.9 shows that the number of measurements limits the precision of the energy estimation. Even a limited number of measurements achieves low infidelities, in the range $10^{-3}-10^{-5}$, below what is expected from the statistical uncertainty in evaluating the cost function. Figure 4.10(b) shows this behavior. It depicts one of the optimization trajectories of the Adam method for the ZGR ansatz and $n=3$. A dashed line represents the evaluation of the cost function as returned by the simulation $\bar{E}_M(\boldsymbol{\theta})$, surrounded by a colored band estimating the statistical uncertainty for $M=8192$ measurements. Without any uncertainty, this trajectory and error band must be compared to the actual energy computed for the same parameters $E(\boldsymbol{\theta})$. The associated relative error $\bar{\varepsilon}\simeq 5\times 10^{-3}$ is one order of magnitude below the statistical uncertainty, illustrating the power of stochastic optimization. This optimization also obtains a high fidelity in the approximation of the solution (Figure 4.10(a)). Even with a low number of qubits, i.e., $n=3$ corresponding to an 8-point grid, it is possible to obtain a good approximation of the continuous exact solution using interpolation. Since the interpolation interval has non-strictly periodic conditions, small Gibbs oscillations appear in the boundary.

The error in estimating the energy of the Hamiltonian PDE (4.1) decreases with the number of evaluations following the law of large numbers. Decreasing this error leads to a more precise computation of the

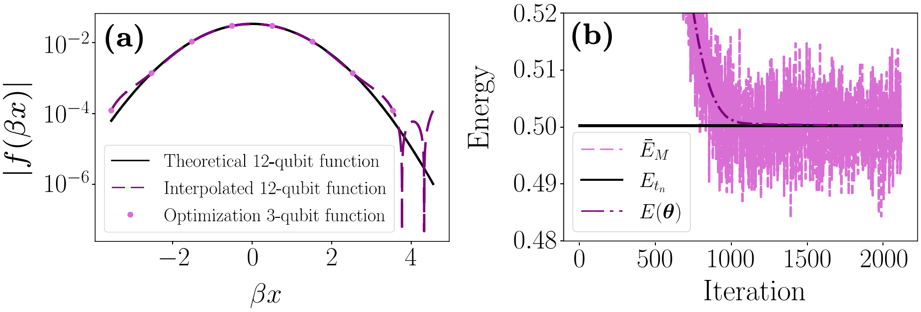


Figure 4.10: Simulation run for the Adam optimizer and the ZGR ansatz for three qubits for the harmonic oscillator. (a) Absolute value of the theoretical and optimization wavefunctions (βx is a dimensionless coordinate where $\beta = \sqrt{m\omega/\hbar}$). (b) Value of the energy for each iteration.

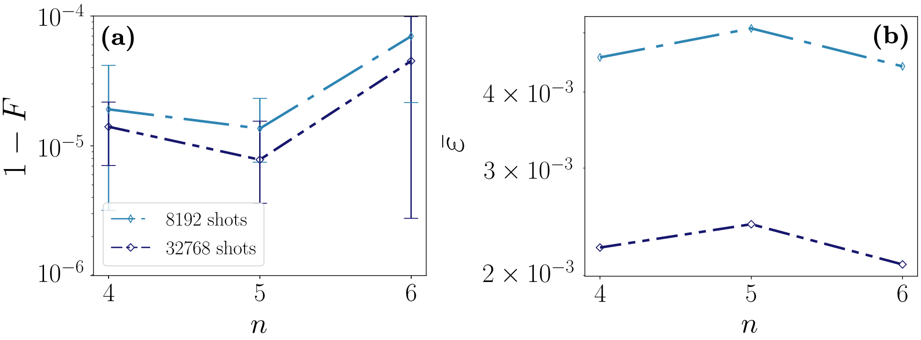


Figure 4.11: Results for the harmonic oscillator simulation with varying evaluations. The simulation uses the Adam optimizer and the ZGR ansatz for 8192 and 32768 evaluations. (a) Continuous infidelity (4.34) with $n+m=12$. (b) Rescaled energy $\bar{\varepsilon}$ (4.35).

cost function, positively affecting the algorithm's performance. In Figure 4.11, the number of evaluations increases, effectively showing that it limits the precision of the computation of the expectation value (Figure 4.11(b)). An increase in the number of evaluations by a factor of 4 consequently decreases $\bar{\varepsilon}$ by a factor 2 and also increases the quality of the solution, as it can be seen in Figure. 4.11(a).

The number of evaluations limits the performance of the variational PDE solver due to the limitation in the cost function estimation. Table B.1 in appendix B displays the exact numerical results of the continuous infidelity for the three ansätze and optimizers considered, including the numerical limit results combining the exact non-probabilistic statevector simulator with the L-BFGS-B optimizer. The results in the numerical limit are equivalent to an infinite number of evaluations, removing this limitation and showing the true potential of our proposal. While the minimum infidelity is $O(10^{-5})$ with a finite number of measurements, it can decrease up to 5 orders of magnitude with a sufficiently large number of evaluations. This means that the quantum Fourier analysis techniques can achieve high precision in the representation of numerical analysis problems in a quantum register, and what limits them is the finite number of measurements.

Transmon qubit

Harmonic potentials constitute a known problem suitable for testing the performance of quantum numerical analysis techniques. Nevertheless, challenging PDEs, such as those presented by superconducting qubit Hamiltonians, provide a more rigorous test for the proposed method. The previous analysis can be extended to such PDEs, such as the transmon qubit equation 4.7—with inductive-to-capacitive energy ratio $E_J/E_C = 50$—, showing the main results in Figure 4.12 and Table B.2. These results confirm the behavior of the methods shown by the harmonic oscillator. The combination of the analytic Adam optimizer and the ZGR ansatz leads to the best result due to the interplay of an analytic estimation of the gradient and a problem-adapted ansatz. Again, the local trapping and statistical uncertainty limit the precision, with no simulation achieving the theoretical limit as the number of qubits increases.

The transmon qubit's fidelity is smaller than the one of the harmonic oscillator. The analytical ground state of the transmon is the zeroth-order solution of the Mathieu equation (4.8), which is more complicated to reproduce than the Gaussian function of the ground state of the harmonic oscillator (4.6). This function, while periodic, does not strictly vanish on the boundaries of the definition interval. Moreover, we cannot maximize the accuracy with the definition of such an interval since the variable φ is limited to a 2π-interval by definition. Even in the case of numerical exact optimizations via L-BFGS-B, the infidelities are lower (Table B.2).

The transmon qubit is a problem of physical interest, motivated by studying a real quantum system with practical applications. Qubits are the building blocks of quantum computers, and it is an amenable idea to use a quantum computing algorithm to study the physics underlying its main components. We need to increase the number of evaluations to use this method to compute actual properties of qubits, such as the energies

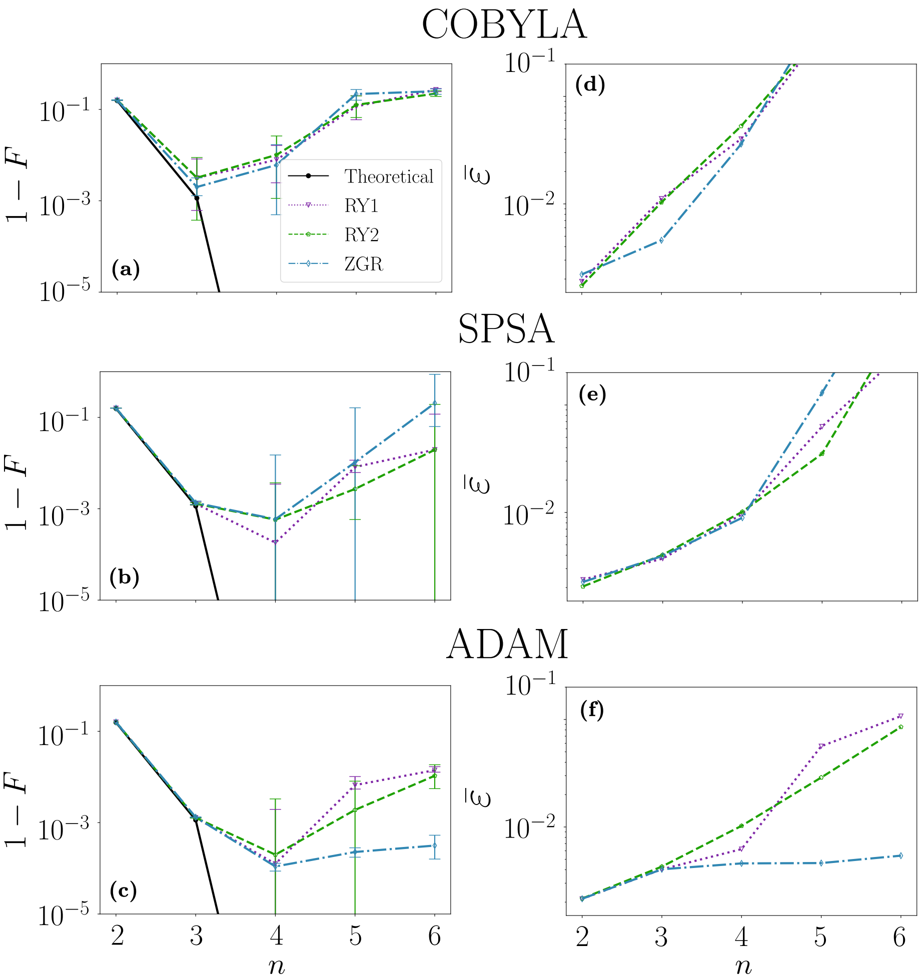


Figure 4.12: Results of the simulations for two, three, four, five, and six qubits with 8192 evaluations for the transmon qubit. The simulations compare the ZGR and the R_Y ansätze with depths 1 (RY1) and 2 (RY2) and the COBYLA, SPSA, and ADAM optimizers. (a) Continuous infidelity (4.34) with $n + m = 12$. (b) Rescaled energy $\bar{\varepsilon}$ (4.35).

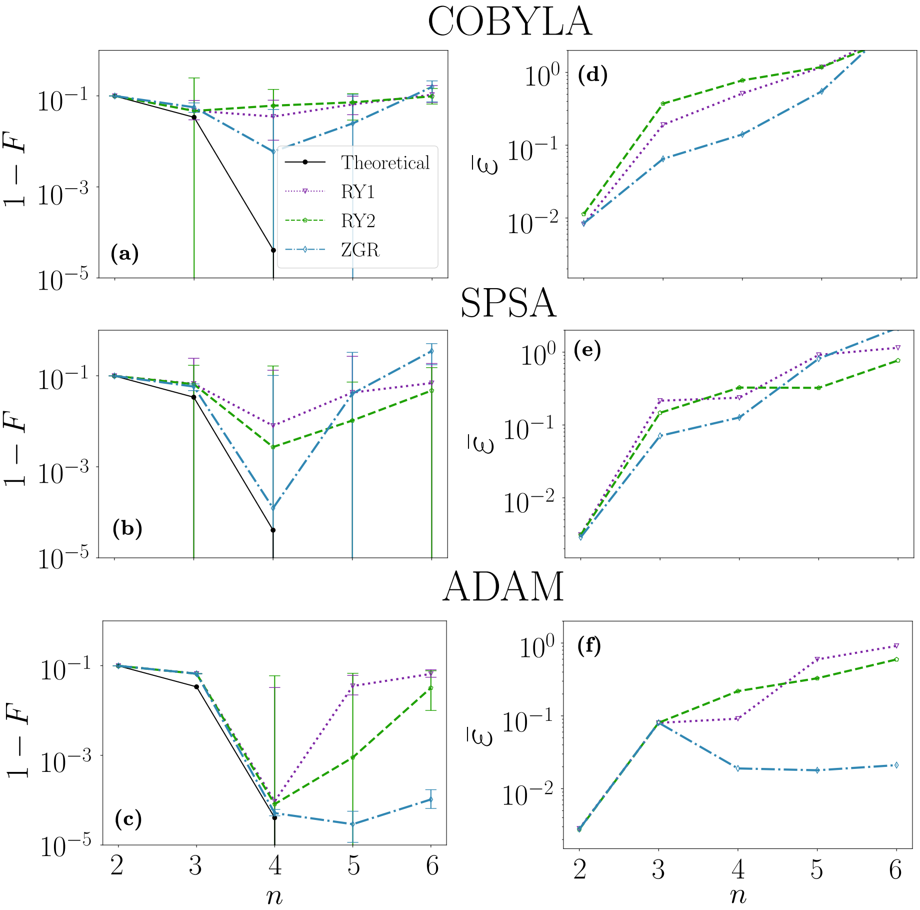


Figure 4.13: Results of the simulations for two, three, four, five, and six qubits with 8192 evaluations for the flux qubit. The simulations compare the ZGR and the R_Y ansätze with depths 1 (RY1) and 2 (RY2) and the COBYLA, SPSA, and ADAM optimizers. (a) Continuous infidelity (4.34) with $n+m=12$. (b) Rescaled energy $\bar{\varepsilon}$ (4.35).

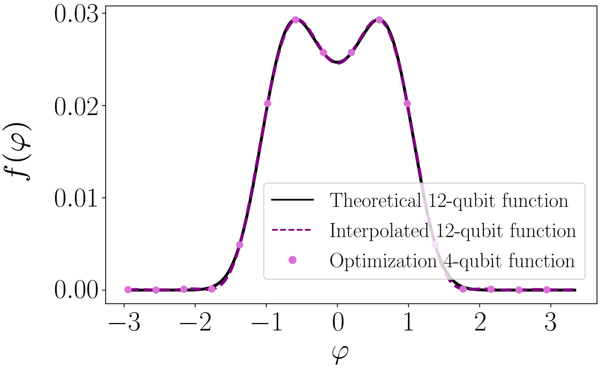


Figure 4.14: Absolute value of the ground state of the flux qubit. We show the theoretical eigenstate and the discretized and interpolated solution of our algorithm for the four-qubit RY1 ansatz using the L-BFGS-B optimizer.

and excitation probabilities. The relative errors obtained, $10^{-2} - 10^{-3}$, are compatible with what can be expected from 8192 shots. To achieve a relative error below 10^{-4} (a fraction of a megahertz), one would need to use about 100-10000 times more measurements in the final stages of the optimization. While this seems doable, it suggests the need to find better strategies for the energy evaluation or even the optimization itself.

Flux qubit

The harmonic oscillator and transmon are two equations that produce very similar wavefunctions. In particular, the transmon can be very well approximated by a Gaussian in the $E_J \gg E_C$ limit. This motivates us to explore a physical model, the flux qubit (4.15), which produces significantly different quantum states. In this case, the Hamiltonian PDE has no analytic solution, so the representation of the qubit on the charge basis approximates the ground state solution for the benchmark. For the simulations, we choose an inductive-to-capacitive energy ratio is $E_J/E_C = 50$, and a junction size $\alpha = 0.7$.

Figure 4.13 and Table B.3 show the results of the solution of the flux qubit equation (4.15). The best performance is obtained for the ZGR ansatz combined with the Adam optimizer, confirming the previous observations, even though the ground state of the flux qubit is not a Gaussian and cannot be approximated by one. Thus, the variational PDE solver with the quantum register Fourier representation of functions succeeds at obtaining different types of solutions, as long as they verify the conditions in section 4.4.

As shown in Figure 4.14, the double-well structure of the qubit's potential creates a state resembling the superposition of two Gaussians. The greater complexity of the flux qubit's ground state demands more qubits for an accurate representation. Still, with already four qubits, we obtain good results of $O(10^{-5})$ infidelity for the function representation. On the other hand, the relative errors for the energy of the qubit are higher, which can be explained by the fact that the ground and first excited state energies are comparatively closer, requiring a smaller absolute precision in evaluating the energy to be accurately approximated.

4.5.3 Application to NISQ devices

The study in section 4.5.2 is a noise-free proof of concept demonstration of our algorithms. It confirms the utility of Fourier analysis to achieve good accuracies with a limited number of parameters. However, the study is confined to an ideal model of a quantum computer, fault-tolerant and without errors in the gates.

VQAs, and hybrid quantum-classical algorithms in general, were designed to work in NISQ devices with faulty gate implementations and qubits with finite lifetimes. Still, even under these imperfect circumstances, variational constructs can be optimized to approach the ideal limits [21]. What follows is a more realistic study that takes into account the limitations of NISQ devices, combining the inaccuracies that may arise from a finite number of measurements with the imperfections in single- and two-qubit operations and also in measurements.

This study runs the numerical experiments using a simulator of a real quantum computer, the ibmq_santiago five-qubit quantum computer [218]. This simulator stores the noise model, coupling map, and basis gates of the quantum computers so that each computation is subject to identical conditions. The noise model includes the gate error probability and gate length for each basis gate and qubit, as well as the readout error probabilities and T_1 and T_2 relaxation times for each qubit. For this device, T_1 and T_2 are of the order of 100 μs, and the readout probability is $\sim 10^{-2}$. The gate error for single-qubit gates is of order $10^{-3} - 10^{-4}$ with a gate length of order 10 ns, while both the gate error and length are one order of magnitude greater for two-qubit gates [2]. Using a simulator of a quantum computer for the numerical experiments avoids depending on the reduced availability of quantum devices, for which the long waiting

[2] The results are for the ibmq_santiago calibration with date 17 March 2021.

times affect the reproducibility of the experiments. Quantum computers are calibrated daily, modifying the value of the noise parameters and, hence, the performance of the experiments. Numerical experiments fix these parameters and reproduce the noise effects accurately.

The choice of ansatz and optimizer can significantly reduce the effect of noise. The R_Y ansatz with depth 1 has the least number of gates and hence noise due to their application and decoherence. Combining this ansatz with an optimizer with a stochastic nature, such as the SPSA, is expected to have more noise resilience. Figure 4.15(a) compares the continuous infidelity in the noiseless and noisy simulations of the most straightforward problem, the 1D quantum harmonic oscillator. When measured against the ideal case, the fidelity of the noisy simulations decreases with the number of qubits due to the more significant number of gates, which introduce more errors and increase the effect of decoherence due to the longer circuit time. However, for a small number of qubits $n = 3$, obtaining a low infidelity of order $O(10^{-4})$ is still possible, recovering the exact solution from the noisy measurements.

Though the noisy optimization successfully reconstructs the continuous solution, the error in the estimation of the energy considerably increases (Figure 4.15(b)). This error is associated with the circuits that evaluate the energy in position and momentum space. The QISKIT quantum tomography toolbox gives the resources to confirm this hypothesis. As shown in Figure 4.15(c), the errors in the two circuits can be quite significant and are larger in momentum space because of the gates required for the QFT.

Despite the errors in these circuits, obtaining a good estimate of the solution suggests using error mitigation [219] to improve the quality of the energy approximation. Error mitigation techniques arise as valuable tools to reduce the effect of noise in current quantum computers since the complete removal of noise, i.e., quantum error correction [220], is still not feasible. One of the most used error mitigation techniques is zero-noise extrapolation [221], in which the value of the computation is extrapolated to the zero noise limit by increasing the noise associated with a certain parameter to have enough points to perform an extrapolation. Let us consider a simpler noise model dominated by thermal relaxation, where T_1 is a tunable variable. This model still uses the coupling map and basis gates of the ibmq_santiago five-qubit device. A quantum circuit encodes the optimal solution for three qubits to perform error mitigation. Then, it runs 100 repetitions of the simulation with 8192 evaluations each for different values of T_1. The mean energy resulting from these

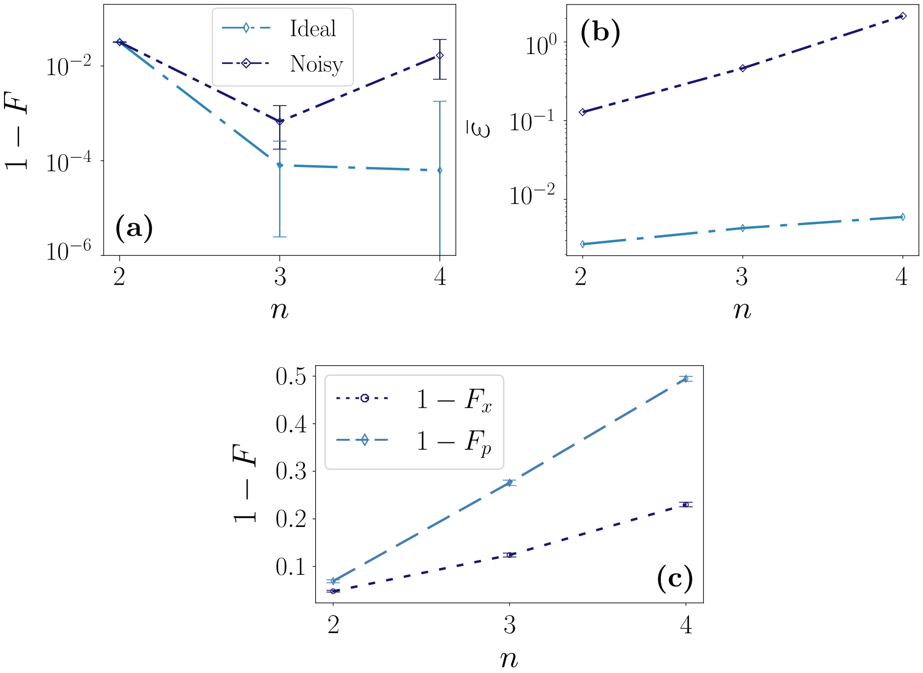


Figure 4.15: Ideal (noiseless) and noisy (ibmq_santiago noise model) optimization for the harmonic oscillator. The simulations use the SPSA optimizer and the RY1 ansatz with 8192 evaluations. (a) Continuous infidelity (4.34) with $n+m=12$. (b) Rescaled energy $\bar{\varepsilon}$ (4.35). (c) Circuit infidelity for the position $1-F_x$ and momentum $1-F_p$ circuits for 100 repetitions with 8192 evaluations each.

measurements computes the error ε (4.35). As Figure 4.16(a) shows, if $T_1 \geqslant 2.5\mu$s, the energy admits a Taylor expansion of fifth order

$$E(T_1) = E_0 + \sum_n \epsilon_n \frac{1}{T_1^n}. \tag{4.36}$$

To further improve the estimation of the energy, we use Richardson extrapolation [221], reaching an error of order $\varepsilon \sim 10^{-2}$ for state-of-the-art values of $T_1 \sim 50 - 100\ \mu s$.

The previous problem is a relatively simple model to test the noise resilience of the proposal. To push the limits of the method harder, let us apply this technique for the Hamiltonian PDE of the flux qubit (4.15). The energy spectrum has a smaller gap between the ground and first excited states than the harmonic oscillator and transmon qubits, and the solution is a superposition of two Gaussians on two separate wells.

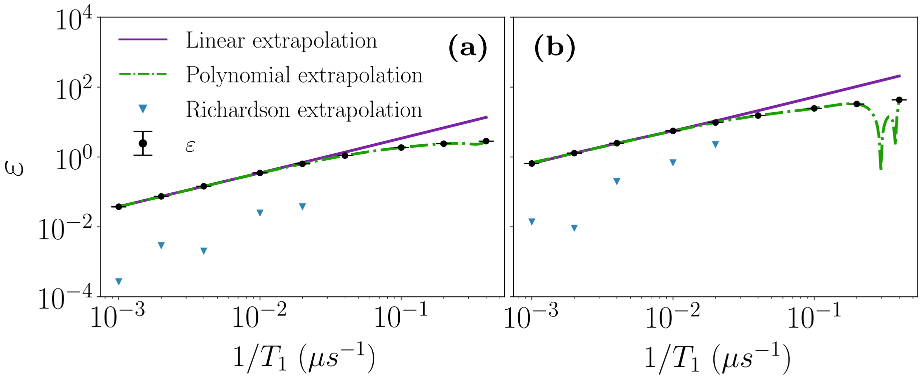


Figure 4.16: Zero-noise extrapolation results. The simulations are performed for the RY1 ansatz with thermal relaxation and 100 repetitions with 8192 evaluations for each simulation. (a) Harmonic oscillator (three qubits). (b) Flux qubit (four qubits).

This higher complexity means more points are needed to achieve a high-fidelity interpolation. In general, the necessity of more qubits leads to a higher error in estimating the energy, even under ideal circumstances (Figure 4.13), due to the larger number of parameters for the optimization. Therefore, error mitigation is expected to perform worse than for the harmonic oscillator.

Figure 4.16(b) shows the results of the noisy simulations with four qubits for the flux qubit. These simulations use $n = 4$; hence, the larger number of gates with their associated errors and the increased circuit length make this implementation more affected by noise. This fact, added to the nature of the equation, leads to worse results, as anticipated. The values of ε admit a fifth-order Taylor expansion only if $T_1 \geqslant 5\mu$s. Even with Richardson extrapolation, the method cannot recover the energy for today's quantum computers T_1, and just for values of $500 - 1000\mu$s we reach errors of order $\varepsilon \sim 10^{-2}$. We conclude that larger thermal relaxation times in quantum computers are crucial for accurately estimating complex Hamiltonian ground-state energies.

Despite the high fidelity results theoretically reachable under ideal circumstances in Section 4.5.2, the results in Figures 4.15 and 4.16 indicate that noise significantly restricts the practical performance of the variational quantum PDE solver. Noise affects the number of qubits accessible to achieve a highly accurate representation and larger thermal relaxation times than those available for current devices are needed to apply error mitigation techniques. These findings suggest the need to explore alternative platforms suitable for encoding functions in a quantum register.

4.6 Conclusions and future perspectives

The main conclusion of this study is that the Fourier-based quantum register encoding in Chapter 2 is a highly efficient technique for representing functions and differential operators. We tested this representation with a new variational algorithm to solve static PDEs, leveraging the quantum register to obtain an error that decreases up to doubly exponentially in the number of qubits. The variational quantum PDE solver combines Fourier techniques with state-of-the-art variational principles to solve problems of interest in designing superconducting circuits. More concretely, we developed newer variational ansätze that are better suited to describe continuous differential functions, including their symmetries, and integrated them with different classical optimizers. The specific implementation of variational functions may have a non-favorable scaling. However, the embedded symmetries in the definition of the quantum circuits and Fourier interpolation allow a decrease in the number of parameters. Fourier interpolation also increases the approximation accuracy of larger grids with a small number of qubits.

Under ideal circumstances, even small grids reach high accuracies (10^{-5} infidelity), which can be increased up to five orders of magnitude in the infinite number of measurement limit, showing the true potential of this encoding. Noisy simulations show a decrease in the fidelity and the reachable number of qubits, as well as the need for error mitigation to approximate the energy. The restricted number of measurements and noise sources present in current NISQ devices restrict the accuracy of our algorithm, as well as the reachable grids due to the limited qubit scaling, preventing it from taking advantage of the favorable scalings provided by our quantum Fourier algorithms. These limitations are stronger for the more complex transmon and flux qubits' PDEs. For example, the experimental precision required for the transmon qubit, with errors $\varepsilon \simeq 10^{-4}$, can be costly in a scenario of NISQ computers with limited access and temporal stability.

An enhancement of classical optimization and the quantum evaluation of properties is needed to achieve this low error. Less demanding optimization techniques [222] could reduce the evaluation cost, allowing the solving of higher dimensional problems. The good results obtained for the R_Y ansatz with low depth and the more general ZGR ansatz suggest the possibility of developing simplified ansätze that encode smooth functions in quantum states with lower circuit depth. Heuristic studies of the entropy growth with the discretization density [38] support this possibility, but precise scaling and the ideal structure of the circuit are still missing. These ideas motivate other works, such as resampling images [223] in quantum machine learning and loading functions in a quantum register [224, 225, 226, 60].

Our algorithm can also compute excited states by following the prescriptions in Ref. [202]. It can also tackle general static equations $(\hat{D}(\nabla)+\hat{V}(\mathbf{x})-E)f=0$ by working with self-adjoint operators, as is normally the case when working with Lindblad operators, by analyzing the square of the equation $(\hat{D}(\nabla)+\hat{V}(\mathbf{x})-E)^{\dagger}(\hat{D}(\nabla)+\hat{V}(\mathbf{x})-E)f=0$. However, this is limited by the cost of products such as $\hat{D}(\nabla)\hat{V}(\mathbf{x})$, which require a larger number of gates, and this should probably be set aside to error-corrected quantum computers. Another possibility would be to study source PDEs and rewrite their solution as a cost function.

Quantum Fourier analysis techniques are also suitable for fault-tolerant quantum computers. Fourier encoding shows advantageous scalability with the number of qubits, especially for smooth, highly differentiable functions in many significant physical problems and other scientific computing domains. An exciting possibility is to combine spectral methods with quantum simulation tools to tackle initial-value problems, which appear more demanding than the eigenvalue problems addressed in this study. Moreover, it would be essential to explore how these methods can be merged with amplitude estimation to interrogate the properties of general functions better.

Nevertheless, quantum computing has still not reached the fault-tolerant era, and while our techniques have many advantages—up to doubly-exponentially decaying error with the number of qubits, exponential speed-up in the QFT, exponential savings in memory—current noisy devices limit their profits. Quantum-inspired tensor network methods for numerical analysis offer an alternative to leverage the quantum register encoding in the NISQ era. The representation of functions and operators as matrix product states and operators in sections 3.6 and 3.7, together with the finite precision algebra in section 3.5, create a suitable framework for developing quantum-inspired algorithms for numerical analysis. This framework constitutes one of the main contributions of this thesis, leading to methods to solve static and time-dependent PDEs in Chapters 5 and 6, respectively.

Chapter 5

Quantum-inspired solution of static PDEs

This chapter introduces new and existing quantum-inspired optimization and diagonalization techniques for solving static partial differential equations (PDEs) using matrix product states (MPS). In chapter 4, we presented the problem of the solution of Hamiltonian PDEs

$$Hf(\mathbf{x}) = [D(-i\nabla) + V(\mathbf{x})]f(\mathbf{x}) = Ef(\mathbf{x}), \tag{5.1}$$

where E is the energy of the Hamiltonian operator H, and $V(\mathbf{x})$ and $D(-i\nabla)$ are the potential and differential operators, respectively. Section 4.4 proposed a novel variational quantum algorithm to solve this problem. However, access and accuracy constraints of current quantum computers restrict the potential of the highly efficient encoding of functions and operators in a quantum register described in chapter 2.

The limitations of quantum computers nowadays have led to seeking quantum-inspired alternatives. Ewin Tang and colleagues have proposed numerous quantum-inspired algorithms that effectively dequantize their quantum counterparts, demonstrating the utility of these techniques in practical applications such as recommendation systems [227], principal component analysis [228], linear [229] and stochastic regression [230], and developing a sampling-based sublinear low-rank matrix arithmetic framework for dequantizing quantum machine learning [231].

The high memory compression and efficiency of tensor networks make them a compelling tool for quantum-inspired algorithms. Initially designed to study quantum many-body systems with low entanglement requirements, tensor networks' expressive power has been successfully used in other fields. These include optimization algorithms [232, 233] and machine learning [234], such as unsupervised and supervised learning [235, 236, 237, 238, 239, 240], generative modeling [237, 130, 241], reinforcement learning [242, 243], and the design of quantum circuits for machine learning [129], among others.

Numerical analysis can benefit from the memory compression of tensor network methods. In Ref. [38], García-Ripoll shows that MPS constitute an efficient and compact representation of highly differentiable multidimensional functions, which may accelerate numerical tasks—ranging from interpolation to solving PDEs. Later, Jobst and colaborators [39] derived a mathematical bound for the error of the MPS representation of classical data with a sufficiently quickly decaying Fourier spectrum, showing that such data can be efficiently represented with a reduced bond dimension. In both previous works, the representation of functions as MPS is an extension of the quantum register encoding in chapter 2, codifying the state of each qubit in a physical index of the MPS as illustrated in section 3.6. Combined with the MPO representation of operators and a basic set of operations and truncation algorithms, this approach leads to a finite-precision algebra (Section 3.5) suitable for developing quantum-inspired algorithms. Using this algebra, MPS constitute an efficient formalism to benefit from the quantum register encoding described in chapter 2.

MPS quantum-inspired numerical analysis overlaps with the field of tensor trains (TTs) and quantized tensor trains (QTTs), which were independently developed in applied mathematics [127]. The initial focus of using these tensorial representations for functions was reducing the scaling with the number of dimensions using their factorization to correlate degrees of freedom along different spatial dimensions. This factorization is the exact structure of certain multidimensional many-body wavefunctions [244] in the functional tensor train (FTT) expansion of a multidimensional function [137]. These techniques have successfully solved a variety of numerical analysis problems, such as stochastic problems [245, 246, 247], the Hamilton-Jacobi-Bellman equations [40, 41, 42], high-dimensional nonlinear PDEs [43], and the Schrödinger equation [248].

QTTs aim to prevent the exponential growth of the tensor size in the FTT by discretizing the coordinates and codifying them into tensor indices [55], either through binary encoding (as discussed in chapter 2), or using a tensor to represent all the coordinates for a given function dimension. Similar encodings have been developed in a quantum-inspired context. Ref. [38] presents a battery of quantum-inspired numerical analysis problems, such as Fourier sampling, interpolation, differentiation, and integration of PDEs. Analogous proposals have successfully solved some specific problems, such as the solution of nonlinear Schrödinger equations [44] and turbulence problems [45].

This chapter explores the solution of Hamiltonian PDEs (5.1) using MPS. In these applications, sophisticated, highly nonlocal MPOs encode PDEs, while MPS—in the QTT encoding—approximate the solution to those equations. The previous chapter (Chapter 4) focused on solving one-dimensional, physically motivated PDEs, specifically, superconducting qubit Hamiltonian PDEs. However, the limitations of current quantum computers prevent the variational quantum PDE solver from successfully addressing multidimensional problems. This chapter aims to solve multidimensional PDEs via quantum-inspired techniques. The numerical experiments tackle the harmonic oscillator PDE, as its well-behaved known solution allows for a fair comparison of the quantum-inspired methods.

The following sections provide a detailed description of the contributions presented in Ref. [49], a key element of this thesis. This study introduces five algorithms for finding the lowest eigenstates of the static Schrödinger Hamiltonian equation (5.1): explicit imaginary-time evolution, conventional and optimized steepest gradient descent, an explicitly restarted Arnoldi method, and a power method. All these algorithms are global optimization methods based on the MPS-MPO finite-precision algebra presented in section 3.5. The solution of the harmonic oscillator PDE (4.4) in up to two dimensions on a regular grid with up to 2^{30} points benchmarks these algorithms against the density matrix renormalization group (DMRG) method and an Arnoldi-like vector-based solver. The analysis shows that MPS-based methods exponentially outperform vector-based exact diagonalization regarding memory efficiency. However, imaginary-time algorithms have a higher cost due to their calibration requirements. The interpolated Arnoldi-like and DMRG algorithms asymptotically outperform the other MPS-based methods and exact diagonalization, providing significant time and memory usage advantages. In addition, the problem of the solution of a Hamiltonian PDE is indeed the search for the minimum eigenvalue. Thus, the algorithms of this chapter are also suitable for performing general optimization problems as long as they are amenable to an MPS encoding. Finally, the advantages found for MPS methods for resolving Hamiltonian PDEs support applying this finite-precision algebra to more complex numerical problems, such as the time evolution of an expansive state discussed in chapter 6.

5.1 Hamiltonian PDEs algorithms

This section introduces the new global optimization algorithms to solve static, Schrödinger-type PDEs (5.1) using MPS proposed in Ref. [49]. Initially intended to find the lowest eigenvalue of a Hermitian operator represented as a matrix, these optimization methods have been adjusted for the MPS-MPO representation of functions and linear operators, employing the finite-precision algebra for its implementation.

First, explicit methods—Euler, improved Euler, Runge-Kutta, and Runge-Kutta Fehlberg—address these PDEs via imaginary-time evolution [103, 104, 97, 249]. This Schrödinger-like equation converges to the lowest eigenstate of a Hermitian operator in the infinite imaginary-time limit. Imaginary-time evolution is a key ground-state search technique. Some notable MPS approaches include the time-evolving block decimation (TEBD) algorithm [103, 104, 97] and the time-dependent variational principle (TDVP) [105, 106, 107]. However, explicit methods incur an extra cost due to the calibration of the imaginary-time step and are also numerically unstable.

To avoid the inconveniences of imaginary-time evolution, we propose alternative techniques that directly minimize the energy functional

$$E[\psi] = \left(\frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle} \right) \tag{5.2}$$

of the desired PDE (5.1)

$$|\varphi_0\rangle = \mathrm{argmin}_{|\psi\rangle} E[\psi]. \tag{5.3}$$

The optimization parameter is the state ψ, and different strategies try to provide the optimal search directions to decrease $E[\psi]$. One of the most representative techniques is gradient descent, which searches for the optimal state following the opposite direction of the gradient of the cost function. Representing its update rule in a Krylov subspace improves this algorithm and paves the path for presenting Krylov-based methods, specifically explicitly restarted Arnoldi iteration and power iteration. The Arnoldi iteration results in an enhancement of the optimization due to adding new search directions, as many vectors of the Krylov subspace are considered.

The section ends by introducing the density matrix renormalization group (DMRG) [87, 88]. DMRG arose as a ground-state search algorithm for quantum many-body physics problems and was later adapted to a variational MPS formulation [91, 250, 142]. This MPS algorithm is also suitable for solving Hamiltonian PDEs (5.1) by understanding them as a ground-state search problem. Indeed, renormalization group methods have already been used for numerical analysis applications [98, 44]. Unlike the new algorithms introduced in this section, DMRG—as well as the previously mentioned TEBD and TVDP—works locally on each tensor. DMRG directly expresses the energy as a nonlinear function of all tensors that is sequentially optimized. The MPO representing a PDE is generally highly entangled; hence, global optimization algorithms are better suited to solve them. Due to the widespread use of DMRG, it acts as a benchmark for the global optimization methods in the numerical simulations in sections 5.2 and 5.3.

5.1.1 Imaginary-time evolution

The Schrödinger equation describes the evolution of a quantum state $\psi(x,t)$

$$i\hbar\partial_t\psi(x,t) = H\psi(x,t) \quad \rightarrow \quad \psi(x,t) = e^{-iHt/\hbar}\psi(x,0). \tag{5.4}$$

The exact evolution operator is a decaying exponential $U(t,H) = e^{-itH/\hbar}$, where t is the time and H is the Hamiltonian of the system. Imaginary-time evolution introduces an imaginary-time variable $t \rightarrow (-i\beta)$, $\beta \in \mathbb{R}$, leading to the expression

$$\partial_\beta\psi(\beta) = -H\psi, \tag{5.5}$$

for $\hbar = 1$. Then, in the infinite imaginary-time limit, the solution converges to the ground state φ_0 of the Hermitian operator H

$$\varphi_0 = \lim_{\beta\rightarrow\infty}\frac{1}{\mathcal{N}}|\psi(\beta)\rangle = \lim_{\beta\rightarrow\infty}\frac{1}{\mathcal{N}}\sum_n e^{-\beta E_n}\varphi_n, \tag{5.6}$$

where $\{\varphi_n\}$ is the basis of eigenstates of H, H has a non-degenerate gapped spectrum with eigenvalues E_n, $E_0 < E_1 \leq E_2 \leq E_3 \ldots$, and $\mathcal{N}$ is the normalization constant. Equation (5.6) holds provided that the initial state $\psi(\beta_0)$ has a non-zero overlap with the ground state φ_0. Initial symmetric states may improve convergence since they have zero overlap with odd eigenstates.

It is possible to construct the MPO of this evolution operator directly. However, while the MPO representation of H may be efficient, its exponential will have an exponentially large bond dimension in general [251]. Therefore, an efficient method is needed to construct an MPO to approximate U. One of the first proposals is the time-evolving block decimation

(TEBD) algorithm [103, 104, 97]. The TEBD algorithm decomposes the evolution into n small time steps Δt. It simplifies the operator via the Suzuki-Trotter decomposition of each unitary step $U(\Delta t, H)$. This method is well-suited for Hamiltonians with nearest neighbor interactions, where two parts compose the Hamiltonian, i.e., $H = H_{\text{odd}} + H_{\text{even}}$. These parts act on the odd and even sites of the MPS, respectively, allowing for the approximation of the exponential evolution in layers of two-site odd and even operators. Higher-order Suzuki-Trotter decompositions reduce the error at the expense of increasing the algorithm's complexity and cost. Different approximations of the evolution operator lead to other methods. The MPO $W^{I,II}$ method [249] generalizes the Euler approximation of the exponential operator and applies it as a local operator. Another alternative is the time-dependent variational principle (TDVP) [105, 106, 107], which relies on the tangent space representation of MPS to search for the optimal update of the quantum state for each evolution step.

Given the long-range correlations and potentially multiple layers of tensors within MPOs in numerical analysis (refer to the QFT MPO in [38]), it is relevant to look for methods that can be approximated using the finite-precision techniques. Explicit methods appear to be well-suited algorithms for this task. These methods rely on the Taylor expansion

$$\psi_{k+1} = \psi_k + \sum_p \frac{1}{p!}(-\Delta\beta H)^p \frac{\partial^{(p)}\psi}{\partial\beta^{(p)}}, \tag{5.7}$$

where the order of truncation p leads to different approximations of the evolution operator. In practice, most approaches choose an order of up to four, which provides a sufficiently accurate approximation at a reasonable implementation cost.

1. Euler method. This is an explicit, first-order Taylor approximation of the evolution, with a local error, i.e., one-step error, $\mathcal{O}(\Delta\beta^2)$ and simple update with a fixed time-step $\beta_k = k\Delta\beta$

$$\begin{aligned} \psi_0 &= \psi(\beta_0), \\ \psi_{k+1} &= \psi_k - \Delta\beta H \psi_k, \quad \text{for } k = 0, 1, \ldots, N-1. \end{aligned} \tag{5.8}$$

2. Improved Euler or Heun method. This is a second-order, fixed-step explicit method that uses two matrix-vector multiplications and two linear combinations of vectors to achieve a local error $\mathcal{O}(\Delta\beta^3)$

$$\psi_{k+1} = \psi_k - \frac{\Delta\beta}{2}\left[v_1 + H(\psi_k - \Delta\beta v_1)\right], \text{ with } v_1 = H\psi_k. \tag{5.9}$$

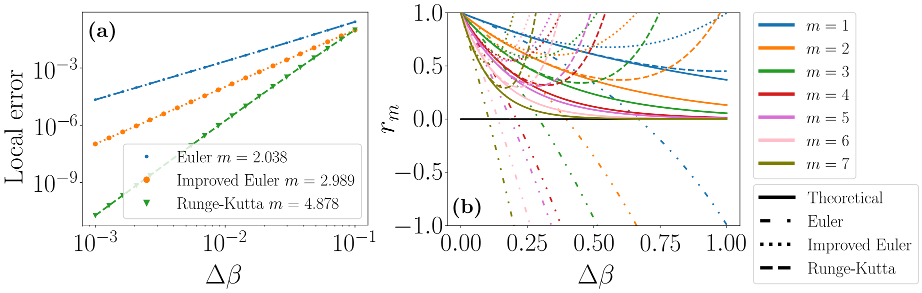


Figure 5.1: Theoretical study of Runge-Kutta methods. (a) Local error of the fixed-step Runge-Kutta methods for the one-dimensional quantum harmonic oscillator (4.4) 6-qubit discretization as a function of $\Delta\beta$. (b) Contraction ratio r_m of the one-dimensional quantum harmonic oscillator (4.4) 3-qubit discretization for $\Delta\beta$. The different line styles correspond to the methods used: theoretical evolution (solid), Euler (dash-dotted), improved Euler (dotted), and Runge-Kutta (dashed).

3. Fourth-order Runge-Kutta method. This algorithm achieves a local error $\mathcal{O}(\Delta\beta^5)$ using four matrix-vector multiplications and four linear combinations of vectors

$$
\begin{aligned}
\psi_{k+1} &= \psi_k + \frac{\Delta\beta}{6}(v_1 + 2v_2 + 2v_3 + v_4), \text{ with} \qquad (5.10)\\
v_1 &= -H\psi_k,\\
v_2 &= -H\left(\psi_k + \frac{\Delta\beta}{2}v_1\right),\\
v_3 &= -H\left(\psi_k + \frac{\Delta\beta}{2}v_2\right),\\
v_4 &= -H\left(\psi_k + \Delta\beta v_3\right).
\end{aligned}
$$

4. Runge-Kutta-Fehlberg method. The Runge-Kutta-Fehlberg (RKF) algorithm is an adaptative step-size solver [252] that integrates a solver with an error per step $O(\Delta\beta^5)$ with an error estimator $O(\Delta\beta^6)$ to dynamically determine a suitable $\Delta\beta$ that maintains the error within a specified tolerance level. The RKF method requires a good initial approximation of the step size, which a lower order, and hence, cheaper, method may provide. Its cost entails six evaluations of matrix-vector multiplication, six linear combinations, and potential repetitions of evolution steps for rejected step sizes.

The higher the order in the previous methods, the better the approximation of the imaginary-time evolution. However, this comes at a cost:

increasing the number of MPS-MPO operations. This leads to longer execution times and more error accumulation due to the successive MPS truncations. Two crucial ingredients for the performance of all methods include the calibration of the approximation order and the time step.

Explicit methods are unstable because they lack converging approximations of the eigenvalues. These algorithms only converge for a region of values of the time step $\Delta\beta$, where the exponential of the relevant eigenvalues is well approximated. The local error, which measures the error of the approximated solution for one step, is determined by the method's order p as $O(\Delta\beta^{p+1})$ (Figure 5.1(a)). A smaller $\Delta\beta$ leads to a better approximation of the imaginary evolution. However, since it does not directly target the minimum energy solution, it does not necessarily describe a better optimization path. We are interested in ground state convergence, so we set a convergence criterion for the step size $\Delta\beta$ based on the contractivity of all eigenvalues above the ground state. The optimal step size is that for which all the basis states, except for the ground state, converge faster to zero, as determined by the contraction ratio r_m for each energy level m,

$$r_m = \frac{\lambda_m(\beta, E_m)}{\lambda_0(\beta, E_0)}, \quad m > 0, \tag{5.11}$$

where $\lambda_n(\beta, E_n)$ is the eigenvalue of the evolution operator for the nth energy level and imaginary time β.

For the exact imaginary evolution, $\lim_{\beta\to\infty} r_m = \lim_{\beta\to\infty} e^{-\beta(E_m - E_0)} = 0$, since $E_m > E_0$ for $m > 0$. However, Runge-Kutta methods approximate this exponential evolution, and the optimal choice of the time step $\Delta\beta$ to minimize the number of steps to convergence depends on the method's order and the problem's spectrum. More precisely, the eigenvalues of the approximate transformation $W_p(\Delta\beta, H) \simeq U(\Delta\beta, H) + \mathcal{O}(\Delta\beta^{p+1})$, $\lambda'(\Delta\beta, E_m)$, may deviate from the contracting limit $|\lambda'(\Delta\beta, E_m)/\lambda'(\Delta\beta, E_0)| < 1$ due to an improper calibration of $\Delta\beta$, leading to a divergence of the method. Figure 5.1(b) shows the contraction ratios $r_m \leq |1|$ of the energy levels corresponding to the quantum harmonic oscillator (4.4) 3-qubit discretization for a range of values of $\Delta\beta$. Higher-order methods provide a more accurate approximation of the exact imaginary-time evolution for small step size values, as expected. Lower-order methods, such as Euler, can achieve smaller contraction ratios and faster convergence to the ground state for larger $\Delta\beta$ than higher-order methods, which better approximate the imaginary evolution path. This indicates the possibility for other paths better suited for energy minimization, like those in the next sections. In practice, estimating the optimum imaginary time step or ensuring stability for a given choice is unviable since it implies knowing the spectrum of the problem to solve.

Another approach to solving the problem of imaginary-time evolution is to use implicit methods, which are generally more stable than explicit ones. For example, the Crank-Nicolson formula, $(1+\Delta H\beta/2)\psi_{k+1} = (1-\Delta H\beta/2)\psi_k$, can be implemented using conjugate-gradient solvers [38]. However, they may involve a high and uncontrolled cost per step, which does not compensate for their potential advantages in this concrete application.

Alternative imaginary-time evolution techniques propose to use the Suzuki-Trotter approximation to separately apply the potential $V(\hat{\mathbf{x}})$ and the differential $D(-i\nabla)$ terms [253]. However, this approximation is only beneficial if the separate application of the operators is more efficient, which is not the case for nonlocal MPOs arising from PDEs. Other options are second-order differencing, Chebyshev polynomial expansion, and Lanczos propagation schemes [253], and the use of real-time evolution techniques [104], with a cost similar to the optimization algorithms introduced in the following section. Finally, while the implementation of Runge-Kutta methods to approximate the time evolution within the DMRG algorithm [98] has also been explored, the advantages of the finite-precision algebra approach described in section 3.5 also pertain in this context.

5.1.2 Gradient descent

The explicit methods in Section 5.1.1 are inherently prone to numerical instability and require step calibration, posing challenges in practical implementation. However, these challenges can be overcome by methods that minimize the energy functional of the PDE (5.3).

One of the most extended optimization routines is gradient descent. Gradient descent updates the approximation of the solution by a displacement along the opposite direction of the gradient, which is the direction of the fastest energy decrease. One step of this optimization method is given by

$$\psi_{k+1} = \psi_k + \Delta\beta \frac{\delta E}{\delta\psi}, \quad \text{with } \frac{\delta E}{\delta\psi} = (H - \langle H\rangle\, \mathbb{I})\psi, \tag{5.12}$$

where $\langle H\rangle = \langle\psi|H|\psi\rangle$. The step $\Delta\beta < 0$ determines how far to move along the direction of the functional gradient $\frac{\delta E}{\delta\psi}$. MPS-based machine learning methods [254, 255, 232] already use gradient-descent, and in this context, the step $\Delta\beta$ is usually known as the learning rate.

It is possible to analytically compute the optimal $\Delta\beta$ to minimize (4.1), avoiding the problem of step calibration that affects imaginary-time evolution methods in section 5.1.1. By substituting the update rule (5.12) in the cost functional (5.2) and minimizing with respect to $\Delta\beta$

$$\Delta\beta_{-} = \frac{\langle H'^3\rangle - \sqrt{\langle H'^3\rangle^2 + 4\langle H'^2\rangle^3}}{2\langle H'^2\rangle^2}, \quad \text{with } H' = H - \langle H\rangle \mathbb{I}. \tag{5.13}$$

Rewriting the optimal step size in terms of the expectation values of $\langle H\rangle, \langle H^2\rangle$ and $\langle H^3\rangle$ reduces the computational cost of gradient descent to two matrix-vector multiplications and one linear combination.

Appendix C includes a detailed derivation of Eqs. (5.12) and (5.13). The gradient descent method enables a global update of the MPS, and the cost function can be adapted to solve other types of PDEs, such as source PDEs (see Appendix D), which are too expensive for DMRG-like algorithms.

5.1.3 Improved gradient descent

Enhancing gradient descent by recasting it as a Krylov subspace method is possible. A Krylov subspace of order L, i.e., with L vectors generated by the linear operator $H \in \mathbb{C}_{\mathbb{N}\times\mathbb{N}}$, with $L < N$, is defined as the vectorial subspace

$$\mathcal{K}_L = \text{lin}\{|\psi_k\rangle, H|\psi_k\rangle, \ldots, H^{L-1}|\psi_k\rangle\}. \tag{5.14}$$

Krylov methods operate within this subspace to iteratively approximate the solution. They are considered one of the most effective modern diagonalization routines, featured in popular libraries like ARPACK [256].

Gradient descent's update rule (5.12) can be written in a Krylov subspace with $L = 2$ as

$$\psi_{k+1} = v_0\psi_k + v_1\xi_k = (1 - \langle H\rangle\Delta\beta)\psi_k + \Delta\beta H\psi_k, \tag{5.15}$$

where the state is now a linear combination of the elements of such Krylov basis with coefficients $v_0 = (1 - \langle H\rangle\Delta\beta)$ and $v_1 = \Delta\beta$. Let us generalize this method by searching the optimal vector $\mathbf{v}^T = (v_0, v_1)$ that leads to the steepest descent at every step. A reformulation of the energy minimization in this two-dimensional Krylov subspace results in its expression as the ratio of two quadratic forms

$$E[\chi] = E(\mathbf{v}) = \frac{\mathbf{v}^\dagger A\mathbf{v}}{\mathbf{v}^\dagger N\mathbf{v}}, \tag{5.16}$$

with Hermitian matrices

$$A = \begin{pmatrix} \langle\psi|H|\psi\rangle & \langle\psi|H|\xi\rangle \\ \langle\xi|H|\psi\rangle & \langle\xi|H|\xi\rangle \end{pmatrix} = \begin{pmatrix} \langle\psi|H|\psi\rangle & \langle\psi|H^2|\psi\rangle \\ \langle\psi|H^2|\psi\rangle & \langle\psi|H^3|\psi\rangle \end{pmatrix}, \text{ and} \quad (5.17)$$

$$N = \begin{pmatrix} \langle\psi|\psi\rangle & \langle\psi|\xi\rangle \\ \langle\xi|\psi\rangle & \langle\xi|\xi\rangle \end{pmatrix} = \begin{pmatrix} \langle\psi|\psi\rangle & \langle\psi|H|\psi\rangle \\ \langle\psi|H|\psi\rangle & \langle\psi|H^2|\psi\rangle \end{pmatrix}. \quad (5.18)$$

The minimum of the cost function (5.16) satisfies

$$\frac{\delta E}{\delta \mathbf{v}^*} = \frac{1}{\mathbf{v}^\dagger N \mathbf{v}} \left(A\mathbf{v} - E(\mathbf{v})N\mathbf{v}\right) = 0, \quad (5.19)$$

and hence the optimal direction $\mathbf{v}$ corresponds to the solution of the generalize eigenvalue problem

$$A\mathbf{v} = \lambda N \mathbf{v}, \quad (5.20)$$

where the minimum eigenvalue $\lambda = E(\mathbf{v})$ gives the optimal energy for the k-th step. The generalized eigenvalue problem (5.20) can be solved analytically and numerically, providing both a new estimate of the energy and a new state $|\psi_{k+1}\rangle$. This method is equivalent to the gradient descent in section 5.1.2, but the implicit normalization improves the numerical stability. The normalization matrix N is relevant for the MPS implementation since the truncation error affects the orthogonality of the basis.

5.1.4 Arnoldi iteration

Let us extend the previous approach to higher-dimensional Krylov subspaces using an update rule

$$\psi_{k+1} = \sum_{m=0}^{L-1} v_m H^m \psi_k. \quad (5.21)$$

This equation describes an Arnoldi method [257] that estimates the energy functional using two matrices, A (5.17) and N (5.18), containing the matrix elements of H and the identity computed in the Krylov basis of order L. The restriction of the maximum number of Krylov vectors n_v limits the cost of the algorithm, and in practical applications, a limited number of vectors is enough to obtain high accuracy, as shown in section 5.3 for the $n_v = 5, 10$ implementation. Solving Eq. (5.20) returns the coefficients $\mathbf{v}_k \in \mathbb{C}^L$ of the Krylov basis. Then, an upgraded version of the Arnoldi method restarts the basis using the new approximation

ψ_{k+1} to construct it. This particular implementation of the Arnoldi iteration is known as an explicitly restarted Arnoldi iteration. An early restart may avoid divergences due to an ill-conditioned matrix N when the method is close to convergence. Adding a memory factor $\gamma = -0.75$ in the update the optimization step $|\xi_{k+1}\rangle = (1-\gamma)\,|\psi_{k+1}\rangle + |\psi_k\rangle$ [258] may also improve convergence.

The computation of the expectation values to construct the matrices A (5.17) and N (5.18) and the linear combination of the optimization step (5.21) dominate the cost of the algorithm. For our problem (4.1), the cost of solving the generalized eigenvalue problem (5.16) is negligible due to the small dimension of the Krylov basis.

Different applications already benefit from the Lanczos and Arnoldi methods for MPS states. Combining these methods with DMRG allows computing dynamical correlation functions [259], which can be improved with the adaptive Lanczos-vector method [260, 261]. In the context of the variational DMRG, Lanczos and Arnoldi methods [108, 109] are used to evolve quantum states. In contrast to the proposed techniques, DMRG-like approaches may lead to larger bond dimension MPS. Additionally, in the same spirit as this work, Chebyshev filter expansions [262] are employed as iterative schemes for approximate diagonalization around regions of the spectrum.

5.1.5 Power method

The power method, also known as power iteration, finds the largest eigenvalue $\lambda_{\max} = \|H\|$ (in absolute value) and the corresponding dominant eigenvector $(H - \lambda_{\max})\varphi_{\max} = 0$ of a linear operator H. This algorithm can be modified to find the smallest eigenvalue $\lambda_{\min}$ (in absolute value), making it adequate for solving the ground state search problem in (5.1). Let us first introduce the original algorithm, as its derivation is easier to understand, facilitating the comprehension of the lowest eigenvalue search.

The algorithm starts with a random vector ψ_0—for faster convergence, an approximation of the desired eigenvector $\varphi_{\max}$—and iteratively approximates the solution by the recurrence relation

$$\psi_{k+1} = \frac{H\psi_k}{\|H\psi_k\|}, \tag{5.22}$$

which constructs the Krylov basis. The method assures convergence to the dominant eigenvector $\lambda_{\max} = \|H\varphi_{\max}\|$ in the limit of infinite iterations under two conditions: (i) $|\lambda_{\max}| > |\lambda_m|$, $\forall \lambda_m \neq \lambda_{\max}$, and (ii)

$\langle \psi_0 | \varphi_{\max} \rangle \neq 0$. To illustrate this, let us rewrite the recurrence relation ((5.22)) as

$$\psi_{k+1} = \frac{H^{k+1}\psi_0}{\|H^{k+1}\psi_0\|}. \tag{5.23}$$

This expression can be written in terms of the Jordan canonical form of H, $H = PJP^{-1}$, where J is an upper triangular matrix with a diagonal of eigenvalues (counting multiplicity) and upper diagonal of ones, and P is the column matrix of the eigenvectors and generalized eigenvectors of H. Thus, if $\lambda_{\max}$ is non-degenerate, the first column of P corresponds to its eigenvector $\varphi_{\max}$. Then,

$$\psi_k = \frac{H^k\psi_0}{\|H^k\psi_0\|} = \frac{(PJP^{-1})^k\psi_0}{\|(PJP^{-1})^k\psi_0\|} = \frac{PJ^kP^{-1}\psi_0}{\|PJ^kP^{-1}\psi_0\|}. \tag{5.24}$$

Expressing ψ_0 as a linear combination of the columns of P, $\psi_0 = c_{\max}\varphi_{\max} + \sum_{i=1}^{N-1} c_i\varphi_i$

$$\psi_k = \left(\frac{\lambda_{\max}}{|\lambda_{\max}|}\right)^k \frac{c_{\max}}{|c_{\max}|} \frac{\varphi_{\max} + \frac{1}{c_{\max}} P \left(\frac{1}{\lambda_{\max}}\right)^k \sum_{i=1}^{N-1} c_i e_i}{\|\varphi_{\max} + \frac{1}{c_{\max}} P \left(\frac{1}{\lambda_{\max}}\right)^k \sum_{i=1}^{N-1} c_i e_i\|}. \tag{5.25}$$

Hence, in the limit of an infinite number of iterations k

$$\psi_k = e^{i\phi_k} \frac{c_{\max}}{|c_{\max}|} \frac{\varphi_{\max}}{\|\varphi_{\max}\|} + r_k, \tag{5.26}$$

where $e^{i\phi_k} = (\lambda_{\max}/|\lambda_{\max}|)^k$, and $\|r_k\| \to 0$ as $k \to \infty$. Equation (5.25) shows that if conditions (i) and (ii) are not verified, then the sequence does not converge to the eigenvector corresponding to the largest eigenvalue. This convergence is geometric, with ratio $|\lambda_{\max}/\lambda_1|$, and hence depends on the spectral gap between the first two dominant eigenvalues, so the larger the gap, the faster the convergence.

The product of the operator H on ψ_k and its normalization dominates the cost of one iteration of the power method, with no linear combination of states to obtain the new approximation of the solution.

To find the smallest eigenvalue $\lambda_{\min}$ in absolute value of a non-negative operator H, the inverse power method uses the shifted and inverted operator $(H - \epsilon)^{-1}$, designed so that $\|(H - \epsilon)^{-1}\| = |\lambda_{\min}|^{-1}$ for $\epsilon = 0$. The new recurrence relation

$$\psi_{k+1} = \frac{H^{-1}\psi_k}{\|H^{-1}\psi_k\|} \tag{5.27}$$

involves solving the linear equation $H\xi = \psi_k$ and normalizing $\psi_{k+1} = \xi / \|\xi\|$. Given our intention to solve this equation using MPO (or combinations of them) of significant complexity, rather than utilizing a DMRG-like solver, we opt for a conjugate gradient method, bringing the power method closer to a gradient descent algorithm. This includes an added cost for the power method, with one matrix multiplication and two linear combinations per conjugate gradient method's iteration.

5.1.6 Density matrix renormalization group

The density matrix renormalization group (DMRG) algorithm was originally proposed by White [87, 88] as a ground-state search algorithm in the context of quantum many-body physics. In its later reformulation for MPS [91, 250, 142], the MPS acts as variational ansatz for the minimization of the energy functional $E[A] = \frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle}$, where A are the constituent tensors of a family of MPS states $|\psi[A]\rangle$. This approach has been further extended to the calculation of excited states [263, 264], dynamical systems [265], and the real-time evolution of quantum systems via time-dependent DMRG (tDMRG) [110, 111, 266]. In this chapter, the energy functional is the Hamiltonian in (5.1), and thus, its minimization yields the solution to static Schrödinger-type Hamiltonian PDEs.

As opposed to the rest of the methods in this section, DMRG performs a local optimization of the MPS sites. This algorithm rewrites the energy functional $E[A]$ in terms of all the constituents tensors $\{A_1, A_2, \ldots, A_N\}$. From the perspective of a single site, the expression is a quotient of quadratic forms

$$E[A_i] = \frac{\sum_{\alpha_n,\alpha_{n+1},i_n} \sum_{\beta_n,\beta_{n+1},j_n} A^{i_n\,*}_{\alpha_n,\alpha_{n+1}} \bar{H}^{i_n,j_n}_{\alpha_n,\alpha_{n+1},\beta_n,\beta_{n+1}} A^{j_n}_{\beta_n,\beta_{n+1}}}{\sum_{\alpha_n,\alpha_{n+1},i_n} \sum_{\beta_n,\beta_{n+1},j_n} A^{i_n\,*}_{\alpha_n,\alpha_{n+1}} \bar{N}^{i_n,j_n}_{\alpha_n,\alpha_{n+1},\beta_n,\beta_{n+1}} A^{j_n}_{\beta_n,\beta_{n+1}}}, \tag{5.28}$$

with tensors $\bar{H}$ and $\bar{N}$ that can be deduced from the operator H and all other tensors in the quantum state $|\psi\rangle$. The tensors $A^{j_n}_{\beta_n,\beta_{n+1}}$ are representations of a quantum state $|\psi^{(n)}\rangle = \sum_{\beta_n,\beta_{n+1},j_n} A^{j_n}_{\beta_n,\beta_{n+1}} |\beta_n j_n \beta_{n+1}\rangle$ in a Hilbert space, and the tensors $\bar{H}$ and $\bar{N}$ as Hermitian operators in that space

$$E[\psi^{(n)}] = \frac{\langle\psi^{[n]}|\bar{H}_n|\psi^{[n]}\rangle}{\langle\psi^{[n]}|\bar{N}_n|\psi^{[n]}\rangle}. \tag{5.29}$$

Therefore, the solution of the eigenvalue problem $\bar{H}_n|\psi^{[n]}\rangle = \lambda\bar{N}_n|\psi^{[n]}\rangle$ leads to the minimum energy with respect to the given site.

This process is iteratively repeated on successive sites, following a similar methodology to the truncation algorithm in section 3.5. DMRG is also suitable to vary the size of the tensors by optimizing two-site tensors $|\psi^{(n)}\rangle \sim A^{j_n}_{\beta_n,\beta_{n+1}} A^{j_{n+1}}_{\beta_{n+1},\beta_{n+2}} |\beta_n, j_n, j_{n+1}, \beta_{n+1}\rangle$, and using the singular value decomposition to determine the optimal size for β_{n+1} on each site.

The DMRG algorithm is included for completeness in the benchmarks in sections 5.2 and 5.3 as a diagonalization algorithm that directly optimizes the MPS tensors given a generic MPO structure. This enables a comparison of MPS local versus global optimization for solving PDEs.

5.2 Method calibration and comparison

Let us compare the MPS algorithms introduced in section 5.1 to determine the most efficient method to solve static Hamiltonian PDEs (5.1). This comparison relies on the one-dimensional quantum harmonic oscillator PDE with $m = \omega = \hbar = 1$

$$\left(-\frac{1}{2}\partial_x^2 + \frac{1}{2}x^2\right)\psi = E\psi, \tag{5.30}$$

which was introduced in section 4.1. The Gaussian ground-state of the harmonic oscillator equation is adequate for an MPS representation [38, 39], enabling a fair comparison of the methods and the evaluation of the quantum register encoding in a framework of limited precision.

This benchmark resorts to the second-order finite difference method (Section 3.7) to approximate the differential operators. Its MPO representation has a bond dimension $\chi = 3$. This, along with the reduced bond dimension of the potential operator, results in an MPO with limited entanglement amenable to both global and local optimization algorithms, allowing a comparable analysis of state-of-the-art DRMG versus the global optimization methods in Section 5.1

The domain of equation (5.30) is defined as $x \in [-L/2, L/2)$, with $L = 10$, and $\Delta x = L/2^n$, where n is the number of sites of the MPS, i.e., qubits of the corresponding quantum register. We use the SElf-Explaining Matrix-Product-State (SeeMPS) [51] library (Chapter 7) to implement this problem—as well as the solution of the 2D Hamiltonian PDE in section 5.3 and the evolution problem in Chapter 6. The SeeMPS library includes the algorithms resulting from this thesis and enables an accessible introduction to MPS and quantum-inspired algorithms.

The study uses four figures of merit to analyze the proposed methods: (i) the difference $\varepsilon = |E_0 - E_n|$ between the exact eigenvalue E_0 and the

	Euler	Improved Euler	Runge-Kutta	RKF	Gradient descent	IRArnoldi
Cost	7	14	28	43	13	$\frac{6(n_v-2)+13}{n_v-1}$

Table 5.1: Cost of the MPS-MPO finite-precision methods. The cost is estimated as a function of the number of MPO-MPS multiplications and MPS linear combinations.

estimate E_n using n qubits, (ii) the 1-norm distance $\|\varphi_0 - \psi_{\rm n}\|_1$ between the approximate solution $\psi_{\rm n}$ and the exact ground state discretized on the same grid φ_0, (iii) the standard deviation of the energy on the final state

$$\sigma = \sqrt{\langle\psi_{\rm n}|H^2|\psi_{\rm n}\rangle - \langle\psi_{\rm n}|H|\psi_{\rm n}\rangle^2}, \tag{5.31}$$

and, (iv) the infidelity with respect to φ_0

$$1 - F = 1 - |\,\langle\varphi_0|\psi_{\rm n}\rangle\,|^2. \tag{5.32}$$

It is not fair to compare these figures of merit based on the number of iterations, as the numerical methods have different costs associated with their MPS operations. The more expensive operations are the application of an MPO on an MPS and the linear combination of MPS. Table 5.1 gauges the cost as the number of MPS combinations and MPO-MPS multiplications of each algorithm. All methods have a constant cost except for the Arnoldi method, which depends on the size n_v of the Krylov basis. The numerical experiments will consider different basis sizes since this choice is heuristic—as in similar state-of-the-art diagonalizers (ARPACK [256], Matlab [267], Scipy [268]). Although these values provide an approximate idea of the cost of the methods, they dismiss the cost of less expensive operations and do not allow an equitable comparison with DMRG and the Power iteration. A better approach is to use the CPU time to quantify the algorithms' costs.

The explicit imaginary-time evolution methods (Section 5.1.1) precise calibration of their optimal step size $\Delta\beta$. Since the solution to the problem is known, a standard classical optimization routine approximates the optimal step size by minimizing the steps to reach a fixed error in the approximation of the ground state energy. This calibration supposes an extra cost for imaginary-time evolution methods, multiplying the total cost by the number of executions performed in the optimization. In addition, this is not possible for practical problems since it implies knowing the solution. Still, we use it in this benchmark to provide the best

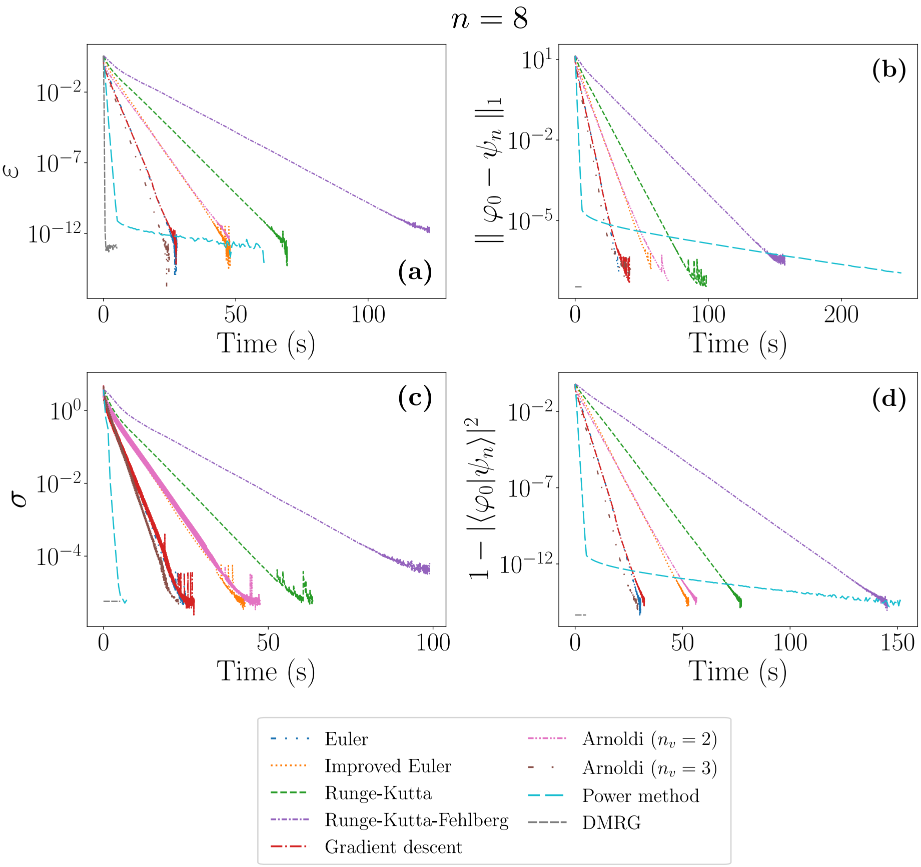


Figure 5.2: Figures of merit—approximation error, norm-1 distance, standard deviation, and infidelity—versus execution time for the harmonic oscillator PDE. The one-dimensional PDE is defined over the interval $x \in [-L/2, L/2)$, with $L = 10$, using a discretization with $n = 8$ qubits and $\Delta x = L/2^n$. We plot (a) the absolute error ε in the estimation of the eigenvalue, (b) the norm-1 distance, (d) the infidelity (5.32) with respect to the numerically exact solution, and (c) the standard deviation (5.31) of the Hamiltonian over the computed eigenstate.

possible scenario for these methods, enabling a fair comparison of all algorithms introduced in Section 5.1.

Figures 5.2(a)-(d) show the values of the figures of merit (i)-(iv) as a function of the CPU execution time when using an 8-qubit discretization and numerically exact truncation operations of the MPS for the norm-2 distance tolerance (3.13). These plots exhibit that all methods reach similar accuracy once convergence is achieved, regardless of the figure of

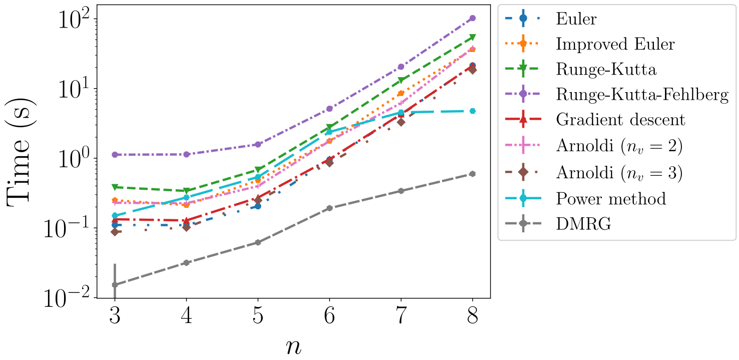


Figure 5.3: The one-dimensional PDE is defined over the interval $x \in [-L/2, L/2)$, with $L = 10$ and a discretization of n qubits, $\Delta x = L/2^n$. The execution time to estimate the Hamiltonian's lowest eigenvalue with an error $\varepsilon < 10^{-10}$ is averaged over ten runs.

merit. Therefore, the subsequent discussion will focus on the error in approximating the energy ε. This representation favors the imaginary-time evolution methods since they use the optimal time step $\Delta\beta$, but they do not consider that this optimization process increases the total time by one order of magnitude, rendering the other methods preferable.

Let us analyze the performance scaling by examining the execution time to achieve $\varepsilon < 10^{-10}$ against the number of qubits (Figure 5.3). Most imaginary-time evolution algorithms perform worse than the other techniques, except the Euler method, which behaves similarly to the improved gradient descent (or Arnoldi with $n_v = 2$) without including the calibration time. DMRG, Arnoldi, and gradient descent are the best-performing methods. Among these, DMRG presents the best resource's scaling and, when applicable, is the best option.

The simulations in Figures 5.2 and 5.3 use a numerically exact precision tolerance, allowing an arbitrary bond dimension. Figure 5.4(a) shows that the methods remain stable when imposing more strict truncation tolerances, with a higher error due to the loss of information of the truncation, as we observe in the decrease in the bond dimension (Figure 5.4(b)). The solution of the 1D quantum harmonic oscillator (4.4) is a Gaussian, and due to its smooth and bandwidth-limited nature, it needs a considerably small bond dimension χ for a highly accurate representation [162, 39] (Figure 3.12(c)), evidencing the efficient approximation of highly differentiable functions with MPS/QTT.

The findings presented in this section conclude that all methods achieve similar accuracy upon convergence, but the time to convergence

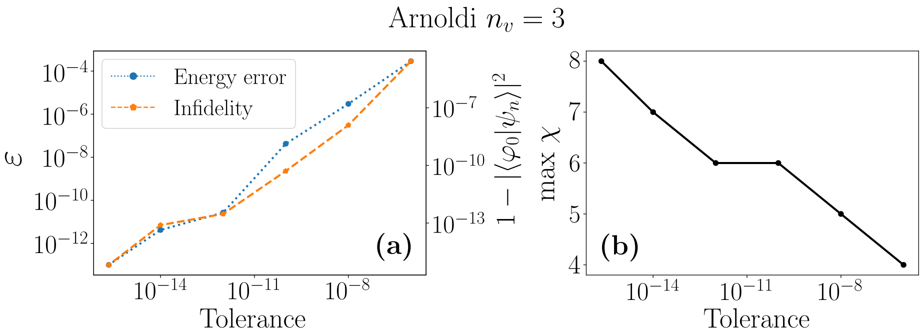


Figure 5.4: Energy error ε and maximum bond dimension χ for different truncation tolerances for the harmonic oscillator PDE. Results for different truncation tolerances in the MPS-MPS and MPO-MPS operations for an Arnoldi diagonalization with $n_v = 3$ vectors and a discretization of $n = 8$ qubits. NE stands for numerically exact, indicating that the truncation tolerance is the machine precision of floating point operations. (a) Error in estimating the energy ε. (b) Maximum bond dimension χ of the resulting MPS solution.

varies for each of them. The inherent calibration of explicit imaginary-time methods renders them inefficient compared to the other proposed methods. Among all algorithms, the fastest convergence of DMRG makes it the best option as long as the entanglement of the MPO is adequate for this local optimization. In situations where DMRG is unsuitable for the given problem, Arnoldi iteration stands out as the most appropriate algorithm, and its efficacy can be enhanced by heuristically selecting the optimal number of vectors n_v.

5.3 Benchmark: squeezed harmonic oscillator

This section studies the suitability of MPS methods as an alternative to state-of-the-art Arnoldi and Lanczos diagonalization packages based on matrix-vector multiplications—ARPACK [256] and Primme [269, 270]. The analysis focuses on the use of the best-performing Hamiltonian PDE solvers—Arnoldi iteration and DMRG—to solve a more demanding problem: the two-dimensional quantum squeezed harmonic oscillator PDE, which increases the entanglement needs for the MPS representation [38]. The Hamiltonian of the multidimensional squeezed harmonic oscillator is given by

$$H = -\frac{\hbar^2}{2m}\nabla^2 + \frac{1}{2}m\mathbf{x}^\dagger A\mathbf{x}. \tag{5.33}$$

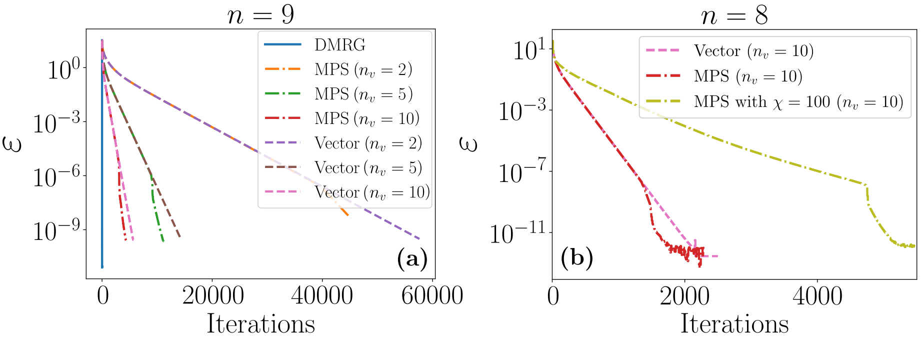


Figure 5.5: Single execution energy error with the number of iterations for the squeezed harmonic oscillator PDE. (a) Error in the approximation of the ground state energy ε with the number of steps for the squeezed harmonic oscillator with $n = 9$ qubits per dimension with arbitrarily large bond dimension for numerically exact truncation in the squared norm of the singular values and simplification and linear combination algorithms. (b) Error in the approximation of the ground state energy ε with the number of steps for the squeezed harmonic oscillator with $n = 8$ qubits per dimension for the Arnoldi method with $n_v = 10$ for the vector and MPS implementation with arbitrarily large bond dimension and maximum bond dimension $\chi = 100$.

This reduces to the one-dimensional equation (5.30) of the previous section for $A = \omega^2$. For the more complex two-dimensional squeezed case, the matrix A contains some degree of correlation. The Hamiltonian represents a two-dimensional squeezed harmonic oscillator, rotated an angle θ and *squeezed* a factor $\sigma_{\text{min}}/\sigma_{\text{max}}$, where

$$A = O^T(\theta) \begin{pmatrix} 1/\sigma_{\text{max}}^4 & 0 \\ 0 & 1/\sigma_{\text{min}}^4 \end{pmatrix} O(\theta), \tag{5.34}$$

with the orthogonal transformation

$$O(\theta) = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}. \tag{5.35}$$

For this matrix and $m = 1$, the ground state energy is given by

$$E_{0,0} = \frac{1}{2}\left(\frac{1}{\sigma_{\text{max}}^2} + \frac{1}{\sigma_{\text{min}}^2}\right). \tag{5.36}$$

The harmonic oscillator is a gapped problem with a non-degenerate spectrum, making it suitable as a benchmark for proof of concepts and comparison performance. However, due to the correlations, its complexity increases exponentially, making it challenging for standard classical approaches.

The parameters chosen for numerical simulations are a rotation angle $\theta = \pi/4$ and a large squeezing $\sigma_{\text{min}}/\sigma_{\text{max}} = 0.5$. As the PDE is now multidimensional, a suitable qubit ordering must be chosen. In this case, the qubits are ordered sequentially, first by coordinate and then by significance, following order A in Ref. [38]. Again, the truncation tolerance is set to the numerically exact machine precision according to the square norm difference, both for the SVD truncation and the simplification and linear combination algorithms.

Analogously to section 5.2, let us start with the scaling of the error in the approximation of the ground state energy ε with the number of iterations for a fixed number of qubits. Figure 5.5(a) displays the error decrease for the different methods considered for a problem discretized with $n = 9$ qubits per dimension. Given enough iterations, all methods achieve low errors, but Krylov-subspace methods require more steps to converge, both in the MPS and vector representations. Indeed, for a sufficiently small truncation tolerance, vectors and MPS follow a similar path up to a point in which the MPS truncation leads to slightly faster dissipation. Setting the truncation tolerance does not set a fixed maximum for the bond dimension, and some implementations need more restrictions to obtain a reasonable implementation cost. Figure 5.5(b) inspects the effect of limiting the maximum bond dimension for the MPS tensors. The MPS implementation no longer follows the exact trajectory but converges to a low error $\varepsilon = 10^{-11}$. Despite the larger number of iterations, reducing the bond dimension can speed up the cost per step and accelerate the total execution time.

To compare the scaling of resources with the number of qubits, let us consider a problem with $n = 3$ qubits per dimension and a tensor size $\chi = 100$, as depicted in Figure 5.6. This benchmark adds the Power iteration since it is usually included as a ground base for Krylov-subspace methods. The asymptotic execution time of MPS-based Krylov methods is better than the one of their vector counterparts, which is confirmed by the slope of the relative execution time for large problems (Figure 5.6(a)). This better performance results from the smaller memory requirements of the MPS representation, as its scaling is algebraic instead of exponential in the vector representation.

In this implementation, DMRG continues outperforming the Krylov-based methods in memory and time, demonstrating an exponential ad-

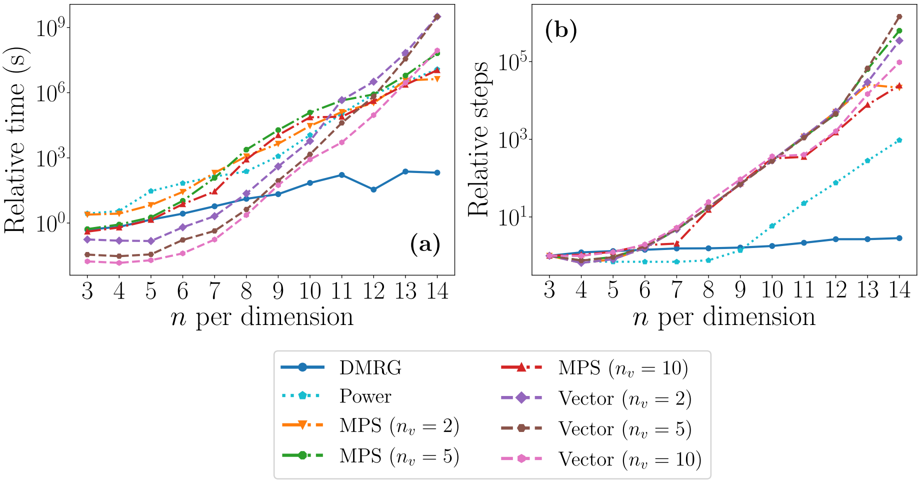


Figure 5.6: Results of the solution of the squeezed harmonic oscillator equation (4.4). (a) Relative time. (b) Relative steps.

vantage in the execution time. This exponential speedup arises in the nonlocal approach in the DMRG optimization. DMRG is a “local” optimization method for MPS, minimizing the energy functional for each pair of neighboring tensors. From a mathematical perspective, the renormalization of DMRG acts like a sort of interpolation. The sweeps from qubit 0 up to qubit $2n-1$ are equivalent to solving the problem along the X and the Y directions, starting with the longest length scales first and refining the solution within each sweep and between consecutive sweeps.

This understanding inspired us to upgrade the Krylov-based method with a similar strategy to decrease its cost. This approach reuses previous solutions, using finite differences interpolation of the n-qubit per dimension MPS solution to obtain the initial approximation for the $(n+1)$-qubit problem. When using interpolation (Figure 5.7(a)), the problem also becomes solvable for the Arnoldi iteration since the initial approximation improves with each increase in the number of qubits. This upgrade is particularly beneficial for the more expensive larger qubit problems, as now they need a few iterations to converge. All methods have similar accuracies, demonstrating the capacity of MPS to obtain high-precision solutions. After this enhancement of the solution technique, the MPS-based Arnoldi is more competitive with DMRG in terms of execution time.

MPS are advantageous in scenarios that enable memory compression while maintaining high accuracy, allowing for more efficient han-

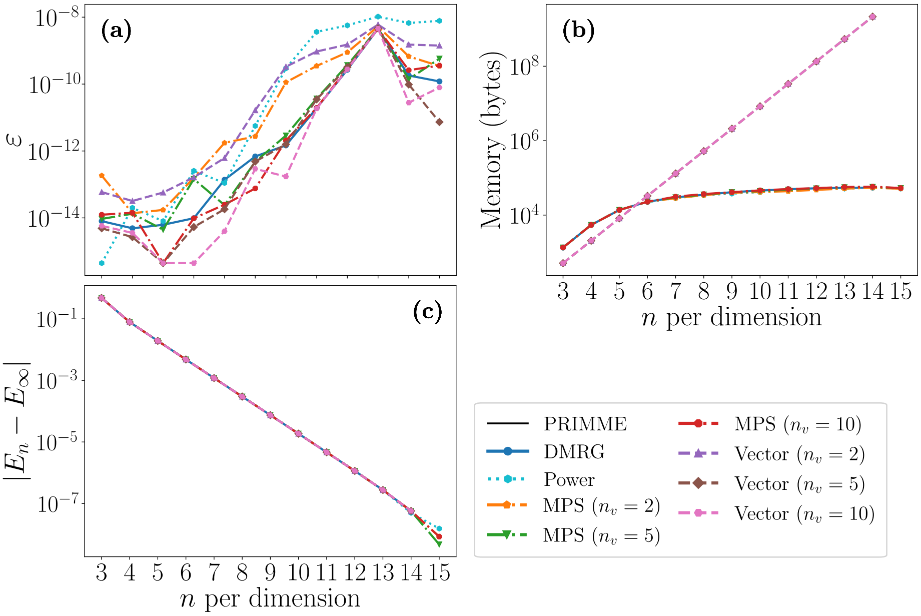


Figure 5.7: Squeezed harmonic oscillator results using finite difference interpolation (5.33). (a) ε. (b) Memory. (c) Theoretical error $|E_n - E_\infty|$.

dling of larger problems than vector-based approaches. As illustrated in Figure 5.7(b), in this application, the vector implementation shows an exponential growth of memory requirements, reaching the limitations in memory of our resources at the number of $n = 14$ qubits per dimension. In contrast, the MPS algorithms saturate at a modest memory size that fits in any computer, allowing us to explore sizes of up to $n = 15$ qubits per dimension—i.e., a $2^{30} \simeq 10^9$ points grid—using around 100 Mb of practical memory (code and data) in a desktop computer. Memory compression is related to the maximum bond dimension achieved by each method. The maximum bond dimension of the solution is bounded (Figure 5.8(a)), as in the 1D Gaussian in section 5.2. However, the maximum bond dimension reached during the solution depends on the operations of each method, as shown in Figure 5.8(b). This bond dimension is higher for a larger Krylov basis but still within a reasonable value for its implementation.

To complete this study, Figure 5.7(c) shows the error $|E_n - E_\infty|$ in the approximation of the exact analytical energy of the ground state (5.36), where E_n is the discrete solution with n qubits and E_∞ is the exact eigenvalue of the PDE in the continuum limit. The second-order finite

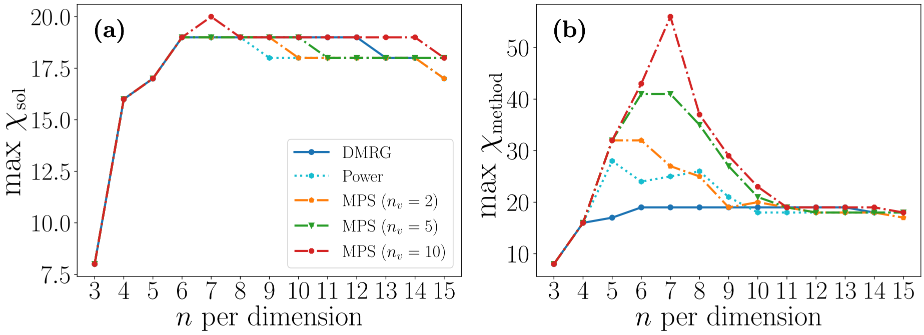


Figure 5.8: Maximum bond dimension scaling with the number of qubits n per dimension for the 2D squeezed harmonic oscillator interpolation experiments. (a) Maximum bond dimension of the solution for each method χ_{sol}. (b) Maximum bond dimension reached for each method χ_{method}.

difference formulas limit the accuracy of the methods. Their error scales algebraically with the discretization size $O(\Delta x^2)$ and thus exponentially with the number of qubits, which agrees with Figure 5.7(c). The plot also includes results obtained using the Lanczos library PRIMME [269]. However, neither this library nor all other vector methods could provide a solution for $n = 15$ qubits per dimension, which the MPS algorithms successfully obtained.

5.4 Conclusions and future perspectives

This chapter studies the solution of Hamiltonian PDEs using quantum-inspired algorithms, following the MPS-MPO representation described in sections 3.6 and 3.7. The analysis introduces new global optimization algorithms based on the limited-precision algebra in section 3.5, using two different techniques: imaginary-time evolution (Runge-Kutta methods) and approximate diagonalization (gradient descent, power iteration, and restarted Arnoldi iteration). These methods are benchmarked against state-of-the-art DMRG and Arnoldi vector-based implementations, exploring their asymptotic performance and practical resources in small and large problems.

The main conclusion is that MPS-based algorithms outperform vector-based algorithms, showing exponential advantages in memory and, thus, enabling grid sizes beyond the memory limitations of standard diagonalization algorithms. When considering current quantum hardware, the results also show that these methods perform better than the variational quantum algorithm in chapter 4, achieving better accuracy and avoiding issues such as barren plateaus, noise, and limited access to devices and also benefitting from an exponential advantage in memory with respect to the standard classical methods.

Among tensor network methods, imaginary-time evolution appears less efficient than a self-calibrated gradient descent. Furthermore, Arnoldi iteration significantly improves gradient descent, allowing a more stable implementation in the limited precision MPS algebra at a small cost. For PDEs that admit a small MPO representation, DMRG is excellent at optimizing the state, performing exponentially better than vector-based methods, even in time. When DMRG is not a suitable choice, Krylov subspace methods combined with interpolation can be applied to more complex MPOs, maintaining the exponential advantage in memory.

This work also paves the path for numerous avenues of optimization and generalization. Upgrading our state-of-the-art Python implementation, the SeeMPS [51] library, to a C++ implementation can lead to a 10x speedup according to preliminary tests. Then, MPS would be competitive with Arpack/PRIMME at even smaller sizes. Another possibility is to parallelize the construction of the Krylov basis and finite-precision algebra operations in the Arnoldi iteration. There are also possible improvements in these algorithms in terms of accuracy and stability, allowing their extension to more applications, such as diagonalization in other areas of the spectrum, computation of excited states, and the incorporation of symmetries. A possible improvement is substituting the finite difference method with a more precise spectral method based on Fourier interpolation [38] and an exponentially more efficient encoding of derivative operators [46]. This method requires the application of the QFT and its inverse MPO, and hence an MPO representation with a reasonable bond dimension scaling with the number of qubits, which may be implemented following the prescriptions in Ref. [148]. Other alternatives are pseudospectral methods, such as the Hermite Distributed Approximate Function (HDAF) [271], which can achieve spectral accuracy with a limited bond dimension, as it is shown for the evolution problem in chapter 6. These upgrades will allow us to address more complex PDEs, such as those present in superconducting circuits, where quantum-inspired algorithms may provide a suitable alternative to overcome the memory limitations of state-of-the-art techniques [150, 151, 152].

These quantum-inspired algorithms are also suitable for other PDEs, such as source PDEs (Appendix D), as long as the equation can be recast as an optimization problem. The good performance shown by these diagonalization methods motivates their application besides numerical analysis. One possibility is the study of many-body physics problems. These techniques can be combined with other methods designed for long-range interactions, such as Chebyshev expansions [272], the generalized TDVP algorithm [273, 274], the MPO $W^{I,II}$ method [249], and the variational uniform matrix product state (VUMPS) [275] algorithm, that combines the DMRG and MPS tangent space concepts. Other studies [276] indicate a possible link between these quantum-inspired techniques and machine learning in the context classification via physical neural networks based on d-level quantum systems (qudits).

Chapter 6

Quantum-inspired solution of time-dependent PDEs

Time-dependent partial differential equations (PDEs) play a fundamental role in Physics, modeling a vast range of phenomena across different areas. Prominent examples include the wave equation, the Schrödinger equation, and the Navier-Stokes equation. Each is a time-dependent PDE that governs essential processes such as wave propagation, quantum state evolution, and fluid dynamics, respectively. Encouraged by the MPS exponential advantages in chapter 5, this chapter introduces the quantum-inspired methods presented in Ref. [50] to address such equations, focusing on quantum mechanics's time evolution problem.

The solutions we developed are motivated by rather paradigmatic (and challenging) problems in the evolution of quantum systems. More precisely, we center the discussion on levitodynamics, a field within optomechanics that studies the control of levitated nano and micro-objects in vacuum [4]. Levitodynamics has experienced steady growth in the past decade [277, 4], driven by its unexploited potential for both fundamental research and practical applications, ranging from sensing and metrology to exploring novel quantum phenomena and complex systems [4]. Some practical experiments require an expansion of the particle's wavefunctions [278, 279, 280] by multiple orders of magnitude. This expansion imposes a severe constraint on the numerical methods used, which must be able to capture both the initial state of the macroscopic particle and the whole expansion. The MPS time evolution techniques developed in this chapter provide an asymptotic acceleration of the problem due to their associated memory compression.

In this study, we focus on the phenomenon of quantum quenching and explore two distinct potentials: a harmonic potential, which has a known analytic solution and serves as a benchmark, and a double-well potential commonly found in experimental settings. Following this, section 6.2 introduces a highly efficient and accurate MPO representation of functions of derivatives provided by a novel extension of the Hermite Distributing Approximating Functionals (HDAF) to the MPS/QTT formalism (Section 6.2). We combine this technique for approximating differential operators with a series of global time evolution algorithms for MPS to solve the time-dependent PDE. The explicit methods in section 5.1.1 and the restarted Arnoldi iteration in section 5.1.4 are adapted for real-time evolution. Two new methods are introduced: the implicit Crank-Nicolson and the trotterized split-step method. A numerical analysis of the quantum quench problem on up to 2^{20} points demonstrates that HDAF has greater accuracy than conventional finite difference techniques with similar implementation costs. Paired with the split-step method, it constitutes an efficient method to solve time-dependent PDEs with comparable error scaling and run time to Fast Fourier Transform (FFT) vector methods while offering a significant memory advantage.

6.1 Wavefunction's expansion of a levitated nanoparticle

Exploring quantum mechanics in the macroscopic world is an exciting new challenge that is becoming a reality in the laboratory [281]. Superconducting quantum computers, the Micius satellite [282] for quantum communication experiments at more than 1000 km and ETH's 30-meter quantum link [283] are just some advances of quantum technologies in larger scales. A particularly interesting research area is levitodynamics—see Ref. [4] for an extensive review on the topic—, which explores nano and microparticles levitated in a vacuum through optical or radiofrequency forces [277, 4]. Levitodynamics focuses on massive macroscopic objects, with a length scale of $10^{-4} - 10^{-7}$ m, enabling new research directions in sensing and metrology, physics of complex systems and novel quantum physics [4]. As stated in Ref. [4], the future of levitodynamics ranges from more practical applications, such as creating commercial sensors and inertial navigation systems, to the detection of astrophysical signals—gravitational waves [284], dark matter, and dark energy [285, 286, 287, 288, 289]—and the study of quantum gravity using the entanglement of two levitated particles via their mutual gravitational interaction [290, 291].

Levitodynamics offers the possibility to enhance our understanding of quantum superposition. The goal is to prepare macroscopic quantum superpositions of nanoparticles containing billions of atoms [4], which could open the path to new research lines like the creation of highly precise sensors [292, 293] and the study of the interplay of quantum mechanics and gravity for macroscopic quantum superposition states [294]. Generating macroscopic quantum superpositions requires highly pure states spread coherently over significant distances. These conditions can be achieved through ground-state motional cooling, followed by expanding the wavefunction via dynamic control of the trapping potential [295, 296, 297, 278, 279, 280, 298, 299]. This results in a spatial expansion of the wavefunction of up to five orders of magnitude, ideally reaching the particle's size ($\approx 10^{-7}\,\mathrm{m}$).

The quantum quench process is suitable for obtaining macroscopic quantum superposition in a levitodynamics context. In this process, an abrupt change in a quantum system's Hamiltonian drives the system out of equilibrium [300, 301]. This causes the system, which was originally in a stationary state of the initial Hamiltonian, to evolve according to the new Hamiltonian, reaching a new equilibrium or exhibiting complex dynamics. Going back to levitodynamics, to achieve such macroscopic quantum superposition states, the particle is initially ground-state cooled in a tight harmonic potential, and then, this potential is modified to a wide anharmonic potential [279, 280]. This causes the evolution of the particle, following a dramatically large wavefunction's expansion. Current experimental approaches have achieved expansions of a factor 24 [278], but future developments require expansions of several orders of magnitude [4]. Numerical simulations are essential for designing and understanding these experiments. However, the dramatic increase in the wavefunction's definition space complicates solving the evolution problem with standard PDE solvers, even for techniques tailored to this simulation [279]. This complexity arises due to memory constraints associated with the expanding space, increasing the necessary number of discretization points during evolution.

The challenge posed by studying macroscopic quantum superposition opens the venue for quantum-inspired MPS methods to solve it. This problem is governed by a time-dependent PDE, the Schrödinger equation

$$i\partial_t\psi(x,t) = H\psi(x,t) = \left(-\frac{\hbar^2}{2m}\partial_x^2 + V(x,t)\right)\psi(x,t), \tag{6.1}$$

which describes the particle's evolution during and after a quantum quench process. In experiments, we expect a sudden change in the strength of the potential that traps the nanoparticle. The trapping potential will be described as harmonic at a low energy limit. The quench then reflects in a discontinuous change of the trapping frequency from ω_0 to ω_H at time $t=0$ given by

$$V(x,t) = \begin{cases} \frac{1}{2}\omega_0^2 x^2, & t \leq 0, \\ \frac{1}{2}\omega_H^2 x^2, & t > 0. \end{cases} \tag{6.2}$$

Assuming the wavefunction at time $t=0$ starts in the ground state of the original trapping potential Hamiltonian,

$$\psi(x,t=0) = \left(\frac{\omega_0}{\pi}\right)^{1/4} \exp\left(-\frac{1}{2}\omega_0 x^2\right), \tag{6.3}$$

its evolution is prescribed according to

$$\psi(x,t) = \left(\frac{\omega(t)}{\pi}\right)^{1/4} \exp\left(-\left[\frac{\omega(t)}{2} + i\beta(t)\right] x^2\right), \tag{6.4}$$

$$\omega(t) = \omega_H \left(\frac{\omega_H}{\omega_0}\cos^2(\omega_H t) + \frac{\omega_0}{\omega_H}\sin^2(\omega_H t)\right)^{-1}, \tag{6.5}$$

$$\beta(t) = \frac{\omega(t)}{4}\left(\frac{\omega_H}{\omega_0} - \frac{\omega_0}{\omega_H}\right)\sin(2\omega_H t). \tag{6.6}$$

Appendix E shows the analytic derivation of equations (6.4)-(6.6).

Figure 6.1 depicts the harmonic potential (6.2) and its associated solution (6.4) for $\omega_0 = 1, \omega_H = 0.1$. For $\omega_H < \omega_0$, the potential increases its width (Figure 6.1(a)), leading to a consequent increase of the width of the solution shown in Figure 6.1(b). The solution (6.4) is a complex Gaussian with width $\sigma(t) = 1/\sqrt{\omega(t)}$ that describes a periodic process with period $T = \pi/\omega_H$. For $\omega_H < \omega_0$, the wavefunction expands during the first half of the period, going from the initial width $\sigma_0 = 1/\sqrt{\omega_0}$ at $t = 0$, which coincides with the minimum width $\sigma_{\min}$, to the maximum one $\sigma_{\max} = \sqrt{\omega_0}/\omega_H$ at time $t = 0.5\pi/\omega_H$. The expansion ratio is quantified by $\sigma_{\max}/\sigma_{\min} = \omega_0/\omega_H$. Thus, the frequency ratio dictates the total amplification of the wavefunction's spatial extent

The harmonic trap quench already poses interesting challenges, such as the wavefunction's size change by an arbitrary number of orders of magnitude. It is thus a useful problem to explore and calibrate the precision of the MPS techniques developed in sections 6.2 and 6.3 to address the time evolution problem. However, this wavefunction is not particularly interesting from an experimental point of view because, although the particle is delocalized, the state cannot be interpreted as a macroscopic superposition of two well-distinguishable states.

Let us consider anharmonic potentials, which are of interest for the creation of macroscopic quantum superpositions [4, 295, 296, 297, 302,

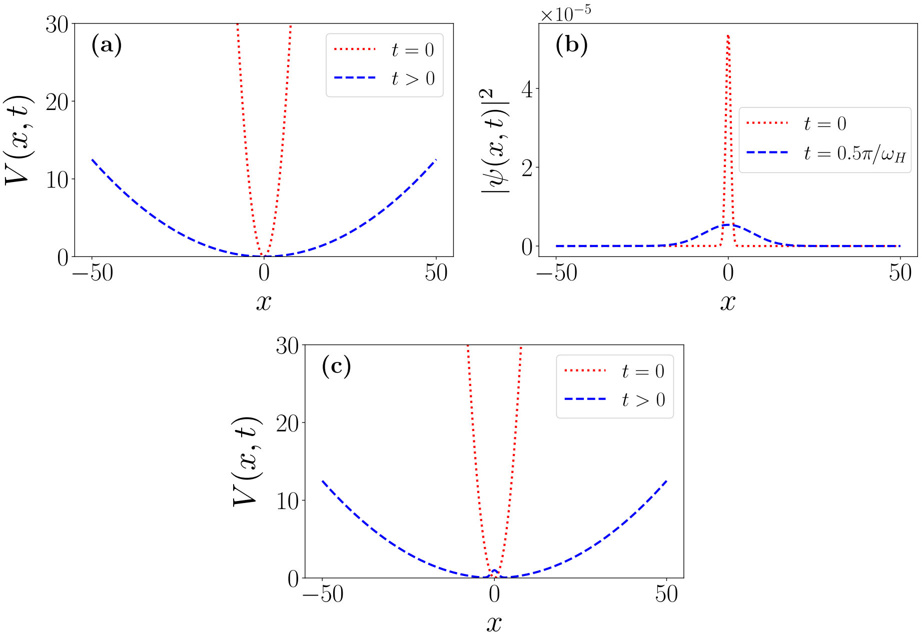


Figure 6.1: Harmonic and double-well potential quantum quench. (a) Harmonic potential (6.2) with $\omega_0 = 1$ and $\omega_H = 0.1$. (b) Harmonic quantum quench solution (6.4) with $\omega_0 = 1$ and $\omega_H = 0.1$. (c) Double-well potential (6.7) with $\omega_0 = 1$, $\omega_H = 0.1$, $u = 1$, and $\sigma = 1$.

303, 280, 278, 298, 299]. In this case, analytical solutions are usually unknown. This occurs, for example, in the case of a double-well potential that combines a harmonic term with an anharmonic Gaussian term

$$V(x,t) = \begin{cases} \frac{1}{2}\omega_0^2 x^2, & t \le 0, \\ \frac{1}{2}\omega_H^2 x^2 + u\exp(-x^2/2\sigma^2), & t > 0. \end{cases} \tag{6.7}$$

For small values of u, the harmonic expansion behavior dominates the wavefunction's evolution, with the Gaussian bump at $x = 0$ acting as a perturbation, as depicted in Figure 6.1(c). This allows us to use the results of the harmonic quench analysis as a basis for studying the double-well potential. When $u > 0$, the Gaussian part of the potential is expected to symmetrically separate the expanding wavefunction, causing the wavefunction's evolution to deviate from the Gaussian form (6.4) and form a state with a two-peaked probability density. Both in the harmonic and the double-well case (for a weak non-harmonic potential), the ratio of the frequencies determines the expansion of the wavefunction as $\sigma_{\max}/\sigma_{\min} = \omega_0/\omega_H$. In contrast, the Gaussian potential u and

σ parameters determine the barrier's width and height.

6.2 Hermite distributed approximating functionals for MPS

6.2.1 Distributed Approximating Functionals

Distributed approximating functionals (DAFs) [149, 304] are resolutions of the identity that can be used to accurately approximate operators. Hoffman originally introduced them to address the time-dependent Schrödinger (6.1) equation as a technique to approximate the free propagation of wave packets by a coarse-grained, highly banded matrix operator. Standard methods to represent the free propagator rely on the Fourier transform, transforming the propagator to momentum space, in which it is a diagonal operator. DAFs avoid extra transformations by only requiring a single representation of the problem, creating a sparse representation of the propagator in coordinate space. Due to their well-tempered property, DAFs have similar accuracy for on and off-grid points, having no "special points" [305]. This property makes DAFs an approximate identity kernel that has been successfully extended to approximate functions and their derivatives [305].

We will focus on the first and most used DAF: the Hermite DAF (HDAF). HDAF approximates a function as a sum of even Hermite polynomials weighted by a Gaussian function

$$\delta_M(x;\sigma) = \frac{\exp\left(\frac{-x^2}{2\sigma^2}\right)}{\sqrt{2\pi}\sigma} \sum_{m=0}^{M/2} \left(-\frac{1}{4}\right)^m \frac{1}{m!} H_{2m}\left(\frac{x}{\sqrt{2}\sigma}\right), \tag{6.8}$$

where $H_m(x) = (-1)^m e^{x^2} \frac{d^m}{dx^m} e^{-x^2}$ is the m-th Hermite polynomial. This expression depends on two terms, σ and M, which determine the width of the Gaussian filter and the maximum order of the Hermite sum, respectively. The HDAF kernel enables to approximate a function $f(x)$ as [305]

$$f(x) \approx \int dx'\, \delta_M(x - x'; \sigma) f(x'), \tag{6.9}$$

reproducing the Dirac delta distribution in the $\sigma/M \to 0$ limit.

Rewriting the HDAF formula (6.8) in terms of the analytic expression of the Hermite polynomials leads to

$$\delta_M(x;\sigma) = \frac{1}{\sqrt{2\pi}\sigma} \sum_{m=0}^{M/2} \frac{1}{m!} \left(-\frac{\sigma^2}{4}\right)^m \frac{\partial^{2m}}{\partial x^{2m}} \exp\left(-\frac{x^2}{2\sigma^2}\right). \tag{6.10}$$

This new form is useful for operating with the HDAF and obtaining different approximations. Since Eq. (6.10) is infinitely differentiable, it is possible to approximate the action of the derivative as

$$D\left[\frac{\partial}{\partial x}\right] f(x) \approx \int dx'\, D\left[\frac{\partial}{\partial x}\right] \delta_M(x-x') f(x'). \tag{6.11}$$

This expression leads to an approximation of the l-th order differential operator given by

$$\delta_M^{(l)}(x;\sigma) = \left(\frac{-1}{\sqrt{2}\sigma}\right)^l \frac{\exp\left(\frac{-x^2}{2\sigma^2}\right)}{\sqrt{2\pi}\sigma} \sum_{m=0}^{M/2} \left(-\frac{1}{4}\right)^m \frac{H_{2m+l}\left(\frac{x}{\sqrt{2}\sigma}\right)}{m!}. \tag{6.12}$$

An interesting concrete example of a differential operator is the free propagator $T(t) = e^{-\frac{it}{2}\partial_x^2}$. Applying $T(t)$ to Eq. (6.10) results in an approximation to the propagator, where the kernel is

$$\begin{aligned} \tilde{\delta}_M(x-x';\sigma,t) &= T(t)\delta_M(x-x';\sigma) \\ &= \frac{\exp\left(\frac{-(x-x')^2}{2\sigma_t^2}\right)}{\sqrt{2\pi}\sigma_t} \sum_{m=0}^{M/2} \left(\frac{-\sigma^2}{4\sigma_t^2}\right)^m \frac{H_{2m}\left(\frac{x-x'}{\sqrt{2}\sigma_t}\right)}{m!}. \end{aligned} \tag{6.13}$$

The action of the propagator on the HDAF kernel spreads the original Gaussian with variance σ^2 to a new Gaussian with variance $\sigma_t^2 = \sigma^2 + it$

Expressions derived from the resolution of the identity provided by HDAF maintain a well-tempered nature with similar accuracy as long as their Fourier transform lies under the "HDAF plateau", i.e., the eigenvalues of the Fourier transform of Eq. (6.8), $\lambda_{M,\sigma}$ verify $0 \le \lambda_{M,\sigma} \le 1$ [306].

Unlike the Delta function, the HDAF kernel is generally a bandwidth-limited, infinitely smooth function and hence can be used for the discretized approximation of functions and operators. We use the midpoint rule quadrature to define a discretized version of HDAF that, given a function $f(x_k)$ defined on a discrete set of points x_k, accurately reconstructs a function $f(x), -\infty < x < \infty$ [305]. In the event of an equal grid spacing Δx and a kernel function of the distance between points, the HDAF approximation becomes

$$f_{\text{app}}(x) = \sum_{k=0}^{N-1} \Delta x \delta_M(x-x_k;\sigma) f(x_k). \tag{6.14}$$

In this case, the HDAF operator turns into a Toeplitz matrix, i.e., a diagonal-constant matrix. This enables an efficient implementation, only storing N values, where $2N-1$ is the total bandwidth. Moreover, Eq. (6.14) becomes a convolution, and its computation in momentum space reduces to the multiplication of a diagonal matrix due to HDAF's Toeplitz nature [305]. In addition, due to the decaying nature of the filter, practical applications require a limited number of points, leading to a highly banded structure of the HDAF kernel. This banded structure is controlled by $\sigma/\Delta x$, but it cannot be arbitrarily decreased since a minimal number of diagonals is needed to discretize the integral accurately.

Similarly to Eq. (6.8), the differential operator approximation (6.12) and the propagator in (6.13) are Toeplitz matrices with a highly banded structure. This leads to a representation of the derivative in position space with a similar structure and application to the standard finite difference technique, avoiding the complexity of the Fourier spectral method while achieving pseudospectral accuracy.

6.2.2 MPO representation

The approximations of the identity and functions of derivatives provided by HDAF are Toeplitz matrices and, hence, suitable for an MPO representation as weighted displacements of the quantum register. The corresponding MPOs are

$$\hat{K} = \Delta x \delta_M(0;\sigma)\mathbb{I} + \sum_{i=1}^{2^n-1} \Delta x \delta_M(i\Delta x;\sigma)\left(\hat{\Sigma}^{+i} + \hat{\Sigma}^{-i}\right), \tag{6.15}$$

$$\hat{K}^{(l)} = \Delta x \delta_M^{(l)}(0;\sigma)\mathbb{I} + \sum_{i=1}^{2^n-1} \Delta x \delta_M^{(l)}(i\Delta x;\sigma)\left(\hat{\Sigma}^{+i} + (-1)^l\hat{\Sigma}^{-i}\right), \tag{6.16}$$

$$\hat{K}_t = \Delta x \tilde{\delta}_M(0;\sigma,t)\mathbb{I} + \sum_{i=1}^{2^n-1} \Delta x \tilde{\delta}_M(i\Delta x;\sigma,t)\left(\hat{\Sigma}^{+i} + \hat{\Sigma}^{-i}\right), \tag{6.17}$$

where the symmetry or antisymmetry of $\delta_M^{(l)}$ has been used, where l is even or odd, respectively.

6.2.3 Heuristic parameter estimation

Carefully tuning the HDAF's parameters is key to obtaining an efficient and accurate approximation. The errors of the HDAF approximation have two main sources: the approximation of the function $f(x)$ by a

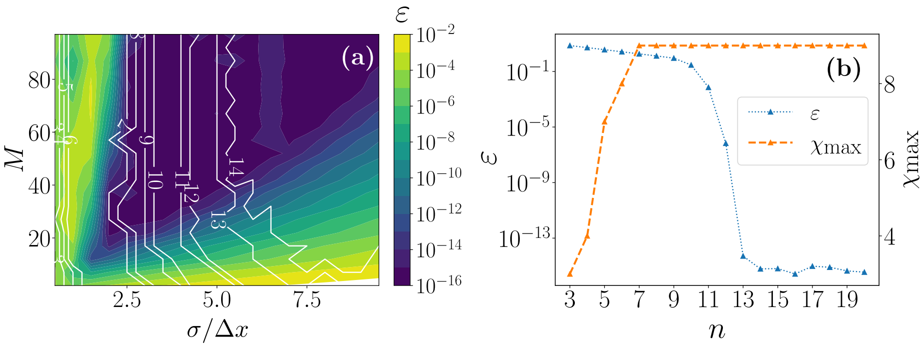


Figure 6.2: HDAF approximation of a normalized Gaussian function. The initial Gaussian has a standard deviation of one and zero mean and is defined in an interval $L = 16\sigma_{\text{max}}$. (a) Error ε (colormap) and bond dimension (white lines) of the HDAF approximation as a function of the maximum degree of the Hermite polynomial M and the ratio $\sigma/\Delta x$ for $n = 14$. (b) Error ε and maximum bond dimension χ_{max} scaling with the number of qubits for $M = 42$ and $\sigma/\Delta x = 3$.

Hermite polynomial of order M under the extent of a Gaussian envelope and the midpoint rule to approximate the infinite sum. While the HDAF approximates the Delta distribution for $\sigma/M \to 0$, it is not possible in practice to indefinitely increase M or decrease σ. For a more oscillatory integrand, a higher value of $\sigma/\Delta x$ will be needed for the Gaussian envelope to cover an adequate number of nodes and attain satisfactory integration accuracy. For a fixed and sufficiently large value of M, it is possible to estimate the value of σ as [50]

$$\sigma = \max(\sigma_M, \sigma_{\text{min}}, \sqrt{t}). \tag{6.18}$$

The first term,

$$\sigma_M = \frac{\Delta x}{\sqrt{2\pi}} \sum_{m=0}^{M/2} \left(\frac{-1}{4}\right)^m \frac{H_{2m}(0)}{m!}, \tag{6.19}$$

makes the reconstruction optimal for a fixed M, since perfect reconstruction is achieved when the M zeros of the HDAF match the grid's zeros, with only the origin term contributing [304]. The second term, $\sigma_{\text{min}} = 3\Delta x$, is heuristically set to ensure convergence of the quadrature's midpoint rule. Finally, $\sqrt{t}$ minimizes the spatial extent of the effective variance of the propagator.

Let us heuristically study the practical accuracy of the HDAF method for approximating a normalized Gaussian function with a standard deviation of one and zero mean. Figure 6.2(a) shows the approximation error

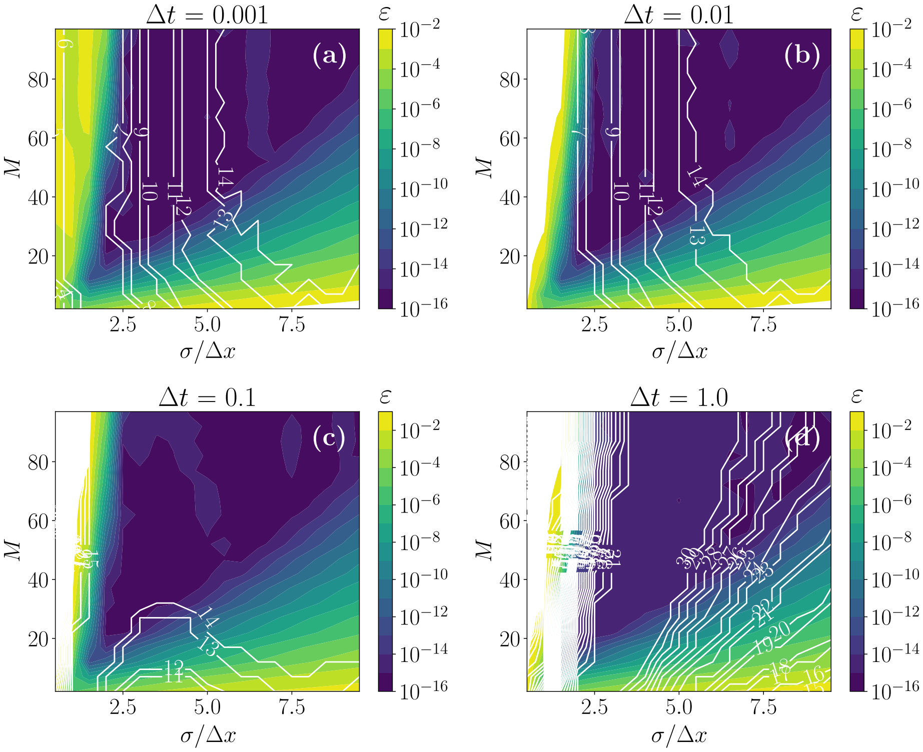


Figure 6.3: HDAF approximation of a one-step evolution under a harmonic oscillator potential of a normalized Gaussian function. The initial Gaussian has a standard deviation of one and zero mean and is defined for a fixed number of qubits $n = 14$ in an interval $L = 16\sigma_{\max}$. The harmonic oscillator for the evolution has parameters $m = \hbar = 1$ and $\omega = 0.01$. (a) $\Delta t = 0.001$, (b) $\Delta t = 0.01$, (c) $\Delta t = 0.1$ (d) $\Delta t = 1.0$.

$\varepsilon = \sqrt{\sum_i |f_{\text{app}}(x_i) - f(x_i)|^2 \Delta x}$ and the maximum bond dimension of the simplified HDAF MPO $\chi_{\max}$ as a function of the HDAF parameters—M and $\sigma/\Delta x$—for a fixed number of qubits $n = 14$, for which the method has already converged. As the value of $\sigma/\Delta x$ increases, so does the accuracy until convergence for $\sigma/\Delta x \gtrapprox 2.5$ and a relatively small number of $M \approx 30$, since for these values, the errors of the quadrature are avoided. The increase of $\sigma/\Delta x$ leads to a less banded matrix and, consequently, a potentially larger maximum bond dimension for a sufficiently large M. In practice, a value of $M \approx 40$ and $\sigma/\Delta x = 3$ is enough to achieve spectral accuracy with a significantly reduced bond dimension $\chi_{\max} = 9$. For fixed values $M = 42$ and $\sigma/\Delta x = 3$, the error ε and the bond dimension $\chi_{\max}$ converge as the number of qubits increases, as depicted

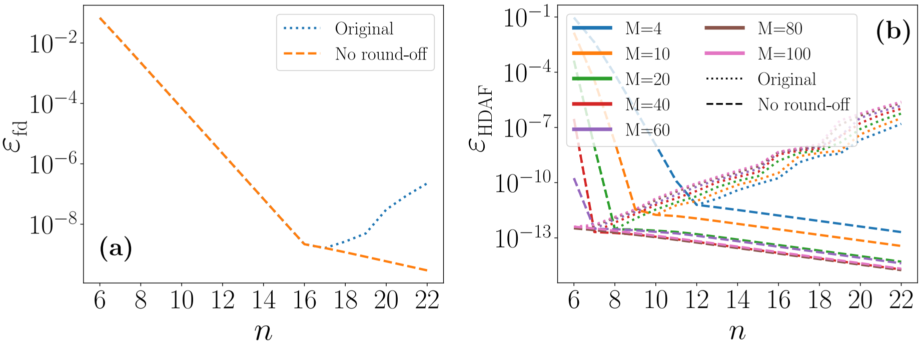


Figure 6.4: Error ε in the second derivative approximation of a Gaussian function with standard deviation one and zero mean for varying qubits. The results consider approximations with and without round-off error correction. (a) Finite differences. (b) HDAF.

in Figure 6.2(b). Thus, there is a minimum discretization for which the HDAF approximation achieves its maximum accuracy, which saturates the bond dimension, allowing the approximation of functions on finer grids without increasing the cost.

Figure 6.3 illustrates that the error profile is similar when considering the time evolution under a harmonic oscillator potential (6.2) with $m = \hbar = 1$ and $\omega_H = 0.01$ for $\Delta t = 0.001, 0.01$. As the time step increases, so does the bond dimension, modifying its behavior and leading to a larger bond dimension for smaller $\sigma/\Delta x$ for $\Delta t = 1$, and a stable value for the optimum error area. Still, the bond dimension is amenable for practical implementations, allowing for large time steps while maintaining machine precision accuracy

In addition to the mentioned errors, the round-off error in standard differentiation techniques is also found in the HDAF approximation. This error bounds the accuracy of differentiation methods since it establishes the minimum grid spacing approximable and the accuracy determined by the truncation errors of the method. The MPO implementation of both the finite difference and HDAF approximation of derivatives provides a very efficient manner to deal with this issue. Once the minimum Δx is achieved, the MPO can be extended to denser grids by adding extra idle sites, maintaining the accuracy without increasing the complexity of the operator.

Figure 6.4 shows the accuracy of the finite difference and HDAF differentiation with and without round-off error correction for approximating the second derivative of a Gaussian function with a standard deviation of one and a mean zero. Both finite difference and HDAF are affected by significant round-off errors. When the method obtains maximum accuracy, the number of qubits can be fixed to correct the round-off. Then, the error is only reduced due to the decrease of Δx, since the norm-2 different error depends on $\Delta x^{1/2}$. HDAF achieves faster convergence than finite difference, increasing convergence speed with M. For $M = 60$, the error is larger than for $M = 40$ due to the heuristic cutoff $\sigma/\Delta x = 3$, and a slightly smaller error could be achieved with more precise fine-tuning.

6.3 Time evolution algorithms

Section 6.1 introduced the problem of the time evolution of a levitated particle. The Schrödinger equation (6.1) governs this evolution, and its formal solution can be expressed as the repeated action of a possibly time-dependent unitary operator $U(t)$ on an initial state $\psi(x, t = 0)$,

$$\psi(x,t) = U(t)\psi(x,0) = e^{-iHt}\psi(x,0). \tag{6.20}$$

The Schrödinger equation is a time-dependent PDE, and MPS and MPO can encode this problem to tackle it from a quantum-inspired perspective. When dealing with time-dependent PDEs, quantum dynamics simulations can be performed using tensor networks methods that combine Fourier techniques with Trotter expansion [307, 308], or Chebyshev propagation schemes [309]. Other methods involve employing one-step implicit time integration with an ALS-type solver or a global space-time formulation to address multi-dimensional parabolic problems [310]. These quantum-inspired techniques have been expanded to other domains, for example, kinetic simulations of collisionless plasmas through the Vlasov-Poisson equation [311].

To address the quantum quench problem in section 6.1, the PDE operators require global evolution schemes independent of the locality of interactions. This section presents a selection of time-evolution methods with an MPO-MPS implementation: explicit—Euler, Improved Euler, and fourth-order Runge-Kutta methods—and implicit Runge-Kutta methods, restarted Arnoldi iteration, and the split-step method. All previous methods are suitable for a finite difference and HDAF approximation of the differential operator. HDAF's propagator approximation also enables the split-step method by approximating the unitary operator $U(t)$ for brief periods of time. The n-site MPS/QTT representation of both the Hamiltonian H or the unitary operator $U(t)$ enables accessing exponentially dense grids with 2^n discretization points in space, which may lead to advantages regarding the vector representation.

6.3.1 Runge-Kutta methods

Runge-Kutta methods use the Taylor expansion of the state's dependence on time, up to p-th order around a given moment t_k, to approximate its time evolution with a local error, i.e., one-step error, that scales algebraically with the expansion order as $O(\Delta x^{p+1})$.

Section 5.1.1 provides an overview of the most extended explicit methods in an imaginary-time context. Substituting the imaginary time β by the real-time t using $\beta = it$ leads to the real-time evolution algorithms. As an example, the Euler method would be

$$\begin{aligned} \psi_0 &= \psi(\beta_0), \\ \psi_{k+1} &= \psi_k - i\Delta t H \psi_k, \quad \text{for } k = 0, 1, \ldots, N-1. \end{aligned} \tag{6.21}$$

In addition to explicit methods, implicit methods should be considered since they may increase stability. Our studies include the Crank-Nicolson method, a second-order algorithm based on the trapezoidal rule that combines the Euler method and its backward version evaluated on the k and $k+1$ iterations, respectively. Thus, the state at the $k+1$ iteration is approximated as

$$\left(\mathbb{I} + \frac{i\Delta t}{2} H\right) \psi_{k+1} = \left(\mathbb{I} - \frac{i\Delta t}{2} H\right) \psi_k. \tag{6.22}$$

Matrix inversion methods may solve the system of equations in its matrix-vector implementation. In the MPS-MPO framework, adaptations of techniques like conjugate gradient descent can approach this problem.

6.3.2 Restarted Arnoldi iteration

Adapting the restarted Arnoldi iteration to approximate the evolution operator on a Krylov basis for a brief period of time is possible. Section 5.1.4 describes the original method, which constructs a Krylov basis $\{v_i\}_{i=1,\ldots,n_v}$ of n_v elements and computes the matrices of the expectation value of the operator H and its norm—A and N, whose matrix elements are $\langle v_i|H|v_j\rangle$ and $\langle v_i|v_j\rangle$, respectively— to find the minimum eigenvalue of H. These matrices are now used to approximate the exact exponential evolution as

$$\psi_{k+1} = e^{-i\Delta t N^{-1} A} \psi_k, \tag{6.23}$$

where ψ_k is expressed using the Krylov basis. The number of Krylov vectors limits the error of the approximation of the exponential operator, which scales with $O(\Delta t^{n_v})$. Therefore, even a few vectors—$n_v = 5, 10$—can provide a highly accurate approximation. Since only a small number of vectors is needed, the matrix inversion and exponentiation cost in Eq. (6.23) is reduced compared to the MPO-MPS operations. Indeed, the number of vectors for an accurate approximation remains constant with system size, avoiding exponential scaling.

6.3.3 Split-step method

Split-step methods rely on an approximate decomposition of the Hamiltonian exponential into a series of exponentials that can be computed efficiently. The first-order method uses the Lie-Trotter product formula to approximate the evolution operator as

$$U(\Delta t) \approx e^{-i\Delta t D(-\partial_x^2)} e^{-i\Delta t V(x)}, \tag{6.24}$$

with an order two error in time $O(\Delta t^2)$. Higher order expansions, such as the Suzuki-Trotter formulas [312, 313], decrease the error scaling with the time step. The third-order decomposition provides a common approximation given by

$$\begin{aligned} e^{-i\Delta t(D(-\partial_x^2)+V(x))} \approx\ & e^{-i\Delta t V(x)/2} e^{-i\Delta t D(-\partial_x^2)} e^{-i\Delta t V(x)/2} \\ & + O(\Delta t^3). \end{aligned} \tag{6.25}$$

While a third-order error commonly suffices for practical applications, higher-order schemes can be constructed at the expense of introducing a larger number of exponential operators [314], increasing the implementation cost.

The split-step (6.25) algorithm usually resorts to the Fourier spectral method to approximate the action of the free propagator $\exp(-i\Delta t D(-\partial_x^2))$, but the HDAF formalism permits the efficient alternative of applying the free-propagator approximation (6.13) directly in the coordinate representation. This avoids the extra application of two operators: the Fourier transform and its inverse.

The potential propagator $\exp(-i\frac{\Delta t}{2}V(x))$ is diagonal in the coordinate basis. When no exact representation is available, it can be efficiently approximated using TT-cross interpolation as discussed in Ref. [52]. This technique is more efficient than series expansion techniques, such as the Chebyshev expansion, in this context [52].

6.3.4 One-step study

To approximate the evolution of the quantum state $\psi(x,t)$, the previous methods are iteratively applied over small time intervals Δt. This study enables determining the most advantageous methods while avoiding the high cost of performing the complete evolution for expansion ratios of interest, which may involve thousands of steps.

As it has been previously advanced, the evolution of a quenched state under a harmonic potential (6.2) has a known analytic solution (6.4), which makes it a suitable benchmark for the methods before moving onto potentials of experimental interest. This problem enables the estimation of errors in the wavefunction for different algorithms, differentiation methods, grids, and time steps.

We study the expansion of a quantum state as we reduce the trapping frequency of the harmonic potential by a factor $\omega_H/\omega_0 = 0.01$. This results in a 100-fold quantum state expansion, increasing its standard deviation from σ_0 to $\sigma_{\max} = 100\sigma_0$. The problem is defined for the domain $x \in [-L/2, L/2)$ with $L = 16\sigma_{\max}$ to capture the maximum expanded wavefunction. The initial state is derived from a tightly confined potential. Hence, the initial wavefunction is narrowly concentrated around $x = 0$. This confinement sets a lower bound for the grid discretization and number of qubits to represent the initial and final state accurately. It is essential to consider that loading a function that is mostly zero outside a narrow interval is challenging for MPS, requiring us to pad the function with zeros to compensate for the possible sampling or representation errors of TT-Cross or Chebyshev methods on the tails of the exponential.

This benchmark focuses on the accuracy and performance of the time evolution algorithms in section 6.3 measured using three figures of merit: (i) the function norm-2 difference

$$\varepsilon = \sqrt{\sum_i |\psi(x_i,\Delta t) - \tilde{\psi}(x_i,\Delta t)|^2 \Delta x}, \tag{6.26}$$

measures the accuracy of the methods—where $\psi(x_i,\Delta t)$ is the analytic solution for $t = \Delta t$, and $\tilde{\psi}(x_i,\Delta t)$ is the one-step state approximated by the method—, (ii) the run time, and (iii) the maximum bond dimension $\chi_{\max}$.

The solution of time-dependent PDEs involves two discretizations —position x and time t—that must be independently analyzed. For a fixed domain, the number of qubits, i.e., points, determines the spatial discretization. Section 6.2.3 already compares the performance of the finite difference and HDAF approximation of derivatives, demonstrating the exponential advantage of the error scaling with the number of qubits for the latter. To test if this performance extends to the solution of PDEs, the methods use the finite difference approximation with filter nine in Ref. [315] to enhance noise suppression, and the HDAF MPO with $M = 40$ and σ computed using Eq. (6.18). Both techniques use periodic boundary conditions, with the operators defined on a grid with $n = 18$ qubits (262144 points).

For a fixed spatial discretization size Δx, the errors in approximating the differential operators remain constant. This allows us to compare the performance of the finite difference and HDAF techniques for time evolution. The trade-off in the error ε (6.26) and the run time scalings with Δt for both implementations of the methods in section 6.3 determines the most efficient differentiation technique. Figure 6.5(a) illustrates that when using HDAF to encode the Hamiltonian operator, the error converges to the scaling predicted by each method for larger time steps, where the MPS truncation error is negligible compared to the time evolution algorithm's error. For shorter times, the MPS accuracy limits the performance, leading to a maximum accuracy plateau, with an error similar to the one imposed by finite precision in standard vector algorithms. In contrast, the truncation error inherent in the finite difference method restricts the accuracy, resulting in an error that exceeds the HDAF plateau, rendering the finite difference method less accurate for the same grid resolution. Appendix F provides a detailed analysis of the numerical scaling of ε with Δt for these numerical simulations.

The bond dimension of the states and operators determines the cost of the contractions, which dominates the MPS algorithms' run time. Figures 6.5(c)-(d) depicts the run time scaling with Δt, which is similar for both finite difference and HDAF approximations due to the comparable bond dimension of both differentiation techniques. Considering the exponential improvement in accuracy for the harmonic quantum quench's benchmark example and the similar cost, HDAF constitutes a more advantageous technique for approximating derivatives than the finite difference method. From now on, we will focus on the HDAF implementation of the PDE numerical methods.

The election of the most efficient time evolution method requires finding the balance between accuracy and run time. The method's error scales algebraically in Δt, with an increasing exponent with the method's order. On the other hand, high-order methods require more operations and, hence, larger run times. When using MPS for their implementa-

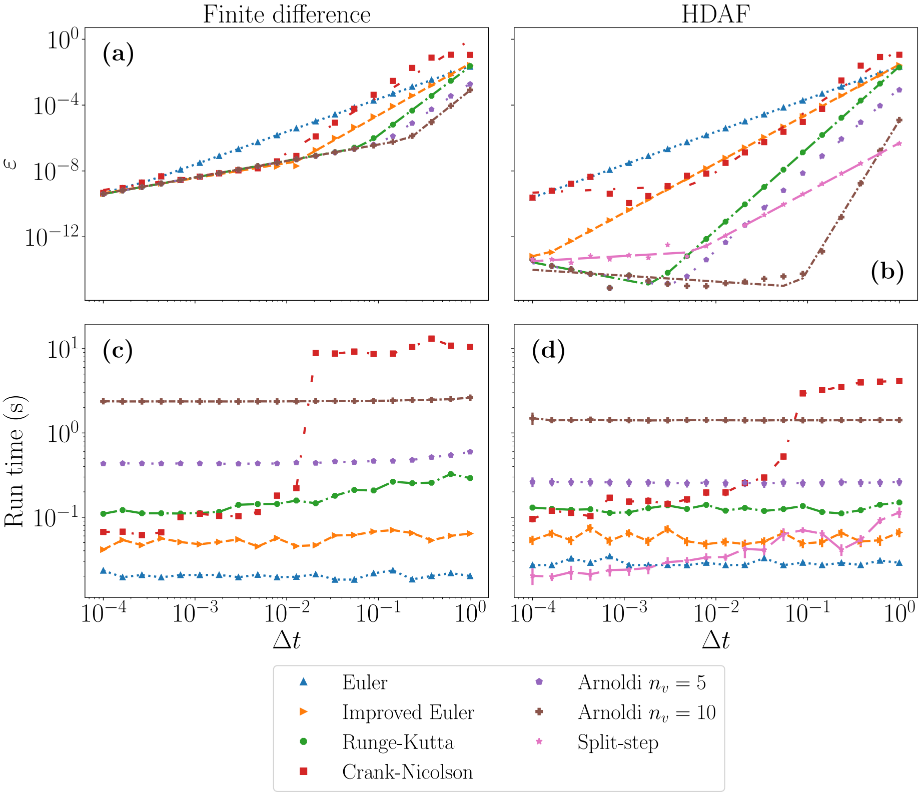


Figure 6.5: Harmonic quantum quench one-step evolution for a range of Δt and a fixed number of qubits $n = 18$. (a) Error ε (finite difference). (b) Error ε (HDAF). (c) Run time (finite difference). (d) Run time (HDAF). The run time is averaged over ten runs.

tion, the impact of the MPS truncation should be considered since this error can accumulate with the successive MPS-MPO operations and consequent simplifications, which may lead to high-order methods being less accurate than their vector counterparts. To avoid this issue, our experiments used a low truncation tolerance that enabled similar accuracies to the vector implementation with exponential memory savings, as it will be later discussed in Figures 6.7 and 6.6. As a consequence of the tolerance chosen, the highest-order method, i.e., the Arnoldi $n_v = 10$ method, shows the best performance in accuracy, enabling larger step sizes. Other lower-order methods— fourth-order Runge-Kutta, Arnoldi $n_v = 5$, and split-step–, while less precise, still reach similar accuracies for a considerable Δt range. The smaller order of the Euler method and

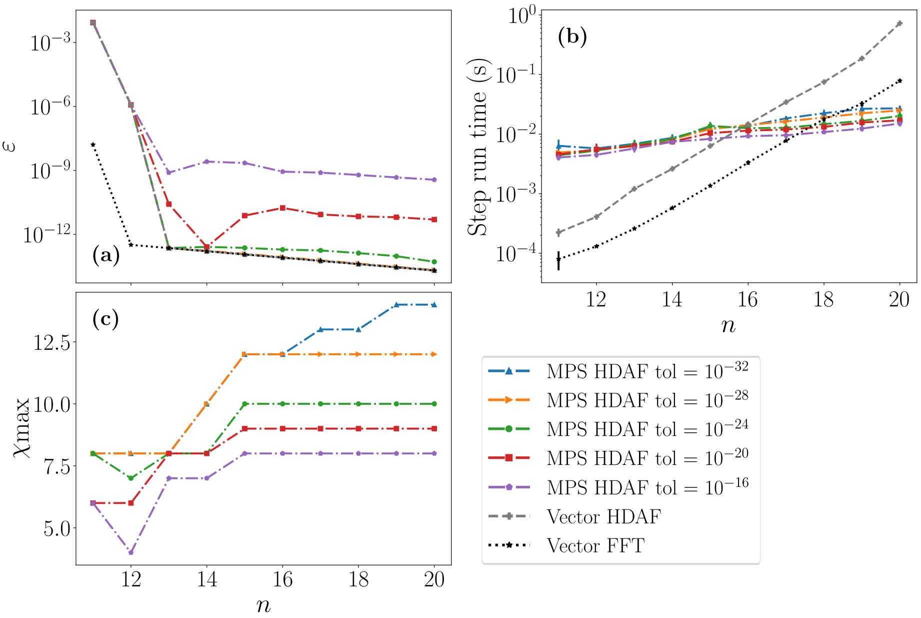


Figure 6.6: Number of qubits n scaling of the split-step one-step evolution for vector-based—HDAF and FFT— and different tolerance MPS-based HDAF for $\Delta t = 0.0001$. (a) Error ε. (b) Run time. (c) Maximum bond dimension $\chi_{\max}$. The run time is averaged over ten runs.

the conjugate gradient implementation in the Crank-Nicolson method restrict the error obtained beyond the intrinsic MPS truncation error.

Although the Arnoldi method with $n_v = 10$ achieves the best error scaling, its run time is significantly slower—by two orders of magnitude—compared to the split-step method. As a result, even with smaller time steps Δt, the split-step method offers a more favorable trade-off between computational cost and accuracy. Moreover, extremely high accuracy may not be required in practical applications, allowing for larger time steps. Therefore, the split-step method with the HDAF approximation of the propagator is the optimal choice for studying the expansion problem.

Having identified split-step as the most efficient MPS method, we compare it with two vector-based alternatives: the HDAF split-step vector implementation and the standard fast Fourier transform (FFT) split-step. Figure 6.6 analyzes the scaling of split-step MPS and vector-based implementations with the discretization size for a fixed time step Δt. The MPS method considers different values of the truncation tolerance for the SVD and the simplifications of the finite precision algebra, mea-

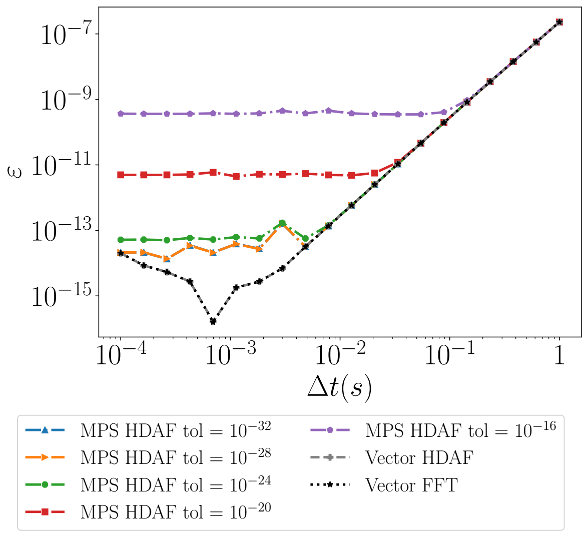


Figure 6.7: Error ε scaling with time step Δt for the split-step one-step evolution for vector-based—HDAF and FFT— and different tolerance MPS-based HDAF for $n = 20$.

sured as the norm-2 difference $\|\psi - \phi\|^2$ between the original state $|\psi\rangle$ with bond dimension χ_ψ and the one projected in the subspace of MPS $|\phi\rangle$ with bond dimension χ_ϕ, MPS_{χ_ϕ}, such that $\chi_\phi < \chi_\psi$.

First, Figure 6.6(a) shows the error ε scaling, demonstrating that the MPS tolerance determines the split-step accuracy. The method achieves numerical precision comparable to the vector implementation for tolerances of order $O(10^{-28})$ or smaller. Moreover, the error scaling shows that a discretization with $\Delta x \approx 10^{-1}$ suffices for HDAF to converge well above the finite difference approach. When considering the run time (Figure 6.6)(b)), the FFT is more efficient than the vector-based HDAF, and MPS perform asymptotically better than vector approaches, as they present an exponential scaling of time with the number of qubits due to the exponential increase in the number of points. The rise in tolerance slightly increases the run time for the MPS implementation while maintaining a similar scaling for a one-step implementation. Previous and forthcoming simulations use a tolerance of 10^{-28} for the SVD truncation, while a tolerance of $O(10^{-16})$ for the simplification algorithm. Finally, Figure 6.7 shows that the MPS error ε scaling with the time step Δt presents a similar behavior to the vector-based implementation for tolerances smaller than 10^{-28}. Increasing tolerances limit the accuracy for smaller time steps. As the time step grows, so does the truncation of the methods, allowing larger tolerances.

6.4 Quantum quench evolution

Section 6.3.4 compared the performance of a selection of MPS methods for solving time-dependent PDEs. More concretely, it benchmarked the one-step time evolution of a particle in a harmonic trap experiencing a sudden weakening of its frequency, whose analytic solution is known. The split-step method combined with the HDAF approximation of the propagator demonstrated the best trade-off between accuracy and implementation cost, motivating its use in problems of physical interest. This section shows the utility of this method for the complete evolution of a particle in a quantum quench process under various potentials. First, section 6.4.1 extends the one-step harmonic potential simulations to the complete expansion of the wavefunction. The known analytic solution enables the assessment of the method's accuracy as the successive steps accumulate their error, both for MPS and vector implementations. Then, section 6.4.2 introduces a new potential, the double-well potential presented in section 6.1.

6.4.1 Harmonic expansion

First, let us address the complete expansion of the wavefunction of a particle in a harmonic potential (6.2) that is 100 times weaker than the trap that confines the initial wavepacket. The simulation results in a 100-fold expansion of the wavefunction, accurately modeled using a discretization using $2^{20} = 1048576$ points or $n = 20$ qubits for $x \in [L/2, L/2)$ with $L = 16\sigma_{\max}$. Initially, the quantum state is in the ground state of the Hamiltonian for $t = 0$ with frequency $\omega_0 = 1$. This corresponds to a real-valued Gaussian function tightly localized around $x = 0$. Following a sudden potential weakening, the trapping frequency decreases to $\omega_H = 0.01$, causing the harmonic trap to relax. The state evolves according to the new potential, leading to an expansion of the wavepacket until $t_f = 0.5\pi/\omega_H$, at which the Gaussian solution (6.4) reaches its maximum width. Let us consider three values for the time steps $\Delta t = 0.01, 0.1, 1$ and a final time $t_f = 158$, so it fits all time steps.

Figures 6.8(a)-(b) depict the scaling of the error ε (6.26) with the time t of the evolution for the MPS HDAF and vector FFT implementations of the split-step method, respectively. It is worth noting that regardless of the implementation, the methods exhibit similar algebraic

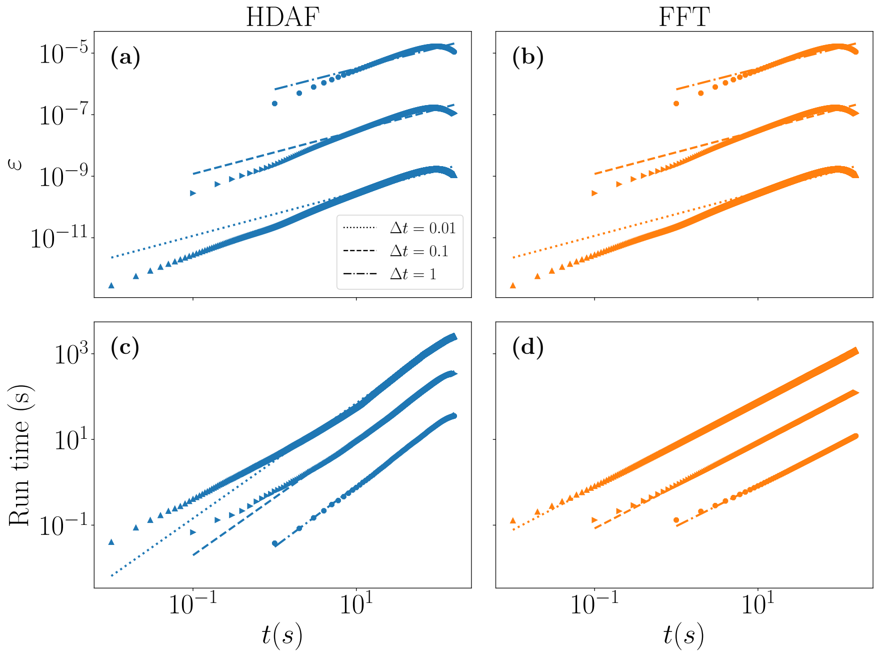


Figure 6.8: Particle's wavefunction expansion in a harmonic quantum quench (6.2). The simulation uses the MPS split-step methods with HDAF differentiation and a state-of-the-art FFT split-step to solve this problem with a frequency ratio $\omega_H/\omega_0 = 0.01$, final time $t_f \approx 0.5\pi/\omega_H = 158$ and $n = 20$. Figures (a)-(c) show the error scaling with the evolution time, and figures (d)-(f) show the run time scaling with the evolution time.

scalings (see Appendix G), which can be attributed to the expected linear accumulation from a symplectic integration algorithm on a periodic Hamiltonian system [316]. This shows that the error accumulation of the MPS method is negligible compared to the vector FFT split-step.

Figures 6.8(c)-(d) gauge the implementation cost as the run time scaling with the evolution time t. This scaling is close to linear for the FFT algorithms, as the problem size fixes the cost of each step, and the total run time is the sum of individual evolution steps. For the MPS implementation, the growth of the bond dimension of the expanding wavefunction leads to a scaling slightly above linear. This increase in bond dimension results in higher memory requirements and costs for the MPS and MPO operations than the one-step example.

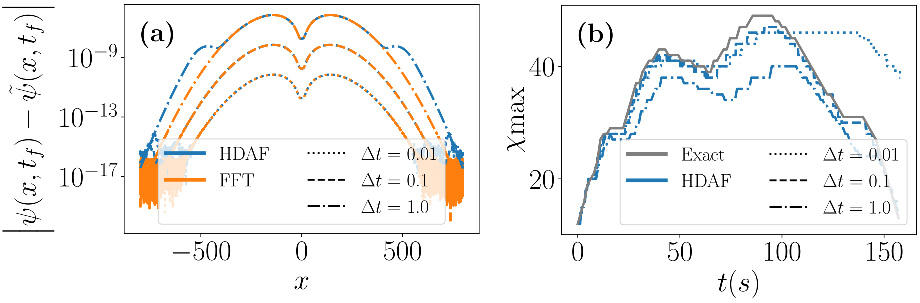


Figure 6.9: Pointwise error and maximum bond dimension $\chi_{\max}$ of a particle's wavefunction expansion in a harmonic quantum quench (6.2). The simulation uses the split-step methods with HDAF differentiation and a state-of-the-art FFT split-step to solve this problem with a frequency ratio $\omega_H/\omega_0 = 0.01$, final time $t_f \approx 0.5\pi/\omega_H = 158$ and $n = 20$. (a) Pointwise error $|\psi(x,t) - \tilde{\psi}(x,t)|$ of the maximum width solution approximated by the methods $\tilde{\psi}(x,t)$ with respect to the analytic solution $\psi(x,t)$ (6.4). (b) Maximum bond dimension $\chi_{\max}$ for each time step.

Figure 6.9(b) studies the evolution of the maximum bond dimension $\chi_{\max}$ with time t. The chirping of the wavefunction leads to an increase in bond dimension, which reaches the minimum values for the initial and final times, for which the imaginary phase is zero (or almost zero since $t \approx 0.5\pi/\omega_H$), i.e., the state is a real Gaussian. The numerical methods show similar behavior, with the bond dimension increasing with the absolute value of the phase $\beta(t)$ of the analytic solution (6.4). Larger time steps lead to lower bond dimensions due to the larger error of the approximation. Still, these methods can accurately reproduce the evolution with a reduced memory cost. This chirping is inherent to the physical setting; hence, the numerical methods may not present this behavior for alternative applications.

In Figure 6.9(a), the pointwise error of the maximum width state for the particle expansion is depicted as $|\psi(x,t) - \tilde{\psi}(x,t)|$. Here, $\tilde{\psi}(x,t)$ represents the approximate solution obtained through numerical methods, and $\psi(x,t)$ is the analytic solution (6.4). Both versions of the split-step method exhibit similar error patterns, with differences observed at the boundaries of the interval for the larger step size, potentially due to errors associated with the MPS representation. Nevertheless, Figure 6.8 shows that the total error ε of the memory-compressed solutions obtained using the MPS HDAF split-step method meets the high accuracy of the vector spectral method while achieving similar run times and avoiding

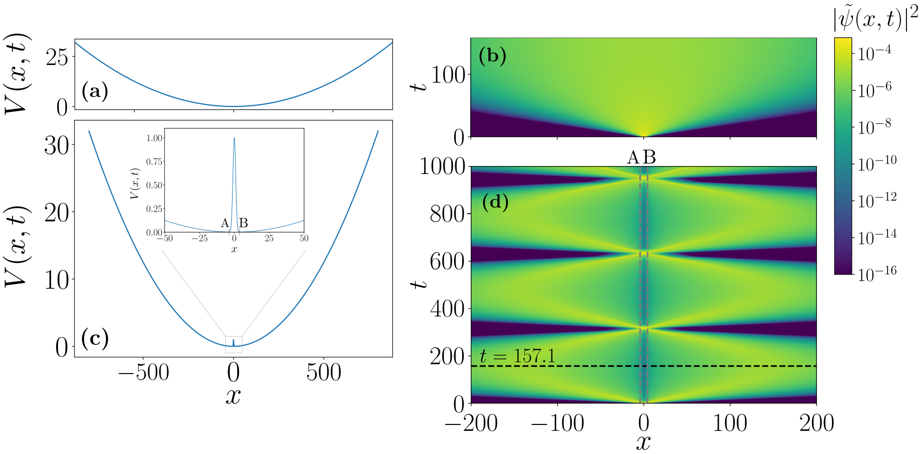


Figure 6.10: Particle's wavefunction expansion in a harmonic and double-well quantum quench (6.7). The simulation uses the MPS split-step method with HDAF differentiation to solve this problem with $\omega_H/\omega_0 = 0.01$, $u = 1$ and $\sigma = 1$, $\Delta t = 0.1$ and $n = 20$. A and B stand for the deepest points of the double well potential. (a) Harmonic oscillator potential $V(x,t)$ (6.2) for $t > 0$. (b) Wavefunction density $|\tilde{\psi}(x,t)|^2$ for harmonic quantum quench. (c) Double-well potential $V(x,t)$ (6.7) for $t > 0$. (d) Wavefunction density $|\tilde{\psi}(x,t)|^2$ for double-well quantum quench.

the multiple MPO operations of the standard MPS spectral method. This renders the HDAF MPS operators a suitable alternative to spectral methods for solving time-dependent PDEs.

6.4.2 Double well potential

Sections 6.3.4 and 6.4.1 studied the MPS numerical methods for a problem with a known solution, enabling a thorough benchmark of the algorithms, spatial discretization, time steps, and truncation errors. This information can now be used to analyze a problem of physical interest: the evolution of a levitated particle after the sudden change from a harmonic to an anharmonic double-well potential described in section 6.1. This is a relevant problem in levitodynamics, where similar anharmonic potentials are used for creating macroscopic quantum superposition, a problem that has been explored both experimentally and computationally [4, 295, 296, 297, 302, 303, 280, 278, 298, 299].

The weakening of the harmonic potential in section 6.4.1 led to an

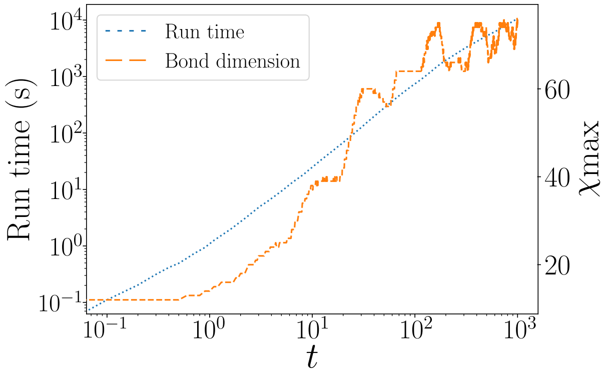


Figure 6.11: Run time and $\chi_{\max}$ of a particle's wavefunction expansion in a double-well quantum quench (6.7). The simulation uses the MPS split-step method with HDAF differentiation to solve this problem with $\omega_H/\omega_0 = 0.01$, $u = 1$ and $\sigma = 1$, $\Delta t = 0.1$ and $n = 20$.

expansion of the particle's wavefunction. Introducing an anharmonic term modifies this behavior, resulting in a particle in a quantum superposition of two macroscopically distinguishable states associated with positions far away in the experiment's left or right halves.

For this particular simulation, the double-well potential is an open harmonic trap divided by a small Gaussian perturbation (6.7), using $u = 1$ and $\sigma = 1$, and a trapping frequency that is once more 100 times smaller than the initial particle trap. For the harmonic trap (Figure 6.10(a)), the new frequency $0.01 = \omega_H < \omega_0{=}1$ weakens the potential and thus the wavefunction's confinement. This potential causes an expansion of the wavefunction's probability density $|\tilde{\psi}(x,t)|^2$, achieving the maximum width for a time $t = 0.5\pi/\omega_H$, as predicted by the analytic solution (6.4) and depicted in Figure 6.10(a). The double-well potential in Figure 6.10(c) adds a repulsive anharmonic Gaussian term since $u > 0$, separating the larger potential in two wells, with a barrier around $x = 0$. Figure 6.10(d) reveals multiple cycles of collapse and revival dynamics of the double-well quantum quench's evolution, a consequence of adding the anharmonic term. The Gaussian term creates a barrier in the potential that divides the probability density into two localization peaks traveling in opposite directions, achieving the desired superposition. On the other hand, the harmonic term maintains the confinement and determines the evolution's period $t_f = 0.5\pi/\omega_H \approx 157.1$, coinciding with the maximum wavefunction spread.

The Gaussian barrier also alters the behavior of the state's bond dimension, which no longer follows a cyclic pattern, as in the case of the harmonic potential (Figure 6.11). Instead, it seems to saturate, decreasing the linear scaling of the run time as the system evolves.

6.5 Conclusions and future perspectives

This chapter introduced an MPS numerical method to simulate the time evolution of a levitated particle under a sudden change in its potential. This is a numerically challenging problem due to the chirp and rapid oscillations in the phase and the large expansion of the particle's wavefunction. The proposed algorithm combined a novel HDAF encoding of differential operators in an MPS/QTT framework with global evolution schemes independent of the locality of interactions. This is a highly efficient technique for solving the time evolution problem, achieving pseudospectral accuracy matching the vector alternatives, with exponential advantages in memory and competitive execution times.

The study benchmarked a selection of quantum-inspired methods to solve time-dependent PDEs: explicit and implicit Runge-Kutta methods, restarted Arnoldi iteration, and a split-step method. In particular, the HDAF operator enables the split-step method by approximating the free propagator unitary operator, which standard finite difference schemes cannot efficiently approximate.

First, the method's performance is assessed for a sudden weakening of a harmonic potential whose analytic solution is known. When used with HDAF, time evolution techniques surpass their finite difference implementation in precision while maintaining a comparable cost. Furthermore, the split-step method offers the most balanced combination of precision and cost and demonstrates that it is competitive with state-of-the-art FFT vector implementations.

When considering one or more expansion cycles, the HDAF's split-step method accurately reproduced the behavior of both the harmonic and anharmonic potentials. This allowed for exponentially efficient function encodings and moderate simulation overheads despite the chirp inherent to the problem. Larger expansions will require considering unitary transformations to avoid the effect of the chirp on the increase of the bond dimension.

Though the implementation cost of the MPS and FFT implementations is similar, only the MPS algorithm can scale up these grid densities to more dimensions due to the exponential scaling of memory of vector-based alternatives. MPS's favorable memory scaling, combined with the pseudospectral accuracy and reduced cost of the HDAF approximation, motivate the further development and optimization of these routines for higher-dimensional systems. Indeed, the developed techniques could be extended to the density matrix formalism to consider the effect of imperfections in the system, which is crucial for attaining the minimal decoherence necessary for the unitary expansion of the wavefunction and creating methods for the effective creation of a macroscopic superposition. [317, 4]. In addition, this technique can be used for other time-evolving systems or time-dependent PDEs in general.

Finally, the implementation of these techniques relies on the SElf-Explaining Matrix Product State (SeeMPS) [51] library (see Chapter 7). This library is based on Python, and the run time prefactors could be improved by low-level optimizations and C/C++ backends, among other alternatives.

Chapter 7

SElf-Explaining Matrix-Product-State library

The SElf-Explaining Matrix-Product-State (SeeMPS) [51] library constitutes a simple introduction to implementing MPS algorithms. Its goal is to provide a user-friendly and highly flexible code to facilitate understanding of these techniques and their adaptation to the user's needs. It is based on Python 3 and sped up via Cython, enabling rapid prototyping and testing before dwelling on more advanced and optimized versions of the algorithms. This said, the library has been used in some heavy-duty simulations involving tens and hundreds of qubits [38, 49, 52, 50].

This library groups the MPS algorithms presented in this thesis, and it is used to obtain the results in chapters 5 and 6. It enables the representation of quantum states and operators as MPS and MPO and supports the execution of fundamental algorithms like the density matrix Renormalization group (DMRG) and time-evolving block decimation (TEBD). In addition to general quantum many-body tools, the SeeMPS library is specifically designed for numerical analysis operations. The library includes the instruments to create the MPS-MPO finite precision algebra described in section 3.5, which acts as a basis for developing quantum-inspired algorithms. SeeMPS uses this algebra to solve PDEs via different interpolation and differentiation methods combined with algorithms for the two problems studied in previous chapters: global optimization (chapter 5) and evolution (chapter 6) of MPS. These functionalities of the SeeMPS library have been created as a result of this thesis, and they are shown in Figure 7.1 together with some complementary ones. This chapter briefly summarizes how to use the SeeMPS library for quantum-inspired applications.

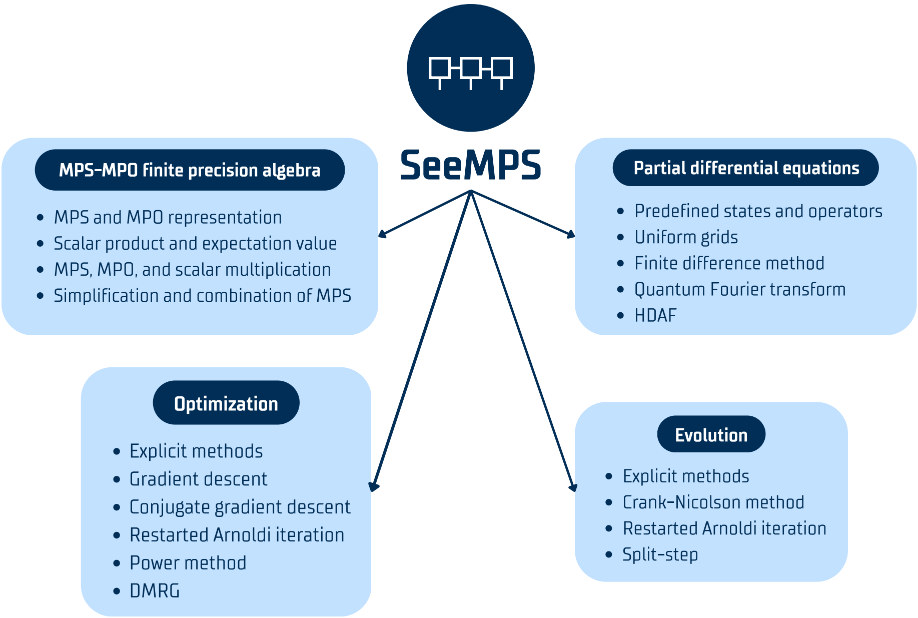

SeeMPS
MPS-MPO finite precision algebra
• MPS and MPO representation
• Scalar product and expectation value
• MPS, MPO, and scalar multiplication
• Simplification and combination of MPS
Partial differential equations
• Predefined states and operators
• Uniform grids
• Finite difference method
• Quantum Fourier transform
• HDAF
Optimization
• Explicit methods
• Gradient descent
• Conjugate gradient descent
• Restarted Arnoldi iteration
• Power method
• DMRG
Evolution
• Explicit methods
• Crank-Nicolson method
• Restarted Arnoldi iteration
• Split-step

Figure 7.1: Functionalities of the SeeMPS library in this thesis. The SeeMPS library includes tools for the general MPS-MPO representation of quantum states and most extended algorithms. This thesis enlarges it for quantum-inspired numerical applications, including MPS-MPO finite precision algebra, differentiation, interpolation, and solution of partial differential equations.

7.1 Function and operator representation

The MPS-MPO finite precision algebra in section 3.5 enables the development of quantum-inspired algorithms for numerical analysis. The first step is to represent functions as MPS following the prescriptions of section 3.6, i.e., mapping the binary encoding of functions in an n-qubit quantum register to an n-site MPS.

Let us encode a Gaussian function $f(x) = e^{-x^2/2}$ as an MPS[1]. The `Space` class creates the function's domain by proving the number of qubits per dimension and its corresponding interval.

```
from seemps.analysis.space import Space
n = 8
interval = [-5,5]
space = Space([n], [interval])
```

[1] The MPS representation of functions does not need functions to be normalized, as opposed to the quantum register encoding.

Then, the from_vector method of the MPS class constructs the MPS corresponding to the vector representation of the function, where the physical dimension of the sites of the MPS will always be two due to the binary encoding.

```
import numpy as np
from seemps.state import MPS
f = np.exp(-space.x[0]**2/2)
f_mps = MPS.from_vector(f, [2]*n, normalize=False)
```

The from_vector method is the most straightforward way of loading a function to an MPS from its vector representation. It uses the SVD with a truncation strategy that can be modified to obtain an exact state representation. However, the method is limited by the capacity of the vector representation to encode the function. SeeMPS also includes function-loading MPS techniques that avoid this problem by directly creating the MPS. In Ref. [52], Rodríguez-Aldavero introduced a function loading technique based on Chebyshev expansion and compared it against TT-cross interpolation and multiscale interpolative constructions. These techniques were added to the library as part of the previous work.

The SeeMPS library includes the representation of predefined functions as MPS. These functions, such as the constant, exponential, sine, and cosine functions, are objects of the MPS class and are exactly constructed from the corresponding list of rank-3 tensors provided to the class. These functions also correspond to diagonal potential operators, which can be applied as an MPS multiplication or recast as MPO.

The MPO class provides the MPO encoding of operators. These operators can be exactly constructed either as a list of rank-4 tensors or approximated through basis expansions, such as the Chebyshev expansion, which relies on polynomials that are exactly representable as MPO.

7.2 Basic operations

The finite precision algebra in Section 3.5 allows working with MPS and MPO similarly to the elements of the standard matrix-vector algebra in numerical analysis applications. This algebra includes basic operations, such as multiplication, addition, and subtraction. The symbols corresponding to these operations are overloaded in the MPS and MPO classes, facilitating its use. In addition, the MPS is automatically truncated to avoid undesirable growth of the bond dimension. This truncation can be modified to the needs of the problem.

For example, we can create the MPO corresponding to the exponential function for the previously defined `Space`.

```
from seemps.analysis.operators import exponential_mpo
op = exponential_mpo(n, interval[0], space.dx[0])
```

Then, the overloaded algebraic symbols allow us to construct a new MPS as the linear combination of the original one and the application of the operator on it.

```
new_f_mps = op @ f_mps
combined_state = 0.1 * new_f_mps - 3 * f_mps
```

It is possible to operate with MPOs similarly, enabling the construction of more complex operators, such as Hamiltonians, and representations of different PDEs.

7.3 Differentiation

The SeeMPS library includes three differentiation techniques: finite difference (Section 2.2.2), Fourier spectral method (Section 2.2.2), and Hermite Distributing Approximating Functionals (HDAF) (Section 6.2). While the finite difference and Fourier methods are commonly used in quantum-inspired applications for numerical analysis, HDAF constitutes a novel technique for approximating functions of derivatives in this context.

As an example, let us approximate the first derivative of the previously defined Gaussian function using the finite difference method.

```
from seemps.analysis.finite_differences import
                                    smooth_finite_differences_mpo

op_diff = smooth_finite_differences_mpo(n, order=1, dx=space.dx
                                    [0])
df = op_diff @ f_mps
```

7.4 Interpolation

There are two interpolation techniques in the SeeMPS library: finite difference and Fourier interpolation, both introduced in Section 2.3. Since the implementation steps are similar, let us focus on the finite difference interpolation for an example.

We present the necessary steps to interpolate the defined Gaussian function. This function is defined in a space with 2^n points, i.e., n qubits/sites for the MPS representation. The goal is to interpolate it to a representation with $m = n + 1$ qubits, doubling the number of grid points. Since the function and the `Space` instance are already created, it is only necessary to apply the function for finite difference interpolation to obtain the interpolated function.

```
from seemps.analysis.interpolation import
                                finite_differences_interpolation

f_int_mps = finite_differences_interpolation(f_mps, space)
```

7.5 Representation of partial differential equations

It is possible to represent as MPOs by combining the differentiation techniques in Section 7.3 with the representation of potential operators. As an example, let us encode the Hamiltonian of the one-dimensional quantum harmonic oscillator

$$H = -\frac{1}{2}\partial_x^2 + \frac{1}{2}x^2, \tag{7.1}$$

which we studied in chapters 4, 5, and 6.

The following function creates the MPO for this PDE using a finite difference approximation of the differential operator. We will construct the Hamiltonian using the previously defined `Space`.

```
from seemps.analysis.operators import x_to_n_mpo
a, dx = space.a[0], space.dx[0]
potential = 0.5 * x_to_n_mpo(n, a, dx, n=2)
derivative = -0.5 * smooth_finite_differences_mpo(n, order=2,
                                dx=dx)
H = (potential + derivative).join()
```

7.6 solution of partial differential equations

The SeeMPS library contains the global optimization and evolution algorithms to solve the Hamiltonian and evolution PDEs in chapters 5 and 6. These algorithms are: explicit methods—Euler, improved Euler, Runge-Kutta, and Runge-Kutta-Fehlberg—and the implicit Crank-Nicolson method, gradient descent, power method, explicitly restarted

Arnoldi iteration, and split-step schemes. It also includes the DMRG algorithm for the optimization problem.

As an example, let us solve the harmonic oscillator (7.1) Hamiltonian PDE $Hf(x) = Ef(x)$. We use the DMRG algorithm, but the implementation of the global optimization algorithms is analogous. Since the Hamiltonian MPO is already defined, only providing it to the `dmrg` function outputs the optimization solution.

```
from seemps.optimization import dmrg
result = dmrg(H)
```

The functions for resolving PDEs admit different parameters, such as the initial guess, truncation strategy, and number of iterations. This enables the optimization of the parameters for each application.

7.7 Conclusions and future perspectives

This thesis significantly contributed to developing the SElf-Explaining Matrix Product State (SeeMPS) [51] library by its extension to numerical analysis problems. We have introduced around 12 new interpolation, differentiation, optimization, and evolution algorithms that rely on a novel MPS-MPO finite precision algebra (Section 3.5). Chapters 5 and 6 demonstrate the effectiveness of these methods for solving static and time-dependent PDEs, showing the versatility of the SeeMPS library. This general framework lays the groundwork for more general optimization and time-dependent problems as long as they are amenable to an MPS representation. The predefined functions and operators and the function loading methods [52] support this extension by facilitating the encoding of new problems into the library. Additionally, implementing finite precision algebra by overloading standard matrix-vector operations for addition and multiplication enables a straightforward adaptation of common vector techniques to this formalism.

Despite the favorable memory and time scaling of quantum-inspired numerical analysis methods shown in chapters 5 and 6, their performance can be enhanced by upgrading the SeeMPS library. We chose Python due to its popularity and user-friendly nature to broaden the library's accessibility. However, as an interpreted language, Python may impose limitations in the algorithms' run time. Preliminary tests indicate that migrating the code to C++ could yield significant performance improvements. Furthermore, parallelizing MPS operations could substantially reduce computation time for heavy computations.

Future upgrades of the SeeMPS library, together with adding more numerical techniques—such as the efficient implementation of the quantum Fourier transform [148]—will pave the path for further applications.

Chapter 8

Conclusions and outlook

"Too many scholars think of research as purely a cerebral pursuit. If we do nothing with the knowledge we gain, then we have wasted our study. Books can store information better than we can—what we do that books cannot is interpret. So if one is not going to draw conclusions, then one might as well just leave the information in the texts."

— Brandon Sanderson, *The Way of Kings*

This thesis introduced quantum and quantum-inspired methods for effectively solving partial differential equations (PDEs). PDEs are prevalent in many fields, including the modeling of quantum systems, where traditional methods face computational challenges. We developed a series of algorithms that efficiently represent functions and PDEs as quantum states and operators, achieving significant memory compression and time savings while maintaining high accuracy.

We first leveraged the quantum register properties to resolve Hamiltonian PDEs via a variational quantum algorithm, integrating Fourier-based encoding with new ansätze to represent functions and their symmetries. This algorithm solved the harmonic oscillator and superconducting qubits PDEs with low errors in approximating the ground state and its energy. Our variational quantum PDE solver [46] and related approaches [24, 23, 25, 26, 27, 28] inspired new variational proposals in quantum numerical analysis, such as generalized differentiation rules [25] and state preparation (function loading) techniques [318, 224, 225, 226, 60]. Quantum Fourier interpolation has also been successfully applied beyond numerical analysis to interpolate quantum encoded images [223].

Despite efforts to develop efficient hybrid algorithms for PDEs, state-of-the-art quantum computers limit their performance. We showed that even a simple harmonic oscillator PDE faces accuracy and scalability challenges due to current quantum computers' noise sources. These constraints also extend to similar approaches [23, 24].

As an alternative, we explored quantum-inspired algorithms using matrix product states (MPS) to harness the computational efficiency of quantum techniques in classical settings. The success in the solution of PDEs of quantum-inspired [44, 45] and quantized tensor trains (QTTs) [319, 320, 310] techniques led us to develop global optimization MPS algorithms for the solution of Hamiltonian PDEs. By mapping the qubits to the tensors of an MPS, we extended the encoding of functions in a quantum register to this formalism [38]. Using this idea, in Ref. [49], we created a new MPS-MPO finite precision algebra that acted as a basis for quantum-inspired algorithms. These MPS algorithms showed an asymptotic exponential advantage in memory and time savings compared to standard vector approaches while maintaining similar accuracy, solving multidimensional Hamiltonian PDEs out of the scope of our quantum proposal. Similar approaches rely on local optimization techniques [44, 45]—density matrix renormalization group (DMRG) [87, 88, 89, 90, 91]—, which we show suitable for PDEs with limited entanglement. However, general matrix product operators (MPO) for PDEs are highly entangled, and the global algorithms proposed in this study—combined with interpolation techniques—are generally better suited to address this complexity.

We further applied these techniques to more complex, time-dependent PDEs, such as the time evolution of a dramatic expansion of the wavefunction of a levitated particle under a sudden weakening of its potential [279, 278]. This problem is of great interest in levitodynamics [4], as it is the basis for studying macroscopic quantum superposition. In Ref. [50], we incorporated global time evolution algorithms with a pseudospectral method to represent functions of derivatives, obtaining pseudospectral accuracy at a cost comparable to finite difference approximations, which suffer from higher errors. In addition, MPS methods achieved similar cost and accuracy to spectral vector techniques, with exponential memory efficiency. This demonstrated the potential for MPS-based algorithms to solve more intricate PDEs with higher grid resolutions. However, challenges such as bond dimension scaling due to the wavefunction's chirping remain in specific cases and may be addressed through unitary transformations.

The created MPS-MPO finite precision algebra constitutes an efficient framework for numerical analysis. All the MPS techniques developed in this thesis are contained in the SeeMPS [51] library. This library includes, but is not limited to, the MPS-MPO finite precision algebra, numerical differentiation, interpolation, and optimization and evolution algorithms. The combination of these functionalities with function loading algorithms [52] makes the SeeMPS library a powerful tool for creating quantum-inspired algorithms for numerical analysis.

Although this thesis showed the potential of MPS PDE solvers, its scope was limited to PDEs tractable by classical methods. This is partly due to the time limitations of this project and the unforeseen challenges from the combination of numerical methods with MPS approximations, with bond dimension truncation errors adding to the usual discretization errors and machine precision. It is important to consider the interplay among these errors when designing quantum-inspired algorithms for numerical analysis, adapting the created methods to an MPS framework, and combining this with a careful study of the algorithm's parameters to ensure convergence. Despite the limitations of our study, the advantages demonstrated by MPS algorithms motivate future research avenues to improve these methods and expand their applications to more complex and diverse problems.

The first step would be to enhance the SeeMPS library. Preliminary tests indicate that upgrading from Python to a C++ implementation can lead to a 10x speedup, with further improvements through parallelization of MPS operations. The library could also benefit from efficiently implementing spectral methods following Ref. [148]. The improved algorithms' run time and accuracy would enable them to solve higher-dimensional problems or denser grids. For instance, it would be possible to develop a quantum-inspired library to diagonalize general superconducting circuits that serves as an alternative to state-of-the-art software [150, 151, 152], restrained by the limitations of sparse matrix diagonalization. In the evolution of a levitated particle, using the density matrix formalism—a multidimensional problem—will model the effect of imperfection, better approximating the experimental settings. In addition, given the good performance of the MPS techniques in this thesis, we believe that these methods will also become useful in the study of many-body physics problems, joining other techniques used for long-range interactions [273, 272, 249, 274].

The physically inspired problems in this thesis open the door for further applications. MPS optimization could be extended to solve general non-hermitian static PDEs using self-adjoint operators, which is common when working with Lindblad operators. Furthermore, these techniques apply to source PDEs reformulated as a cost minimization and general optimization problems as long as they can be encoded as a Hamiltonian MPO. MPS time-dependent PDE solvers could find applications in finance via the solution of the Black-Scholes equation [321], fluid dynamics, and reaction-diffusion equations. Indeed, similar problems have already been tackled from a quantum-inspired perspective [322, 323, 45], expanding tensor networks beyond quantum physics and sparking increasing interest, extending from academia to industry. However, most of these studies are heuristic, and an in-depth analysis of which problems are appropriate for this formalism remains a necessary task. Some efforts have already been made to formalize the bond dimension bounds of certain functions, and the results in this thesis corroborate them.

Even in a future with fault-tolerant quantum computing, tensor networks will continue to be useful in low-entanglement scenarios and for problems unsuitable for quantum algorithms, such as those that do not meet the unitarity and normalization requirements. Therefore, the development of tensor-network quantum-inspired methods remains an important long-term task.

This thesis contributes to studying and developing quantum-inspired methods, particularly in a numerical analysis context. The MPS-MPO finite precision algebra is a highly efficient tool for creating such algorithms. This efficiency has been demonstrated through a series of global algorithms to tackle static and time-dependent PDEs, showing exponential memory compression and asymptotic time savings with respect to standard vector techniques while matching its accuracy. Although many applications of this thesis are still amenable to vector techniques, the results prove the suitability of MPS methods and encourage their application for higher dimensional problems or denser grids. Finally, future enhancement of the SeeMPS algorithm will reduce the run time and memory scaling of MPS operations, enabling finite precision algebra to approximate more complex problems and explore new quantum-inspired applications.

Conclusiones y perspectivas

Esta tesis ha introducido métodos cuánticos y de inspiración cuántica para resolver de manera efectiva ecuaciones en derivadas parciales (EDPs). Las EDPs son comunes en muchos campos, incluyendo la modelización de sistemas cuánticos, donde los métodos tradicionales enfrentan desafíos computacionales. Desarrollamos una serie de algoritmos que representan funciones y EDPs como estados y operadores cuánticos, logrando una compresión significativa de memoria y ahorro de tiempo, manteniendo una alta precisión.

Primero, aprovechamos las propiedades del registro cuántico para resolver EDPs Hamiltonianas mediante un algoritmo cuántico variacional, integrando codificación basada en Fourier con nuevos ansätze para representar funciones y sus simetrías. Este algoritmo resolvió las EDPs del oscilador armónico y de los qubits superconductores con bajos errores en la aproximación del estado base y su energía. Nuestro resolutor variacional de EDPs cuánticas [46] y otras propuestas relacionados [24, 23, 25, 26, 27, 28] inspiraron nuevos algoritmos variacionales en análisis numérico cuántico, como reglas de diferenciación generalizadas [25] y técnicas de preparación de estados (carga de funciones) [318, 224, 225, 226, 60]. La interpolación cuántica de Fourier también se ha aplicado con éxito más allá del análisis numérico para interpolar imágenes codificadas cuánticamente [223].

A pesar de los esfuerzos para desarrollar algoritmos híbridos eficientes para las EDPs, los ordenadores cuánticos actuales limitan su rendimiento. Demostramos que incluso una simple EDP de oscilador armónico enfrenta desafíos de precisión y escalabilidad debido a las fuentes de ruido de los ordenadores cuánticos actuales. Estas limitaciones también se extienden a técnicas similares [23, 24].

Como alternativa, exploramos algoritmos inspirados en la computación cuántica utilizando estados producto de matriz (MPS) para aprovechar la eficiencia computacional de las técnicas cuánticas en contextos clásicos.

El éxito en la resolución de EDPs mediante técnicas de inspiración cuántica [44, 45] y trenes de tensores cuantizados (QTTs) [319, 320, 310] nos llevó a desarrollar algoritmos de optimización global MPS para la resolución de EDPs Hamiltonianas. Al mapear los qubits a los tensores de un MPS, extendimos la codificación de funciones en un registro cuántico a este formalismo [38]. Usando esta idea, en Ref. [49], creamos un nuevo álgebra de precisión finita MPS-MPO que sirvió como base para algoritmos de inspiración cuántica. Estos algoritmos MPS mostraron una ventaja exponencial asintótica en memoria y ahorro de tiempo en comparación con los enfoques vectoriales estándar, manteniendo una precisión similar, resolviendo EDPs Hamiltonianas multidimensionales fuera del alcance de nuestra propuesta cuántica. Métodos similares se basan en técnicas de optimización local [44, 45]—grupo de renormalización de matriz densidad (DMRG) [87, 88, 89, 90, 91]—, que demostramos ser adecuadas para EDPs con entrelazamiento limitado. Sin embargo, los operadores de producto de matriz (MPO) generales para EDPs están altamente entrelazados, y los algoritmos globales propuestos en este estudio—combinados con técnicas de interpolación—son en general más adecuados para abordar esta complejidad.

Aplicamos además estas técnicas a EDPs más complejas y dependientes del tiempo, como la evolución temporal de una dramática expansión de la función de onda de una partícula levitada bajo un debilitamiento repentino de su potencial [279, 278]. Este problema es de gran interés en levitodinámica [4], ya que es la base para estudiar la superposición cuántica macroscópica. En Ref. [50], incorporamos algoritmos de evolución temporal global con un método seudoespectral para representar funciones de derivadas, obteniendo precisión seudoespectral a un costo comparable a las aproximaciones de diferencias finitas, que sufren de mayores errores. Además, los métodos de MPS lograron un coste y precisión similares a las técnicas vectoriales espectrales, con una eficiencia exponencial en memoria. Esto demostró el potencial de los algoritmos basados en MPS para resolver EDPs más intrincadas con resoluciones de malla más altas. Sin embargo, en casos específicos, desafíos como el escalado de la dimensión de enlace debido al “chirp” de la función de onda aún persisten, y podrían abordarse mediante transformaciones unitarias.

El álgebra de precisión finita MPS-MPO creada constituye un marco eficiente para el análisis numérico. Todas las técnicas MPS desarrolladas en esta tesis están contenidas en la biblioteca SeeMPS [51]. Esta biblioteca incluye, entre otros, el álgebra de precisión finita MPS-MPO, diferenciación numérica, interpolación, y algoritmos de optimización y evolución. La combinación de estas funcionalidades con algoritmos de carga de funciones [52] hace de la biblioteca SeeMPS una herramienta poderosa para crear algoritmos de inspiración cuántica para el análisis numérico.

Aunque esta tesis mostró el potencial de los resolutores de EDPs basados en MPS, su alcance se limitó a EDPs tratables mediante métodos clásicos. Esto se debe en parte a las limitaciones de tiempo de este proyecto y a los desafíos imprevistos derivados de la combinación de métodos numéricos con aproximaciones MPS, donde los errores de truncamiento de la dimensión de enlace se suman a los errores habituales de discretización y precisión de máquina. Es importante considerar la interacción entre estos errores al diseñar algoritmos de inspiración cuántica para el análisis numérico, adaptando los métodos creados a un marco MPS y combinándolos con un estudio cuidadoso de los parámetros del algoritmo para garantizar la convergencia. A pesar de las limitaciones de nuestro estudio, las ventajas demostradas por los algoritmos MPS motivan futuras líneas de investigación para mejorar estos métodos y expandir sus aplicaciones a problemas más complejos y diversos.

El primer paso sería mejorar la biblioteca SeeMPS. Las pruebas preliminares indican que actualizar de Python a una implementación en C++ puede llevar a un aumento de velocidad de 10 veces, con mejoras adicionales a través de la paralelización de las operaciones MPS. La biblioteca también se beneficiaría de implementar métodos espectrales de manera eficiente siguiendo Ref. [148]. El tiempo de ejecución y la precisión de los algoritmos mejorados les permitiría resolver problemas de mayor dimensión o mallas más densas. Por ejemplo, sería posible desarrollar una biblioteca de inspiración cuántica para diagonalizar circuitos superconductores generales que sirva como una alternativa al software de vanguardia [150, 151, 152], limitado por las restricciones de la diagonalización de matrices dispersas. En la evolución de una partícula levitada, el uso del formalismo de matriz densidad—un problema multidimensional—modelará mejor el efecto de las imperfecciones, aproximándose más a los entornos experimentales. Además, dado el buen rendimiento de las técnicas MPS en esta tesis, creemos que estos métodos también serán útiles en el estudio de problemas de física de muchos cuerpos, uniéndose a otras técnicas utilizadas para interacciones de largo alcance [273, 272, 249, 274].

Los problemas inspirados en la física en esta tesis abren la puerta a nuevas aplicaciones. La optimización MPS podría extenderse para resolver EDPs estáticas generales no hermíticas utilizando operadores autoadjuntos, algo común al trabajar con operadores de Lindblad. Además, estas técnicas se aplican a EDPs fuente reformuladas como problemas de minimización de costo y de optimización general, siempre que puedan codificarse como un MPO Hamiltoniano. Los solucionadores MPS de EDPs dependientes del tiempo podrían encontrar aplicaciones en finanzas mediante la resolución de la ecuación de Black-Scholes [321], dinámica de fluidos y ecuaciones de reacción-difusión. De hecho, problemas similares ya han sido abordados desde una perspectiva de inspiración cuántica [322, 323, 45], expandiendo las redes tensoriales más allá de la física cuántica y generando un creciente interés, que se extiende desde la academia hasta la industria. Sin embargo, la mayoría de estos estudios son heurísticos, y un análisis en profundidad sobre qué problemas son apropiados para este formalismo sigue siendo una tarea necesaria. Ya se han hecho algunos esfuerzos para formalizar los límites de la dimensión de enlace de ciertas funciones, y los resultados en esta tesis los corroboran.

Incluso en un futuro con computación cuántica tolerante a fallos, las redes tensoriales seguirán siendo útiles en escenarios de bajo entrelazamiento y para problemas inadecuados para algoritmos cuánticos, como aquellos que no cumplen con los requisitos de unitariedad y normalización. Por lo tanto, el desarrollo de métodos de inspiración cuántica basados en redes tensoriales sigue siendo una tarea importante a largo plazo.

Esta tesis contribuye al estudio y desarrollo de métodos de inspiración cuántica, particularmente en un contexto de análisis numérico. El álgebra de precisión finita MPS-MPO es una herramienta altamente eficiente para crear tales algoritmos. Esta eficiencia se ha demostrado a través de una serie de algoritmos globales para abordar EDPs estáticas y dependientes del tiempo, mostrando una compresión exponencial de memoria y ahorros asintóticos de tiempo respecto a las técnicas vectoriales estándar, mientras iguala su precisión. Aunque muchas aplicaciones de esta tesis aún son abordables mediante técnicas vectoriales, los resultados prueban la idoneidad de los métodos de MPS y animan su aplicación para problemas de mayor dimensión o mallas más densas. Finalmente, la mejora futura del algoritmo SeeMPS reducirá el tiempo de ejecución y la escalabilidad de memoria de las operaciones MPS, permitiendo que el álgebra de precisión finita aproxime problemas más complejos y explore nuevas aplicaciones de inspiración cuántica.

Appendices

Appendix A

Fourier interpolation

Fourier interpolation is a very efficient method for interpolating periodic bandwidth-limited functions. Section 2.3 introduces this method and its general implementation. This appendix provides a mathematical explanation of how the steps taken in that section lead to interpolation. Additionally, we include a Python code to implement this method.

The process of Fourier interpolation can be divided into three steps, which are explained below.

1. Fourier transform of the original function

   The first step involves transforming the original function $f(x_s)$, with $s = 0, \dots, N-1$, into momentum space using the discrete Fourier transform (DFT)

   $$\tilde{f}(p_l) = \frac{1}{\sqrt{N}} \sum_{s=0}^{N-1} e^{-2\pi i l s/N} f(x_s), \quad l = -N/2, \dots, N/2-1, \tag{A.1}$$

   where l is the momentum (or frequency) index.

2. Zero padding in Fourier space

   To increase the resolution of the function in position space, we extend the grid from N points to M points, where $M > N$, typically with $N = 2^n, M = 2^m$, matching the quantum register encoding.

   The original DFT coefficients $\tilde{f}(p_l)$ are defined only for l in the range $-\frac{N}{2} \leq l \leq \frac{N}{2} - 1$. We extend these Fourier coefficients by adding zeros for frequencies outside this range to achieve higher resolution in position space. The extended Fourier coefficients $\tilde{g}(p_{\tilde{l}})$ are given by

   $$\tilde{g}(p_{\tilde{l}}) = \begin{cases} \sqrt{\frac{M}{N}} \tilde{f}(p_{\tilde{l}}) & \text{for } -\frac{N}{2} \leq \tilde{l} \leq \frac{N}{2} - 1, \\ 0 & \text{otherwise,} \end{cases} \tag{A.2}$$

   where $\tilde{l}$ is the momentum index for the extended grid in Fourier space and $\sqrt{M/N}$ is a rescaling factor that ensures that the amplitude of the function is preserved after interpolation.

3. Inverse Fourier transform to obtain the interpolated function

   The inverse Fourier transform leads to the interpolated function

$$f^{(I)}(\tilde{x}_{\tilde{s}}) = \frac{1}{\sqrt{M}} \sum_{\tilde{l}=-\frac{M}{2}}^{\frac{M}{2}-1} \tilde{g}(p_{\tilde{l}}) e^{2\pi i \tilde{l}\tilde{s}/M}, \quad \tilde{s} = 0, 1, \ldots, M-1 \quad \text{(A.3)}$$

   Here, $\tilde{s}$ represents the index for the interpolated grid in position space, with the new grid points $\tilde{x}_{\tilde{s}} = -L_x/2 + \tilde{s}\Delta\tilde{x}$, where $\Delta\tilde{x} = \Delta x \frac{N}{M}$, resulting a denser grid after interpolation.

The code below shows a Fourier interpolation Python code. Shifts of the function are performed to order the Fourier coefficients correctly.

```
def fourier_interpolation(f, M):
    """ 1D Fourier interpolation.

    Parameters
    ----------
    f: numpy.ndarray
    Function to interpolate.
    M: int
    Number of points of the final function.

    Return
    ------
    Fourier interpolated f with M  points.
    """
    N = f.size
    F = np.fft.fftshift(np.fft.fft(f, norm='ortho'))
    F_padded = np.pad(F, (int((M-N)/2), int((M-N)/2)), mode='
                                            constant', constant_values=
                                            0)
    return np.sqrt(M/N)*np.fft.ifft(np.fft.ifftshift(F_padded),
                                             norm='ortho')
```

Appendix B

Numerical results of the ideal simulations of the variational quantum PDE solver

This appendix shows the numerical results of Section 4.5.2.

Qubits	Ansätze	Parameters	CNOT gates	$1 - F^{\infty}$ COBYLA	$1 - F^{\infty}$ SPSA	$1 - F^{\infty}$ Adam	$1 - F^{\infty}$ L-BFGS-B
2	RY1	2	1	0.0320(7)	0.03182(7)	0.03185(4)	$3.19 \cdot 10^{-2}$
2	RY2	3	1	**0.0310(8)**	0.03184(3)	0.03185(0)	$3.19 \cdot 10^{-2}$
2	ZGR	1	1	0.0319(7)	0.03186(8)	0.03188(0)	$3.19 \cdot 10^{-2}$
3	RY1	4	3	0.002(3)	0.00008(0)	0.000060(8)	$5.89 \cdot 10^{-5}$
3	RY2	6	4	0.001(4)	0.000070(3)	**0.000060(5)**	$5.89 \cdot 10^{-5}$
3	ZGR	3	4	0.0019(4)	0.0003(8)	0.000069(9)	$5.89 \cdot 10^{-5}$
4	RY1	6	6	0.005(8)	0.0000(6)	0.0000(2)	$2.13 \cdot 10^{-5}$
4	RY2	9	9	0.006(8)	0.0002(2)	0.0000(8)	$1.72 \cdot 10^{-10}$
4	ZGR	7	9	0.006(6)	0.000(1)	**0.000019(1)**	$1.73 \cdot 10^{-9}$
5	RY1	8	10	0.02(7)	0.0019(2)	0.001(7)	$1.47 \cdot 10^{-3}$
5	RY2	12	16	0.02(5)	0.001(8)	0.003(4)	$4.51 \cdot 10^{-8}$
5	ZGR	15	18	0.01(2)	0.002(6)	**0.000013(5)**	$6.23 \cdot 10^{-8}$
6	RY1	10	15	0.09(3)	0.007(4)	0.006(2)	$5.71 \cdot 10^{-3}$
6	RY2	15	25	0.14(1)	0.003(8)	0.008(2)	$1.47 \cdot 10^{-5}$
6	ZGR	31	35	0.10(1)	0.00(1)	**0.00006(9)**	$4.68 \cdot 10^{-8}$

Table B.1: Noiseless results of the simulations for the harmonic oscillator for each number of qubits, ansätze, and optimizers. The number of parameters and CNOT gates of each ansatz are shown for each number of qubits. The infidelities for the COBYLA, SPSA, and Adam optimizers are the median of the infidelities over 100 simulations using the QISKIT QASM simulator with 8192 evaluations. The L-BFGS-B optimizer is combined with the statevector simulator to establish the numerical limit of each ansatz. We highlight the best result for each number of qubits in bold letter font.

Qubits	Ansätze	Parameters	CNOT gates	$1 - F^\infty$ COBYLA	$1 - F^\infty$ SPSA	$1 - F^\infty$ Adam	$1 - F^\infty$ L-BFGS-B
2	RY1	2	1	0.159(5)	0.15907(4)	0.15910(0)	$1.59 \cdot 10^{-1}$
2	RY2	3	1	0.1602(3)	0.15909(8)	0.15911(0)	$1.59 \cdot 10^{-1}$
2	ZGR	1	1	0.159(2)	**0.15906(5)**	0.15917(8)	$1.59 \cdot 10^{-1}$
3	RY1	4	3	0.003(1)	0.00129(0)	0.00128(4)	$1.28 \cdot 10^{-3}$
3	RY2	6	4	0.003(1)	0.00129(2)	**0.001280(2)**	$1.28 \cdot 10^{-3}$
3	ZGR	3	4	0.0020(0)	0.00134(4)	0.001314(4)	$1.28 \cdot 10^{-3}$
4	RY1	6	6	0.007(9)	0.0001(8)	0.0001(2)	$1.06 \cdot 10^{-4}$
4	RY2	9	9	0.010(1)	0.0005(7)	0.0001(9)	$5.67 \cdot 10^{-11}$
4	ZGR	7	9	0.006(0)	0.000(5)	**0.000109(6)**	$5.78 \cdot 10^{-11}$
5	RY1	8	10	0.11(6)	0.008(1)	0.0066(3)	$6.15 \cdot 10^{-3}$
5	RY2	12	16	0.12(5)	0.002(7)	0.001(8)	$1.51 \cdot 10^{-6}$
5	ZGR	15	18	0.21(7)	0.01(0)	**0.00022(7)**	$1.47 \cdot 10^{-7}$
6	RY1	10	15	0.271(1)	0.01(9)	0.0140(7)	$1.30 \cdot 10^{-2}$
6	RY2	15	25	0.22(2)	0.01(9)	0.010(6)	$3.44 \cdot 10^{-4}$
6	ZGR	31	35	0.25(0)	0.2(0)	**0.00031(4)**	$1.49 \cdot 10^{-5}$

Table B.2: Noiseless results of the simulations for the transmon qubit for each number of qubits, ansätze, and optimizers. We highlight the best result for each number of qubits in bold letter font.

Qubits	Ansätze	Parameters	CNOT gates	$1 - F^{\infty}$ COBYLA	$1 - F^{\infty}$ SPSA	$1 - F^{\infty}$ Adam	$1 - F^{\infty}$ L-BFGS-B
2	RY1	2	1	0.0995(6)	0.09919(4)	0.09919(8)	$9.92 \cdot 10^{-2}$
2	RY2	3	1	0.0996(0)	0.09920(4)	0.09919(6)	$9.92 \cdot 10^{-2}$
2	ZGR	1	1	0.0994(7)	**0.09918(4)**	0.09921(8)	$9.92 \cdot 10^{-2}$
3	RY1	4	3	0.04(7)	0.06(6)	0.0662(9)	$6.64 \cdot 10^{-2}$
3	RY2	6	4	**0.04(6)**	0.06(5)	0.0662(2)	$6.64 \cdot 10^{-2}$
3	ZGR	3	4	0.056(1)	0.057(9)	0.0663(2)	$6.64 \cdot 10^{-2}$
4	RY1	6	6	0.03(5)	0.00(7)	0.000(0)	$8.00 \cdot 10^{-5}$
4	RY2	9	9	0.06(0)	0.00(2)	0.00(0)	$4.35 \cdot 10^{-5}$
4	ZGR	7	9	0.00(5)	0.00(0)	**0.000051(4)**	$4.35 \cdot 10^{-5}$
5	RY1	8	10	0.06(4)	0.04(2)	0.035(2)	$3.36 \cdot 10^{-2}$
5	RY2	12	16	0.07(2)	0.01(0)	0.00(0)	$6.65 \cdot 10^{-5}$
5	ZGR	15	18	0.02(4)	0.04(0)	**0.000029(0)**	$6.96 \cdot 10^{-8}$
6	RY1	10	15	0.10(5)	0.06(9)	0.065(4)	$6.33 \cdot 10^{-2}$
6	RY2	15	25	0.09(7)	0.04(7)	0.03(2)	$6.95 \cdot 10^{-4}$
6	ZGR	31	35	0.15(7)	0.34(8)	**0.00010(3)**	$3.33 \cdot 10^{-6}$

Table B.3: Noiseless results of the simulations for the flux qubit for each number of qubits, ansätze, and optimizers. We highlight the best result for each number of qubits in bold letter font.

Appendix C

Optimum learning rate for gradient descent

The optimum learning rate $\Delta\beta$ of the gradient descent algorithm update rule (5.12) can be analytically computed. This transforms the gradient descent into an adaptative-step method that maximizes the energy decrease at each step, with no need for an accurate initial approximation, as opposed to adaptative-step Runge-Kutta-Fehlberg (section 5.1.1) method.

The first step to compute $\Delta\beta$ is to obtain the functional derivative of the cost function. The variation of any function f by an infinitesimal quantity ε in an arbitrary direction given by the arbitrary function $\eta(\vec{x})$ is represented by

$$\delta f(\vec{x}) = \varepsilon\eta(\vec{x}). \tag{C.1}$$

Hence, the variation of a functional $F[f]$, which in this context is defined as a “function of a function”, is caused by the variation of f by a quantity δf,

$$\delta F := F[f+\delta f] - F[f] = F[f+\varepsilon\eta] - F[f]. \tag{C.2}$$

Evaluating δF using the Taylor expansion of $F[f+\varepsilon\eta]$ in powers of ε leads to

$$\begin{aligned} F[f+\varepsilon\eta] &= F[f] + \left.\frac{dF[f+\varepsilon\eta]}{d\varepsilon}\right|_{\varepsilon=0}\varepsilon + \frac{1}{2}\left.\frac{d^2F[f+\varepsilon\eta]}{d\varepsilon^2}\right|_{\varepsilon=0}\varepsilon^2 + \dots \\ &= \sum_{n=0}^{N}\frac{1}{n!}\left.\frac{d^nF[f+\varepsilon\eta]}{d\varepsilon^n}\right|_{\varepsilon=0}\varepsilon^n + \mathcal{O}(\varepsilon^{N+1}). \end{aligned} \tag{C.3}$$

By definition, the n-th order derivative of F with respect to ε is

$$\left.\frac{d^nF[f+\varepsilon\eta]}{d\varepsilon^n}\right|_{\varepsilon=0} = \int dx_1 ... dx_n \frac{\delta^n F[f]}{\delta f(x_1) ... f(x_n)}\eta(x_1). \tag{C.4}$$

Thus

$$\begin{aligned}\delta F := F[f+\varepsilon\eta] - F[f] &= \frac{dF[f+\varepsilon\eta]}{d\varepsilon}\bigg|_{\varepsilon=0}\varepsilon + \mathcal{O}(\varepsilon^2)\\ &= \varepsilon\left[\int dx_1 \frac{\delta F[f]}{\delta f(x_1)}\eta(x_1)\right] + \mathcal{O}(\varepsilon^2). \end{aligned} \tag{C.5}$$

Let us apply this formula to obtain the energy functional $E[\psi]$ (5.2). First, the energy functional is rewritten as

$$E[\psi+\varepsilon\eta] = \frac{\langle\psi|H|\psi\rangle + \varepsilon\langle\psi|H|\eta\rangle + \varepsilon^*\langle\eta|H|\psi\rangle + \varepsilon^*\varepsilon\langle\eta|H|\eta\rangle}{\langle\psi|\psi\rangle + \varepsilon\langle\psi|\eta\rangle + \varepsilon^*\langle\eta|\psi\rangle + \varepsilon^*\varepsilon\langle\eta|\eta\rangle}. \tag{C.6}$$

Differenciating the previous expression with respect to ε and evaluating it a $\varepsilon = 0$ leads to

$$\begin{aligned}\frac{dE[\psi+\varepsilon\eta]}{d\varepsilon}\bigg|_{\varepsilon=0} &= \frac{\langle\psi|H|\eta\rangle + \varepsilon^*\langle\eta|H|\eta\rangle}{\langle\psi|\psi\rangle + \varepsilon\langle\psi|\eta\rangle + \varepsilon^*\langle\eta|\psi\rangle + \varepsilon^*\varepsilon\langle\eta|\eta\rangle}\\ &- \frac{\langle\psi|\eta\rangle + \varepsilon^*\langle\eta|\eta\rangle}{\left(\langle\psi|\psi\rangle + \varepsilon\langle\psi|\eta\rangle + \varepsilon^*\langle\eta|\psi\rangle + \varepsilon^*\varepsilon\langle\eta|\eta\rangle\right)^2}\\ &\times\left(\langle\psi|H|\psi\rangle + \varepsilon\langle\psi|H|\eta\rangle + \varepsilon^*\langle\eta|H|\psi\rangle + \varepsilon^*\varepsilon\langle\eta|H|\eta\rangle\right)\bigg|_{\varepsilon=0}\\ &= \frac{\langle\psi|H|\eta\rangle}{\langle\psi|\psi\rangle} - \frac{\langle\psi|\eta\rangle}{\langle\psi|\psi\rangle^2}\langle\psi|H|\psi\rangle. \end{aligned} \tag{C.7}$$

As $\langle\psi|\psi\rangle = 1$, the functional derivative of the energy is

$$\frac{\delta E}{\delta\psi} = (H - \langle H\rangle\mathbb{I})\psi. \tag{C.8}$$

Therefore, the exact step for the descent algorithm is given by

$$\psi_{n+1} = \psi_n + \Delta\beta\frac{\delta E}{\delta\psi} = \psi_n + \Delta\beta(H - \langle H\rangle\mathbb{I})\psi, \tag{C.9}$$

and the expression for $E(\Delta\beta, H)$ is

$$E(\Delta\beta, H) = \frac{\langle H\rangle + 2\Delta\beta\langle(H-\langle H\rangle\mathbb{I})^2\rangle + \Delta\beta^2\langle(H-\langle H\rangle\mathbb{I})H(H-\langle H\rangle\mathbb{I})\rangle}{1+\Delta\beta^2\langle(H-\langle H\rangle\mathbb{I})^2\rangle}, \tag{C.10}$$

where we have used that $\langle(H - E\mathbb{I})\rangle = 0$. Minimizing Eq. (C.10) results in

$$\Delta\beta_\pm = \frac{\langle(H-\langle H\rangle\mathbb{I})^3\rangle \pm \sqrt{\langle(H-\langle H\rangle\mathbb{I})^3\rangle^2 + 4\langle(H-\langle H\rangle\mathbb{I})^2\rangle^3}}{2\langle(H-\langle H\rangle\mathbb{I})^2\rangle^2}, \tag{C.11}$$

where the optimum step size is

$$\Delta\beta_- = \frac{\langle H'^3 \rangle - \sqrt{\langle H'^3 \rangle^2 + 4 \langle H'^2 \rangle^3}}{2 \langle H'^2 \rangle^2}, \text{ with } H' = H - \langle H \rangle \mathbb{I}, \qquad \text{(C.12)}$$

as $\Delta\beta < 0$ for the displacement in ψ to be made in the opposite direction of the gradient $\frac{\delta E}{\delta \psi}$. All the terms in Eq. (C.12) can be written in terms of the expectation value of H and its powers.

Appendix D

Approximate diagonalization for other PDEs

It is possible to modify the cost functional in Eq. (5.2) to approximate other types of PDEs by rewriting them to be the solution of the optimization. A suitable extension is the solution of PDEs with a source term $g(x)$,

$$Df(x) = g(x), \quad f(x), g(x) \in \mathbb{C}^N. \tag{D.1}$$

Let us define the cost functional $C[f]$,

$$C[f] = \parallel Df(x) - g(x) \parallel^2, \tag{D.2}$$

leading to an update rule

$$f_{n+1} = f_n + \Delta\beta D^\dagger (Df - g) = f_n + \Delta\beta D^\dagger w, \tag{D.3}$$

with optimum $\Delta\beta$

$$\Delta\beta_{\text{opt}} = -\frac{\langle w|DD^\dagger|w\rangle}{\langle w|DD^\dagger DD^\dagger|w\rangle}. \tag{D.4}$$

This equation is of great importance due to its multiple applications. Relevant PDEs such as Poisson's equation and the heat equation are source PDEs and many mathematical models are reduced to these equations, with applications in various fields, such as the Black-Scholes equation [324] in finance, or applications in the magnetic resonance imaging [325].

Appendix E

Analytic solution of a harmonic quantum quench

Section 6.1 introduces the concept of quantum quench in the context of levitodynamics. For a change in a harmonic potential, Eqs. (6.4)-(6.6) describe the evolution of the wavefunction. These equations are obtained through the expectation values of the $\hat{x}$ and $\hat{p}$ operators and functions of them.

Let us start by assuming that the state at $t = 0$ is the ground state of the initial Hamiltonian with frequency ω_0, i.e.,

$$\psi(x, t = 0) = \left(\frac{\omega_0}{\pi}\right)^{1/4} e^{-\frac{\omega_0 x^2}{2}}, \tag{E.1}$$

and that the state $\psi(x, t)$ at time t is given by

$$\psi(x,t) = \left(\frac{\omega(t)}{\pi}\right)^{1/4} e^{-z(t)x^2}, \quad \text{where } z(t) \in \mathbb{C}, z(t) = \frac{\omega(t)}{2} + i\beta(t). \tag{E.2}$$

Then, obtaining the expressions of $\omega(t)$ and $\beta(t)$ leads to solving the quench dynamics.

To obtain $\omega(t)$, let us start by computing the expectation value of $x^2(t)$

$$\langle \hat{x}^2(t) \rangle = \sqrt{\frac{\omega(t)}{\pi}} \int x^2 e^{-2\Re(z(t))x^2} dx = \frac{1}{2\omega(t)}, \tag{E.3}$$

where we have used that $\sigma(t) = 1/\sqrt{\omega(t)}$ and

$$\langle \hat{x}^2 \rangle = \int x^2 \frac{1}{\mathcal{N}^2} e^{-\frac{x^2}{\sigma^2}} dx = \frac{\sigma^3 \sqrt{\pi}}{2\mathcal{N}^2}. \tag{E.4}$$

The value $\langle \hat{x}^2 \rangle$ can also be calculated from

$$\frac{d}{dt}\hat{x}(t) = -i[\hat{x}, \hat{H}], \tag{E.5}$$

where $\hbar = 1$. This expression allows obtaining $\hat{x}(t)$, and hence the expectation value $\langle \hat{x}^2(t) \rangle$ to substitute in Eq. (E.3) and compute $\omega(t)$.

The first step is to introduce the Hamiltonian $\hat{H} = \frac{\hat{p}^2}{2} + \frac{1}{2}\omega_H^2 \hat{x}^2$ in Equation (E.5)

$$\begin{aligned} \frac{d}{dt}\hat{x}(t) &= -i\left[\hat{x}, \frac{\hat{p}^2}{2} + \frac{1}{2}\omega_H^2\hat{x}^2\right] = -\frac{i}{2}[\hat{x}, \hat{p}^2] \\ &= -\frac{i}{2}([\hat{p}[\hat{x}, \hat{p}] + [\hat{x}, \hat{p}]\hat{p}) = \hat{p}. \end{aligned} \tag{E.6}$$

Analogously for $\hat{p}$

$$\frac{d}{dt}\hat{p} = -i\left[\hat{p}, \frac{1}{2}\omega^2\hat{x}^2\right] = -\omega_H^2\hat{x}. \tag{E.7}$$

Equations (E.6) and (E.7) form a system of equations that leads to

$$\frac{d^2}{dt^2}\hat{x} = \frac{d}{dt}\hat{p} = -\omega_H^2\hat{x} \ \rightarrow \ \frac{d^2}{dt^2}\hat{x} + \omega_H^2\hat{x} = 0, \tag{E.8}$$

As expected, $\hat{x}(t)$ follows the equation of motion of the harmonic oscillator, whose solution is given by

$$\hat{x}(t) = \hat{x}(0)\cos(\omega_H t) + \frac{\hat{p}(0)}{\omega_H}\sin(\omega_H t). \tag{E.9}$$

We can use this expression to compute the expectation value $\langle \hat{x}^2(t) \rangle$

$$\begin{aligned} \langle \hat{x}^2(t) \rangle = \langle \hat{x}^2(0) \rangle \cos^2(\omega_H t) + \frac{\langle \hat{p}^2(0) \rangle}{\omega_H^2}\sin^2(\omega_H t) \\ + \frac{1}{\omega_H}(\langle \hat{x}(0)\hat{p}(0) \rangle + \langle \hat{p}(0)\hat{x}(0) \rangle)\cos(\omega_H t)\sin(\omega_H t). \end{aligned} \tag{E.10}$$

The next step is calculating the expectation values in Equation (E.10). Since these values will be used in later expressions, we will compute them for a general time t and then substitute for $t = 0$. This leads to

$$\langle \hat{x}^2(t) \rangle = \sqrt{\frac{\omega(t)}{\pi}} \int x^2 e^{-2\Re(z(t))x^2}\, dx = \frac{1}{2\omega(t)}, \tag{E.11}$$

$$\begin{aligned}\langle \hat{p}^2(t) \rangle &= \sqrt{\frac{\omega(t)}{\pi}} \int e^{-z^*(t)x^2} \left(-\partial_x^2 \left(e^{-z(t)x^2}\right)\right) dx \\ &= \sqrt{\frac{\omega(t)}{\pi}} 2z(t) \int (1 - 2z(t)x^2) e^{-2\Re(z(t))x^2}\, dx = \frac{\omega(t)}{2} + \frac{2\beta(t)^2}{\omega},\end{aligned} \tag{E.12}$$

$$\begin{aligned}\langle \hat{x}(t)\hat{p}(t) \rangle &= \sqrt{\frac{\omega(t)}{\pi}} \int e^{-z^*(t)x^2} x \left(-i\partial_x e^{-z(t)x^2}\right) dx \\ &= i\sqrt{\frac{\omega(t)}{\pi}} 2z(t) \int x^2 e^{-2\Re(z(t))x^2}\, dx = i\frac{z(t)}{\omega(t)},\end{aligned} \tag{E.13}$$

$$\begin{aligned}\langle \hat{p}(t)\hat{x}(t) \rangle &= \sqrt{\frac{\omega(t)}{\pi}} \int e^{-z^*(t)x^2} \left(-i\partial_x \left(x e^{-z(t)x^2}\right)\right) dx \\ &= \sqrt{\frac{\omega(t)}{\pi}} (-i) \int (1 - 2z(t)x^2) e^{-2\Re(z(t))x^2}\, dx = i\frac{z(t)}{\omega(t)} - i.\end{aligned} \tag{E.14}$$

For $\beta(0) = 0$ with $\omega(0) = \omega_0$

$$\langle \hat{x}^2(0) \rangle = \frac{1}{2\omega_0}, \quad \langle \hat{x}(0)\hat{p}(0) \rangle + \langle \hat{p}(0)\hat{x}(0) \rangle = 0, \quad \langle \hat{p}^2(0) \rangle = \frac{\omega_0}{2}. \tag{E.15}$$

Substituting in Equation (E.10) leads to

$$\langle \hat{x}^2(t) \rangle = \frac{1}{2\omega_0} \cos^2(\omega_H t) + \frac{\omega_0}{2\omega_H^2} \sin^2(\omega_H t). \tag{E.16}$$

and combining this expression with Equation (E.3) results in

$$\omega(t) = \left(\frac{1}{\omega_0} \cos^2(\omega_H t) + \frac{\omega_0}{\omega_H^2} \sin^2(\omega_H t)\right)^{-1}, \tag{E.17}$$

The next step to obtain the evolved state $\psi(x,t)$ is to determine $\beta(t)$.

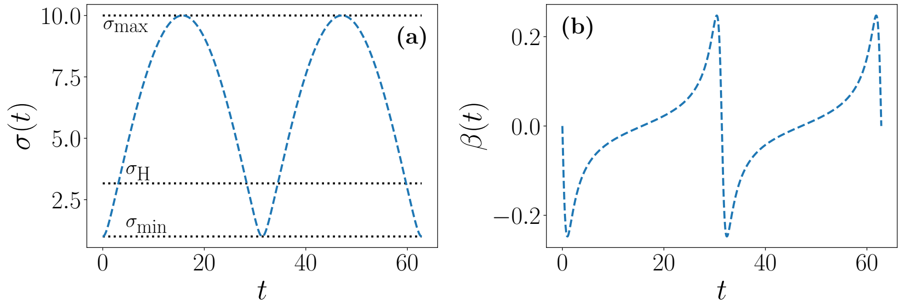


Figure E.1: Analytic expression of the parameters of the harmonic oscillator quantum quench. Analytic solution the quantum quench evolution under a harmonic potential (6.2) with $\omega_0 = 1$ and $\omega_H = 0.1$. (a) $\sigma(t)$ (6.5). (b) $\beta(t)$ (6.6).

Combining Eqs. (E.13) and (E.14)

$$\begin{aligned}\langle x(t)p(t)\rangle + \langle p(t)x(t)\rangle & \\ &= -2\frac{\beta(t)}{\omega(t)} \\ &= \left(-\omega_H\langle x^2(0)\rangle + \frac{\langle p^2(0)\rangle}{\omega_H}\right) 2\sin(\omega_H t)\cos(\omega_H t),\end{aligned} \tag{E.18}$$

which leads to

$$\beta(t) = \frac{\omega(t)}{4}\left(\frac{\omega_H}{\omega_0} - \frac{\omega_0}{\omega_H}\right)\sin(2\omega_H t). \tag{E.19}$$

Once the expression of $\omega(t)$ (E.17) and $\beta(t)$ (E.19) are known, we can obtain the evolution of the quantum state from Eq (6.4). Figure E.1 shows the evolution of $\sigma(t)$ and $\beta(t)$ for a particle expansion with $\omega_H < \omega_0$. The system expands during the first time of the period from a Gaussian with $\sigma_0 = 1/\sqrt{\omega_0}$ at $t = 0$, which coincides with the minimum width $\sigma_{\rm min}$, to the maximum one $\sigma_{\rm max} = \sqrt{\omega_0}/\omega_H$ at time $t = 0.5\pi/\omega_H$. The maximum width Gaussian overcomes the one of the ground state of the new Hamiltonian with frequency ω_H. Then, it continues evolving until the initial state, following a periodic process. The imaginary phase $\beta(t)$ (Figure E.1(b)) also changes periodically, being zero at the initial, middle, and final points of the period, i.e., the states corresponding to the maximum and minimum widths are real.

Appendix F

One-step ε scaling with Δt

Figures 6.5(a)-(b) show a fit for the error ε with Δt for the methods in Section 6.3, for the finite difference and HDAF derivative approximation, respectively. Tables F.1 and F.2 contain the concrete numerical data of the fit $\varepsilon = C\Delta t^p$.

Method	C	p
Euler	2.29×10^{-2}	1.99
Improved Euler	3.37×10^{-2}	3.14
Runge-Kutta	2.20×10^{-2}	4.18
Crank-Nicolson	8.87×10^{0}	3.44
Arnoldi $n_v = 5$	2.04×10^{-3}	3.79
Arnoldi $n_v = 10$	7.88×10^{-4}	4.45

Table F.1: Function error ε (6.26) fit, $\varepsilon = C\Delta t^p$, for each method for a $n = 18$ discretization and finite difference approximation of the derivative.

Method	C	p
Euler	2.13×10^{-2}	2.00
Improved Euler	2.88×10^{-2}	3.00
Runge-Kutta	2.21×10^{-2}	4.97
Crank-Nicolson	2.52×10^{-1}	3.73
Arnoldi $n_v = 5$	1.02×10^{-3}	4.92
Arnoldi $n_v = 10$	1.16×10^{-5}	9.11
Split-step	4.94×10^{-7}	2.96

Table F.2: Function error ε (6.26) fit, $\varepsilon = C\Delta t^p$, for each method for a $n = 18$ discretization and HDAF approximation of the derivative.

Appendix G

Harmonic quantum quench evolution scaling

Figure 6.8 shows the error ε scaling and run time T of the harmonic quantum quench evolution with time for a $n = 20$ discretization. Tables G.1-G.4 contain the concrete numerical data of the fits $\varepsilon = Ct^p$ and $T = Ct^p$.

Δt	C	p
0.01	5.89×10^{-11}	0.71
0.1	6.00×10^{-9}	0.70
1.0	6.65×10^{-7}	0.68

Table G.1: Function error ε (6.26) fit, $\varepsilon = Ct^p$, for the split-step HDAF MPS method, using different step-sizes Δt and $n = 20$.

Δt	C	p
0.01	5.89×10^{-11}	0.71
0.1	6.00×10^{-9}	0.70
1.0	6.65×10^{-7}	0.68

Table G.2: Function error ε (6.26) fit, $\varepsilon = Ct^p$, for the split-step FFT method, using different step-sizes Δt and $n = 20$.

Δt	C	p
0.01	3.19	1.35
0.1	0.44	1.34
1.0	0.03	1.42

Table G.3: Run time fit, $T = Ct^p$, for the split-step HDAF MPS method, using different step-sizes Δt and $n = 20$.

Δt	C	p
0.01	7.66	1.00
0.1	0.82	0.99
1.0	0.09	0.95

Table G.4: Run time fit, $T = Ct^p$, for the split-step FFT method, using different step-sizes Δt.

Bibliography

[1] Jens Koch, Terri M. Yu, Jay Gambetta, A. A. Houck, D. I. Schuster, J. Majer, Alexandre Blais, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf. Charge-insensitive qubit design derived from the Cooper pair box. *Physical Review A*, 76(4), 2007.

[2] J. E. Mooij, T. P. Orlando, Leonid Levitov, Lin Tian, Caspar H. van der Wal, and Seth Lloyd. Josephson persistent-current qubit. *Science*, 285(5430):1036–1039, 1999.

[3] Juan José García Ripoll. *Quantum Information and Quantum Optics with Superconducting Circuits.* Cambridge University Press, 2022.

[4] Carlos Gonzalez-Ballestero, Markus Aspelmeyer, Lukas Novotny, Romain Quidant, and Orioul Romero-Isart. Levitodynamics: Levitation and control of microscopic objects in vacuum. *Science*, 374(6564):eabg3027, 2021.

[5] Dominic W. Berry. High-order quantum algorithm for solving linear differential equations. *Journal of Physics A: Mathematical and Theoretical*, 47(10):105301, 2014.

[6] Ashley Montanaro and Sam Pallister. Quantum algorithms and the finite element method. *Physical Review A*, 93(3), 2016.

[7] Dominic W. Berry, Andrew M. Childs, Aaron Ostrander, and Guoming Wang. Quantum Algorithm for Linear Differential Equations with Exponentially Improved Dependence on Precision. *Communications in Mathematical Physics*, 356(3):1057–1081, 2017.

[8] Andrew M. Childs, Jin-Peng Liu, and Aaron Ostrander. High-precision quantum algorithms for partial differential equations. *Quantum*, 5:574, 2021.

[9] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum Algorithm for Linear Systems of Equations. *Physical Review Letters*, 103(15), 2009.

[10] Yudong Cao, Anargyros Papageorgiou, Iasonas Petras, Joseph Traub, and Sabre Kais. Quantum algorithm and circuit design solving the Poisson equation. *New Journal of Physics*, 15(1):013021, 2013.

[11] Noah Linden, Ashley Montanaro, and Changpeng Shao. Quantum vs. Classical Algorithms for Solving the Heat Equation. *Communications in Mathematical Physics*, 395(2):601–641, 2022.

[12] Bolesław Kacewicz. Almost optimal solution of initial-value problems by randomized and quantum algorithms. *Journal of Complexity*, 22(5):676–690, 2006.

[13] Frank Gaitan. Finding flows of a Navier–Stokes fluid through quantum computing. *npj Quantum Information*, 6(1), 2020.

[14] Frank Gaitan. Finding Solutions of the Navier-Stokes Equations through Quantum Computing—Recent Progress, a Generalization, and Next Steps Forward. *Advanced Quantum Technologies*, 4(10):2100055, 2021.

[15] Furkan Oz, Rohit K. S. S. Vuppala, Kursat Kara, and Frank Gaitan. Solving Burgers' equation with quantum computing. *Quantum Information Processing*, 21(1), 2021.

[16] Furkan Oz, Omer San, and Kursat Kara. An efficient quantum partial differential equation solver with Chebyshev points. *Scientific Reports*, 13(1), 2023.

[17] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation, 2002.

[18] John Preskill. Quantum Computing in the NISQ era and beyond. *Quantum*, 2:79, 2018.

[19] Paula García-Molina, Ana Martin, Mikel Garcia de Andoin, and Mikel Sanz. Mitigating noise in digital and digital–analog quantum computation, 2024.

[20] Jarrod R. McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18(2):023023, 2016.

[21] Kunal Sharma, Sumeet Khatri, M. Cerezo, and Patrick J. Coles. Noise resilience of variational quantum compiling. *New Journal of Physics*, 22(4):043006, 2020.

[22] Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C. Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R. McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick J. Coles. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, 2021.

[23] Michael Lubasch, Jaewoo Joo, Pierre Moinier, Martin Kiffner, and Dieter Jaksch. Variational quantum algorithms for nonlinear problems. *Physical Review A*, 101(1), 2020.

[24] Sam McArdle, Tyson Jones, Suguru Endo, Ying Li, Simon C. Benjamin, and Xiao Yuan. Variational ansatz-based quantum simulation of imaginary time evolution. *npj Quantum Information*, 5(1), 2019.

[25] Oleksandr Kyriienko, Annie E. Paine, and Vincent E. Elfving. Solving nonlinear differential equations with differentiable quantum circuits. *Phys. Rev. A*, 103:052416, 2021.

[26] Martin Knudsen and Christian B. Mendl. Solving Differential Equations via Continuous-Variable Quantum Computers. *arXiv e-prints*, arXiv:2012.12220, 2020.

[27] Filipe Fontanela, Antoine Jacquier, and Mugad Oumgari. Short Communication: A Quantum Algorithm for Linear PDEs Arising in Finance. *SIAM Journal on Financial Mathematics*, 12(4):SC98–SC114, 2021.

[28] Albert J. Pool, Alejandro D. Somoza, Michael Lubasch, and Birger Horstmann. Solving partial differential equations using a quantum computer. In *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 864–866, 2022.

[29] Hedayat Alghassi, Amol Deshmukh, Noelle Ibrahim, Nicolas Robles, Stefan Woerner, and Christa Zoufal. A variational quantum algorithm for the Feynman-Kac formula. *Quantum*, 6:730, 2022.

[30] Kenji Kubo, Yuya O. Nakagawa, Suguru Endo, and Shota Nagayama. Variational quantum simulations of stochastic differential equations. *Phys. Rev. A*, 103:052425, 2021.

[31] Hai-Ling Liu, Yu-Sen Wu, Lin-Chun Wan, Shi-Jie Pan, Su-Juan Qin, Fei Gao, and Qiao-Yan Wen. Variational quantum algorithm for the Poisson equation. *Phys. Rev. A*, 104:022418, 2021.

[32] Yuki Sato, Ruho Kondo, Satoshi Koide, Hideki Takamatsu, and Nobuyuki Imoto. Variational quantum algorithm based on the minimum potential energy for solving the Poisson equation. *Phys. Rev. A*, 104:052409, 2021.

[33] Reuben Demirdjian, Daniel Gunlycke, Carolyn A. Reynolds, James D. Doyle, and Sergio Tafur. Variational quantum solutions to the advection–diffusion equation for applications in fluid dynamics. *Quantum Information Processing*, 21(9), 2022.

[34] Nikita M. Guseynov, A. A. Zhukov, Walter V. Pogosov, and A. V. Lebedev. Depth analysis of variational quantum algorithms for the heat equation. *Phys. Rev. A*, 107:052422, 2023.

[35] Dieter Jaksch, Peyman Givi, Andrew J. Daley, and Thomas Rung. Variational Quantum Algorithms for Computational Fluid Dynamics. *AIAA Journal*, 61(5):1885–1894, 2023.

[36] Fong Yew Leong, Dax Enshan Koh, Wei-Bin Ewe, and Jian Feng Kong. Variational quantum simulation of partial differential equations: applications in colloidal transport. *International Journal of Numerical Methods for Heat &amp; Fluid Flow*, 33(11):3669–3690, 2023.

[37] Román Orús. A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Annals of Physics*, 349:117–158, 2014.

[38] Juan José García-Ripoll. Quantum-inspired algorithms for multivariate analysis: from interpolation to partial differential equations. *Quantum*, 5:431, 2021.

[39] Bernhard Jobst, Kevin Shen, Carlos A. Riofrío, Elvira Shishenina, and Frank Pollmann. Efficient MPS representations and quantum circuits from the Fourier modes of classical image data. *Quantum*, 8:1544, December 2024.

[40] Alex Gorodetsky, Sertac Karaman, and Youssef Marzouk. High-dimensional stochastic optimal control using continuous tensor decompositions. *The International Journal of Robotics Research*, 37(2-3):340–377, 2018.

[41] Sergey Dolgov, Dante Kalise, and Karl K. Kunisch. Tensor Decomposition Methods for High-dimensional Hamilton–Jacobi–Bellman Equations. *SIAM Journal on Scientific Computing*, 43(3):A1625–A1650, 2021.

[42] Mathias Oster, Leon Sallandt, and Reinhold Schneider. Approximating the Stationary Bellman Equation by Hierarchical Tensor Products. *Journal of Computational Mathematics*, 42(3):638–661, 2024.

[43] Alec Dektor, Abram Rodgers, and Daniele Venturi. Rank-Adaptive Tensor Methods for High-Dimensional Nonlinear PDEs. *Journal of Scientific Computing*, 88(2), 2021.

[44] Michael Lubasch, Pierre Moinier, and Dieter Jaksch. Multigrid renormalization. *Journal of Computational Physics*, 372:587–602, 2018.

[45] Nikita Gourianov, Michael Lubasch, Sergey Dolgov, Quincy Y. van den Berg, Hessam Babaee, Peyman Givi, Martin Kiffner, and Dieter Jaksch. A quantum-inspired approach to exploit turbulence structures. *Nature Computational Science*, 2(1):30–37, 2022.

[46] Paula García-Molina, Javier Rodríguez-Mediavilla, and Juan José García-Ripoll. Quantum Fourier analysis for multivariate functions and applications to a class of Schrödinger-type partial differential equations. *Phys. Rev. A*, 105:012433, 2022.

[47] Christof Zalka. Simulating quantum systems on a quantum computer. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 454(1969):313–322, 1998.

[48] Lov Grover and Terry Rudolph. Creating superpositions that correspond to efficiently integrable probability distributions. *arXiv e-prints*, pages quant–ph/0208112, 2002.

[49] Paula García-Molina, Luca Tagliacozzo, and Juan José García-Ripoll. Global optimization of MPS in quantum-inspired numerical analysis. *arXiv e-prints*, arXiv:2303.09430, 2023.

[50] Jorge Gidi, Paula García-Molina, Luca Tagliacozzo, and Juan José García-Ripoll. Pseudospectral method for solving PDEs using Matrix Product States. *arXiv e-prints*, arXiv:2409.02916, 2024.

[51] Juan José García-Ripoll, Paula García-Molina, Burcin Danaci, Juan José Rodríguez-Aldavero, and Jorge Gidi. SElf-Explaining Matrix-Product-State library. https://github.com/juanjosegarciaripoll/seemps2/tree/e3fafea8546d6c194b4650cd77741d5024f4c9ae, 2024.

[52] Juan José Rodríguez-Aldavero, Paula García-Molina, Luca Tagliacozzo, and Juan José García-Ripoll. Chebyshev approximation and composition of functions in matrix product states for quantum-inspired numerical analysis. *arXiv e-prints*, arXiv:2407.09609, 2024.

[53] John C. Mason and David C. Handscomb. *Chebyshev Polynomials*. CRC Press, Boca Raton; London New York, 2003.

[54] Lloyd N. Trefethen. *Approximation Theory and Approximation Practice, Extended Edition*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2019.

[55] Ivan Oseledets and Eugene Tyrtyshnikov. TT-cross approximation for multidimensional arrays. *Linear Algebra and its Applications*, 432(1):70–88, 2010.

[56] Iordanis Kerenidis and Alessandro Luongo. Classification of the MNIST data set with quantum slow feature analysis. *Phys. Rev. A*, 101:062327, 2020.

[57] Ryan LaRose and Brian Coyle. Robust data encodings for quantum classifiers. *Phys. Rev. A*, 102:032420, 2020.

[58] Maria Schuld. Supervised quantum machine learning models are kernel methods. *arXiv e-prints*, arXiv:2101.11020, 2021.

[59] Hsin-Yuan Huang, Michael Broughton, Masoud Mohseni, Ryan Babbush, Sergio Boixo, Hartmut Neven, and Jarrod R. McClean. Power of data in quantum machine learning. *Nature Communications*, 12(1):2631, 2021.

[60] Javier Gonzalez-Conde, Thomas W. Watts, Pablo Rodriguez-Grasa, and Mikel Sanz. Efficient quantum amplitude encoding of polynomial functions. *Quantum*, 8:1297, 2024.

[61] Harry Nyquist. Certain Topics in Telegraph Transmission Theory. *Transactions of the American Institute of Electrical Engineers*, 47(2):617–644, 1928.

[62] Claude E. Shannon. Communication in the Presence of Noise. *Proceedings of the IRE*, 37(1):10–21, 1949.

[63] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.

[64] Feihong Shen and Jun Liu. Quantum Fourier Convolutional Network. *ACM Trans. Multimedia Comput. Commun. Appl.*, 19(1), 2023.

[65] Donlad M. Monro. Interpolation by fast Fourier and Chebyshev transforms. *International Journal for Numerical Methods in Engineering*, 14(11):1679–1692, 1979.

[66] Pascal Frey. Simulación numérica de problemas de EDP sobre dominios complejos, 2017. Course Material.

[67] David A. Kopriva. Implementing Spectral Methods for Partial Differential Equations. In *Implementing Spectral Methods for Partial Differential Equations*, pages 3–22. Springer Netherlands, 2009.

[68] Claudio Canuto, M. Youssuff Hussaini, Alfio Quarteroni, and Thomas A. Zang. Spectral Methods: Fundamentals in Single Domains. In *Spectral Methods: Fundamentals in Single Domains.*, pages 41–68, 267–275. Springer Berlin Heidelberg, 2006.

[69] Heinz-Otto Kreiss and Joseph Oliger. Stability of the Fourier Method. *SIAM Journal on Numerical Analysis*, 16(3):421–433, 1979.

[70] Eitan Tadmor. The Exponential Accuracy of Fourier and Chebyshev Differencing Methods. *SIAM Journal on Numerical Analysis*, 23(1):1–10, 1986.

[71] Google Quantum AI. Choosing hardware for your qsim simulation, 2024. Accessed: 2024-10-25.

[72] J. Ignacio Cirac and Frank Verstraete. Renormalization and tensor product states in spin chains and lattices. *Journal of Physics A: Mathematical and Theoretical*, 42(50):504004, 2009.

[73] Mari Carmen Bañuls. Tensor network algorithms: A route map. *Annual Review of Condensed Matter Physics*, 14(Volume 14, 2023):173–191, 2023.

[74] Örs Legeza and Jeno Sólyom. Quantum data compression, quantum information generation, and the density-matrix renormalization-group method. *Phys. Rev. B*, 70(20):205118, 2004.

[75] Guifré Vidal, José Ignacio Latorre, Enrique Rico, and Alexei Kitaev. Entanglement in Quantum Critical Phenomena. *Phys. Rev. Lett.*, 90(22):227902, 2003.

[76] Pasquale Calabrese and John Cardy. Entanglement entropy and quantum field theory. *Journal of Statistical Mechanics: Theory and Experiment*, 2004(06):P06002, 2004.

[77] Mark Srednicki. Entropy and area. *Phys. Rev. Lett.*, 71(5):666–669, 1993.

[78] Martin B. Plenio, Jens Eisert, Julian Dreißig, and Marcus Cramer. Entropy, Entanglement, and Area: Analytical Results for Harmonic Lattice Systems. *Phys. Rev. Lett.*, 94(6):060503, 2005.

[79] David Perez-Garcia, Frank Verstraete, Michael. M. Wolf, and J. Ignacio Cirac. Matrix product state representations. *Quantum Info. Comput.*, 7(5):401–430, 2007.

[80] Hendrik Anthony Kramers and Gregory Hugh Wannier. Statistics of the two-dimensional ferromagnet. part ii. *Phys. Rev.*, 60:263–276, 1941.

[81] J. Ignacio Cirac, David Pérez-García, Norbert Schuch, and Frank Verstraete. Matrix product states and projected entangled pair states: Concepts, symmetries, theorems. *Rev. Mod. Phys.*, 93:045003, 2021.

[82] Rodney J. Baxter. Dimers on a rectangular lattice. *Journal of Mathematical Physics*, 9(4):650–654, 1968.

[83] Rodney J. Baxter. Corner transfer matrices. *Physica A: Statistical Mechanics and its Applications*, 106(1):18–27, 1981.

[84] Luigi Accardi. Topics in quantum probability. *Physics Reports*, 77(3):169–192, 1981.

[85] Ian Affleck, Tom Kennedy, Elliott H. Lieb, and Hal Tasaki. Rigorous results on valence-bond ground states in antiferromagnets. *Phys. Rev. Lett.*, 59:799–802, 1987.

[86] Ian Affleck, Tom Kennedy, Elliott H. Lieb, and Hal Tasaki. *Valence Bond Ground States in Isotropic Quantum Antiferromagnets*, page 253–304. Springer Berlin Heidelberg, 1988.

[87] Steven R. White. Density matrix formulation for quantum renormalization groups. *Physical Review Letters*, 69(19):2863–2866, 1992.

[88] Steven R. White. Density-matrix algorithms for quantum renormalization groups. *Phys. Rev. B*, 48(14):10345–10356, 1993.

[89] Stellan Östlund and Stefan Rommer. Thermodynamic limit of density matrix renormalization. *Phys. Rev. Lett.*, 75:3537–3540, 1995.

[90] Jorge Dukelsky, Miguel A. Martín-Delgado, Tomotoshi Nishino, and Germán Sierra. Equivalence of the variational matrix product method and the density matrix renormalization group applied to spin chains. *Europhysics Letters*, 43(4):457, 1998.

[91] Frank Verstraete, Diego Porras, and J. Ignacio Cirac. Density Matrix Renormalization Group and Periodic Boundary Conditions: A Quantum Information Perspective. *Phys. Rev. Lett.*, 93:227205, 2004.

[92] Ryota Suzuki and Akihisa Koga. Cluster mean-field approach with density matrix renormalization group: Application to the hard-core bosonic Hubbard model on a triangular lattice. In *Proceedings of the International Conference on Strongly Correlated Electron Systems (SCES2013)*. Journal of the Physical Society of Japan, 2014.

[93] Aaron Szasz, Johannes Motruk, Michael P. Zaletel, and Joel E. Moore. Chiral Spin Liquid Phase of the Triangular Lattice Hubbard Model: A Density Matrix Renormalization Group Study. *Physical Review X*, 10(2):021042, 2020.

[94] Debashree Ghosh, Johannes Hachmann, Takeshi Yanai, and Garnet Kin-Lic Chan. Orbital optimization in the density matrix renormalization group, with applications to polyenes and β-carotene. *The Journal of Chemical Physics*, 128(14):144117, 2008.

[95] Alberto Baiardi and Markus Reiher. The density matrix renormalization group in chemistry and molecular physics: Recent developments and new challenges. *The Journal of Chemical Physics*, 152(4):040903, 2020.

[96] Haibo Ma, U. Schollwöck, and Zhigang Shuai. *Density matrix renormalization group (Dmrg)-based approaches in computational chemistry.* Elsevier, Amsterdam, Netherlands, 2022.

[97] Frank Verstraete, Juan José García-Ripoll, and J. Ignacio Cirac. Matrix Product Density Operators: Simulation of Finite-Temperature and Dissipative Systems. *Phys. Rev. Lett.*, 93(20):207204, 2004.

[98] Adrian E. Feiguin and Steven R. White. Time-step targeting methods for real-time dynamics using the density matrix renormalization group. *Phys. Rev. B*, 72:020404, 2005.

[99] Gregory M. Crosswhite, A. C. Doherty, and Guifré Vidal. Applying matrix product operators to model systems with long-range interactions. *Phys. Rev. B*, 78:035116, 2008.

[100] Yaoyun. Shi, Luming. Duan, and Guifré Vidal. Classical simulation of quantum many-body systems with a tree tensor network. *Phys. Rev. A*, 74:022320, 2006.

[101] Luca Tagliacozzo, Glen Evenbly, and Guifré Vidal. Simulation of two-dimensional quantum systems using a tree tensor network that exploits the entropic area law. *Phys. Rev. B*, 80:235127, 2009.

[102] Katharine Hyatt and E. M. Stoudenmire. DMRG Approach to Optimizing Two-Dimensional Tensor Networks. *arXiv e-prints*, arXiv:1908.08833, 2020.

[103] Guifré Vidal. Efficient Simulation of One-Dimensional Quantum Many-Body Systems. *Phys. Rev. Lett.*, 93(4):040502, 2004.

[104] Michael Zwolak and Guifré Vidal. Mixed-State Dynamics in One-Dimensional Quantum Lattice Systems: A Time-Dependent Superoperator Renormalization Algorithm. *Phys. Rev. Lett.*, 93(20):207205, 2004.

[105] Jutho Haegeman, J. Ignacio Cirac, Tobias J. Osborne, Iztok Pižorn, Henri Verschelde, and Frank Verstraete. Time-Dependent Variational Principle for Quantum Lattices. *Physical Review Letters*, 107(7), 2011.

[106] Laurens Vanderstraeten, Jutho Haegeman, and Frank Verstraete. Tangent-space methods for uniform matrix product states. *SciPost Physics Lecture Notes*, 2019.

[107] Sebastian Paeckel, Thomas Köhler, Andreas Swoboda, Salvatore R. Manmana, Ulrich Schollwöck, and Claudius Hubig. Time-evolution methods for matrix-product states. *Annals of Physics*, 411:167998, 2019.

[108] Salvatore R. Manmana. Time evolution of one-dimensional Quantum Many Body Systems. In *AIP Conference Proceedings*. AIP, 2005.

[109] Juan José García-Ripoll. Time evolution of Matrix Product States. *New Journal of Physics*, 8(12):305, 2006.

[110] Andrew J. Daley, Corinna Kollath, Ulrich Schollwöck, and Guifré Vidal. Time-dependent density-matrix renormalization-group using adaptive effective Hilbert spaces. *Journal of Statistical Mechanics: Theory and Experiment*, 2004(04):P04005, 2004.

[111] Steven R. White and Adrian E. Feiguin. Real-Time Evolution Using the Density Matrix Renormalization Group. *Phys. Rev. Lett.*, 93:076401, 2004.

[112] Mari Carmen Bañuls, J. Ignacio Cirac, and Michael M. Wolf. Entanglement in fermionic systems. *Phys. Rev. A*, 76:022311, 2007.

[113] Mari Carmen Bañuls, Mathew B. Hastings, Frank Verstraete, and J. Ignacio Cirac. Matrix product states for dynamical simulation of infinite chains. *Phys. Rev. Lett.*, 102:240603, 2009.

[114] Thomas Barthel, Carlos Pineda, and Jens Eisert. Contraction of fermionic operator circuits and the simulation of strongly correlated fermions. *Phys. Rev. A*, 80:042333, 2009.

[115] Philippe Corboz and Guifré Vidal. Fermionic multiscale entanglement renormalization ansatz. *Phys. Rev. B*, 80:165129, 2009.

[116] Philippe Corboz, Glen Evenbly, Frank Verstraete, and Guifré Vidal. Simulation of interacting fermions with entanglement renormalization. *Phys. Rev. A*, 81:010303, 2010.

[117] Philipp Hauke, Fernando M Cucchietti, Alexander Müller-Hermes, Mari Carmen Bañuls, J Ignacio Cirac, and Maciej Lewenstein. Complete devil's staircase and crystal–superfluid transitions in a dipolarXXZspin chain: a trapped ion quantum simulation. *New Journal of Physics*, 12(11):113037, 2010.

[118] Philippe Corboz, Román Orús, Bela Bauer, and Guifré Vidal. Simulation of strongly correlated fermions in two spatial dimensions with fermionic projected entangled-pair states. *Phys. Rev. B*, 81:165104, 2010.

[119] Christina V. Kraus, Norbert Schuch, Frank Verstraete, and J. Ignacio Cirac. Fermionic projected entangled pair states. *Phys. Rev. A*, 81:052338, 2010.

[120] Iztok Pižorn and Frank Verstraete. Fermionic implementation of projected entangled pair states algorithm. *Phys. Rev. B*, 81:245110, 2010.

[121] Román Orús. Advances on tensor network theory: symmetries, fermions, entanglement, and holography. *The European Physical Journal B*, 87(11), 2014.

[122] Mari Carmen Bañuls, Michal P. Heller, Karl Jansen, Johannes Knaute, and Viktor Svensson. From spin chains to real-time thermal field theory using tensor networks. *Phys. Rev. Res.*, 2:033301, 2020.

[123] Frank Verstraete, Michael M. Wolf, David Perez-Garcia, and J. Ignacio Cirac. Criticality, the Area Law, and the Computational Power of Projected Entangled Pair States. *Phys. Rev. Lett.*, 96:220601, 2006.

[124] Norbert Schuch, David Pérez-García, and Ignacio Cirac. Classifying quantum phases using matrix product states and projected entangled pair states. *Phys. Rev. B*, 84:165139, 2011.

[125] Xie Chen, Zheng-Cheng Gu, and Xiao-Gang Wen. Classification of gapped symmetric phases in one-dimensional spin systems. *Phys. Rev. B*, 83:035107, 2011.

[126] Mari Carmen Bañuls, Krzysztof Cichy, J. Ignacio Cirac, Karl Jansen, and Stefan Kühn. Density Induced Phase Transitions in the Schwinger Model: A Study with Matrix Product States. *Phys. Rev. Lett.*, 118:071601, 2017.

[127] Ivan V. Oseledets. Tensor-Train Decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

[128] Boris N. Khoromskij. O(dlog N)-Quantics Approximation of N-d Tensors in High-Dimensional Numerical Modeling. *Constructive Approximation*, 34(2):257–280, 2011.

[129] William Huggins, Piyush Patil, Bradley Mitchell, K Birgitta Whaley, and E Miles Stoudenmire. Towards quantum machine learning with tensor networks. *Quantum Science and Technology*, 4(2):024001, 2019.

[130] Song Cheng, Lei Wang, Tao Xiang, and Pan Zhang. Tree tensor networks for generative modeling. *Phys. Rev. B*, 99:155131, 2019.

[131] Yinchong Yang, Denis Krompass, and Volker Tresp. Tensor-Train Recurrent Neural Networks for Video Classification. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3891–3900. PMLR, 2017.

[132] Cong Chen, Kim Batselier, Ching-Yun Ko, and Ngai Wong. A Support Tensor Train Machine. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2019.

[133] Alexander Novikov, Pavel Izmailov, Valentin Khrulkov, Michael Figurnov, and Ivan Oseledets. Tensor Train Decomposition on TensorFlow (T3F). *Journal of Machine Learning Research*, 21(30):1–7, 2020.

[134] Konstantin Sozykin, Andrei Chertkov, Roman Schutski, Anh-Huy Phan, Andrzej S Chichocki, and Ivan Oseledets. TTOpt: A Maximum Volume Quantized Tensor Train-based Optimization and its Application to Reinforcement Learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 26052–26065. Curran Associates, Inc., 2022.

[135] Alejandro Pozas-Kerstjens, Senaida Hernández-Santana, José Ramón Pareja Monturiol, Marco Castrillón López, Giannicola Scarpa, Carlos E. González-Guillén, and David Pérez-García. Privacy-preserving machine learning with tensor networks. *Quantum*, 8:1425, 2024.

[136] José Ramón Pareja Monturiol, David Pérez-García, and Alejandro Pozas-Kerstjens. Tensorkrowch: Smooth integration of tensor networks in machine learning. *Quantum*, 8:1364, 2024.

[137] Alex Gorodetsky, Sertac Karaman, and Youssef Marzouk. A continuous analogue of the tensor-train decomposition. *Computer methods in applied mechanics and engineering*, 347:59–84, 2019.

[138] Roger Penrose. Applications of negative dimensional tensors. *Combinatorial Mathematics and its Applications*, 1971.

[139] Guifré Vidal. Efficient Classical Simulation of Slightly Entangled Quantum Computations. *Phys. Rev. Lett.*, 91:147902, 2003.

[140] Michael M. Wolf, Frank Verstraete, Matthew B. Hastings, and J. Ignacio Cirac. Area Laws in Quantum Systems: Mutual Information and Correlations. *Phys. Rev. Lett.*, 100(7):070502, 2008.

[141] Glen Evenbly and Guifré Vidal. Tensor Network States and Geometry. *Journal of Statistical Physics*, 145(4):891–918, 2011.

[142] Ulrich Schollwöck. The density-matrix renormalization group in the age of matrix product states. *Annals of Physics*, 326(1):96–192, 2011.

[143] Garnet Kin-Lic Chan, Anna Keselman, Naoki Nakatani, Zhendong Li, and Steven R. White. Matrix product operators, matrix product states, and ab initio density matrix renormalization group algorithms. *The Journal of Chemical Physics*, 145(1):014102, 2016.

[144] Frank Verstraete and J. Ignacio Cirac. Matrix product states represent ground states faithfully. *Phys. Rev. B*, 73:094423, 2006.

[145] Michael Lindsey. Multiscale Interpolative Construction of Quantized Tensor Trains. *arXiv e-prints*, arXiv:2311.12554, 2023.

[146] Ivan Oseledets and Eugene Tyrtyshnikov. TT-cross approximation for multidimensional arrays. *Linear Algebra and its Applications*, 432(1):70–88, 2010.

[147] Sergey Dolgov and Dmitry Savostyanov. Parallel cross interpolation for high-precision calculation of high-dimensional integrals. *Computer Physics Communications*, 246:106869, 2020.

[148] Jielun Chen, E.M. Stoudenmire, and Steven R. White. Quantum Fourier Transform Has Small Entanglement. *PRX Quantum*, 4:040318, 2023.

[149] David K. Hoffman, Naresh Nayar, Omar A. Sharafeddin, and D. J. Kouri. Analytic banded approximation for the discretized free propagator. *The Journal of Physical Chemistry*, 95(21):8299–8305, 1991.

[150] Peter Groszkowski and Jens Koch. Scqubits: a Python package for superconducting qubits. *Quantum*, 5:583, 2021.

[151] Philipp Aumann, Tim Menke, William D Oliver, and Wolfgang Lechner. Circuitq: an open-source toolbox for superconducting circuits. *New Journal of Physics*, 24(9):093012, 2022.

[152] Taha Rajabzadeh, Zhaoyou Wang, Nathan Lee, Takuma Makihara, Yudan Guo, and Amir H. Safavi-Naeini. Analysis of arbitrary superconducting quantum circuits accompanied by a python package: Sqcircuit. *Quantum*, 7:1118, 2023.

[153] Youcef Saad and Martin H. Schultz. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986.

[154] Yousef Saad. *Iterative Methods for Sparse Linear Systems.* Society for Industrial and Applied Mathematics, second edition, 2003.

[155] Gene H. Golub and Charles F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, MD, 4th edition, 2013.

[156] Claudio Canuto, M. Yousuff Hussaini, Alfio Quarteroni, and Thomas A. Zang. *Spectral Methods: Evolution to Complex Geometries and Applications to Fluid Dynamics.* Springer, Berlin, 2007.

[157] Randall J. LeVeque. *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems.* SIAM, Philadelphia, PA, 2007.

[158] Michel H. Devoret, Andreas Wallraff, and John M. Martinis. Superconducting qubits: A short review. *arXiv e-prints*, arXiv:cond-mat/0411174, 2004.

[159] Yuiga Nakamura, Yu. A. Pashkin, and Jaw-Shen Tsai. Coherent control of macroscopic quantum states in a single-Cooper-pair box. *Nature*, 398(6730):786–788, 1999.

[160] Edgar Bright Wilson, J. C. Decius, and Paul C. Cross. *Molecular Vibrations: The Theory of Infrared and Raman Vibrational Spectra.* McGraw-Hill, New York, 1955.

[161] Gilbert Grynberg, Alain Aspect, and Claude Fabre. *Introduction to Quantum Optics: From the Semi-classical Approach to Quantized Light.* Cambridge University Press, 2010.

[162] Juan José García Ripoll. *Quantum Information and Quantum Optics with Superconducting Circuits.* Cambridge University Press, 2022.

[163] T. P. Orlando, J. E. Mooij, Lin Tian, Caspar H. van der Wal, Leonid Levitov, Seth Lloyd, and Juan José Mazo. Superconducting persistent-current qubit. *Phys. Rev. B*, 60:15398–15413, 1999.

[164] Changpeng Shao. Computing eigenvalues of diagonalizable matrices on a quantum computer. *ACM Transactions on Quantum Computing*, 3(4), 2022.

[165] Jianxin Chen, Andrew M. Childs, and Shih-Han Hung. Quantum algorithm for multivariate polynomial interpolation. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 474(2209):20170480, 2018.

[166] Sarah K. Leyton and Tobias J. Osborne. A quantum algorithm to solve nonlinear differential equations. *arXiv e-prints*, arXiv:0812.4423, 2008.

[167] Pedro C. S. Costa, Stephen Jordan, and Aaron Ostrander. Quantum algorithm for simulating the wave equation. *Physical Review A*, 99(1), 2019.

[168] Lewis Wright, Conor Mc Keever, Jeremy T. First, Rory Johnston, Jeremy Tillay, Skylar Chaney, Matthias Rosenkranz, and Michael Lubasch. Noisy intermediate-scale quantum simulation of the one-dimensional wave equation. *arXiv e-prints*, arXiv:2402.19247, 2024.

[169] Adrien Suau, Gabriel Staffelbach, and Henri Calandra. Practical Quantum Computing: Solving the Wave Equation Using a Quantum Approach. *ACM Transactions on Quantum Computing*, 2(1):1–35, 2021.

[170] Jin-Peng Liu, Herman Øie Kolden, Hari K. Krovi, Nuno F. Loureiro, Konstantina Trivisa, and Andrew M. Childs. Efficient quantum algorithm for dissipative nonlinear differential equations. *Proceedings of the National Academy of Sciences*, 118(35):e2026805118, 2021.

[171] Seth Lloyd, Giacomo De Palma, Can Gokler, Bobak Kiani, Zi-Wen Liu, Milad Marvian, Felix Tennie, and Tim Palmer. Quantum algorithm for nonlinear differential equations. *arXiv e-prints*, arXiv:2011.06571, 2020.

[172] Tao Xin, Shijie Wei, Jianlian Cui, Junxiang Xiao, Iñigo Arrazola, Lucas Lamata, Xiangyu Kong, Dawei Lu, Enrique Solano, and Guilu Long. Quantum algorithm for solving linear differential equations: Theory and experiment. *Physical Review A*, 101(3), 2020.

[173] Benjamin Zanger, Christian B. Mendl, Martin Schulz, and Martin Schreiber. Quantum Algorithms for Solving Ordinary Differential Equations via Classical Integration Methods. *Quantum*, 5:502, 2021.

[174] Juan Miguel Arrazola, Timjan Kalajdzievski, Christian Weedbrook, and Seth Lloyd. Quantum algorithm for nonhomogeneous linear partial differential equations. *Physical Review A*, 100(3), 2019.

[175] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O'Brien. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 5(1), 2014.

[176] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M. Chow, and Jay M. Gambetta. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature*, 549(7671):242–246, 2017.

[177] Robert M. Parrish, Edward G. Hohenstein, Peter L. McMahon, and Todd J. Martínez. Quantum computation of electronic transitions using a variational quantum eigensolver. *Phys. Rev. Lett.*, 122:230401, 2019.

[178] Alexey Uvarov, Jacob D. Biamonte, and Dmitry Yudin. Variational quantum eigensolver for frustrated quantum systems. *Phys. Rev. B*, 102:075104, 2020.

[179] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv e-prints*, arXiv:1411.4028, 2014.

[180] Pontus Vikstål, Mattias Grönkvist, Marika Svensson, Martin Andersson, Göran Johansson, and Giulia Ferrini. Applying the quantum approximate optimization algorithm to the tail-assignment problem. *Phys. Rev. Appl.*, 14:034009, 2020.

[181] Utkarsh Azad, Bikash K. Behera, Emad A. Ahmed, Prasanta K. Panigrahi, and Ahmed Farouk. Solving vehicle routing problem using quantum approximate optimization algorithm. *IEEE Transactions on Intelligent Transportation Systems*, 24(7):7564–7573, 2023.

[182] Wenyang Qian, Robert A. M. Basili, Mary Mehrnoosh Eshaghian-Wilner, Ashfaq Khokhar, Glenn Luecke, and James P. Vary. Comparative Study of Variations in Quantum Approximate Optimization Algorithms for the Traveling Salesman Problem. *Entropy*, 25(8), 2023.

[183] Ajinkya Borle, Vincent E. Elfving, and Samuel J. Lomonaco. Quantum approximate optimization for hard problems in linear algebra. *SciPost Phys. Core*, 4:031, 2021.

[184] Pablo Díez-Valle, Jorge Luis-Hita, Senaida Hernández-Santana, Fernando Martínez-García, Álvaro Díaz-Fernández, Eva Andrés, Juan José García-Ripoll, Escolástico Sánchez-Martínez, and Diego Porras. Multiobjective variational quantum optimization for constrained problems: an application to cash handling. *Quantum Science and Technology*, 8(4):045009, 2023.

[185] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum Support Vector Machine for Big Data Classification. *Phys. Rev. Lett.*, 113:130503, 2014.

[186] Riccardo Mengoni and Alessandra Di Pierro. Kernel methods in quantum machine learning. *Quantum Machine Intelligence*, 1(3–4):65–71, 2019.

[187] Kushal Batra, Kimberley M. Zorn, Daniel H. Foil, Eni Minerali, Victor O. Gawriljuk, Thomas R. Lane, and Sean Ekins. Quantum machine learning algorithms for drug discovery applications. *Journal of Chemical Information and Modeling*, 61(6):2641–2647, 2021.

[188] Wen Guan, Gabriel Perdue, Arthur Pesah, Maria Schuld, Koji Terashi, Sofia Vallecorsa, and Jean-Roch Vlimant. Quantum machine learning in high energy physics. *Machine Learning: Science and Technology*, 2(1):011003, 2021.

[189] Jonathan Romero, Jonathan P Olson, and Alan Aspuru-Guzik. Quantum autoencoders for efficient compression of quantum data. *Quantum Science and Technology*, 2(4):045001, 2017.

[190] Alex Pepper, Nora Tischler, and Geoff J. Pryde. Experimental realization of a quantum autoencoder: The compression of qutrits via machine learning. *Phys. Rev. Lett.*, 122:060501, 2019.

[191] Dmytro Bondarenko and Polina Feldmann. Quantum autoencoders to denoise quantum data. *Phys. Rev. Lett.*, 124:130502, 2020.

[192] Xun Gao, Z.-Y. Zhang, and Luming Duan. A quantum machine learning algorithm based on generative models. *Science Advances*, 4(12):eaat9004, 2018.

[193] Junde Li, Rasit O. Topaloglu, and Swaroop Ghosh. Quantum generative models for small molecule drug discovery. *IEEE Transactions on Quantum Engineering*, 2:1–8, 2021.

[194] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. The quest for a quantum neural network. *Quantum Information Processing*, 13(11):2567–2586, 2014.

[195] Amira Abbas, David Sutter, Christa Zoufal, Aurelien Lucchi, Alessio Figalli, and Stefan Woerner. The power of quantum neural networks. *Nature Computational Science*, 1(6):403–409, 2021.

[196] Nouredine Zettili. *Quantum Mechanics: Concepts and Applications*. Wiley, english edition edition, 2022.

[197] James C. Spall. An overview of the simultaneous perturbation method for efficient optimization. *Johns Hopkins APL Technical Digest*, 19(4):482–492, 1998.

[198] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv e-prints*, arXiv:1412.6980, 2017.

[199] Kosuke Mitarai, Makoto Negoro, Masahiro Kitagawa, and Keisuke Fujii. Quantum circuit learning. *Physical Review A*, 98(3), 2018.

[200] Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. Evaluating analytic gradients on quantum hardware. *Physical Review A*, 99(3), 2019.

[201] Ville Bergholm, Josh Izaac, Maria Schuld, Christian Gogolin, Shahnawaz Ahmed, Vishnu Ajith, M. Sohaib Alam, Guillermo Alonso-Linaje, B. AkashNarayanan, Ali Asadi, Juan Miguel Arrazola, Utkarsh Azad, Sam Banning, Carsten Blank, Thomas R Bromley, Benjamin A. Cordier, Jack Ceroni, Alain Delgado, Olivia Di Matteo, Amintor Dusko, Tanya Garg, Diego Guala, Anthony Hayes, Ryan Hill, Aroosa Ijaz, Theodor Isacsson, David Ittah, Soran Jahangiri, Prateek Jain, Edward Jiang, Ankit Khandelwal, Korbinian Kottmann, Robert A. Lang, Christina Lee, Thomas Loke, Angus Lowe, Keri McKiernan, Johannes Jakob Meyer, J. A. Montañez-Barrera, Romain Moyard, Zeyue Niu, Lee James O'Riordan, Steven Oud, Ashish Panigrahi, Chae-Yeun Park, Daniel Polatajko, Nicolás Quesada, Chase Roberts, Nahum Sá, Isidor Schoch, Borun Shi, Shuli Shu, Sukin Sim, Arshpreet Singh, Ingrid Strandberg, Jay Soni, Antal Száva, Slimane Thabet, Rodrigo A. Vargas-Hernández, Trevor Vincent, Nicola Vitucci, Maurice Weber, David Wierichs, Roeland Wiersema, Moritz Willmann, Vincent Wong, Shaoming Zhang, and Nathan Killoran. Pennylane: Automatic differentiation of hybrid quantum-classical computations. *arXiv e-prints*, arXiv:1811.04968, 2022.

[202] Oscar Higgott, Daochen Wang, and Stephen Brierley. Variational quantum computation of excited states. *Quantum*, 3:156, 2019.

[203] Tyson Jones, Suguru Endo, Sam McArdle, Xiao Yuan, and Simon C. Benjamin. Variational quantum algorithms for discovering hamiltonian spectra. *Phys. Rev. A*, 99:062304, 2019.

[204] Ken M. Nakanishi, Kosuke Mitarai, and Keisuke Fujii. Subspace-search variational quantum eigensolver for excited states. *Phys. Rev. Res.*, 1:033062, 2019.

[205] Martin Larocca, Supanut Thanasilp, Samson Wang, Kunal Sharma, Jacob Biamonte, Patrick J. Coles, Lukasz Cincio, Jarrod R. McClean, Zoë Holmes, and Marco Cerezo. A review of barren plateaus in variational quantum computing. *arXiv e-prints*, arXiv:2405.00781, 2024.

[206] George S. Barron and Christopher J. Wood. Measurement error mitigation for variational quantum algorithms. *arXiv e-prints*, arXiv:2010.0852, 2020.

[207] Suguru Endo, Zhenyu Cai, Simon C. Benjamin, and Xiao Yuan. Hybrid quantum-classical algorithms and quantum error mitigation. *Journal of the Physical Society of Japan*, 90(3):032001, 2021.

[208] Ludmila Botelho, Adam Glos, Akash Kundu, Jarosław Adam Miszczak, Özlem Salehi, and Zoltán Zimborás. Error mitigation for variational quantum algorithms through mid-circuit measurements. *Phys. Rev. A*, 105:022441, 2022.

[209] Albert J. Pool, Alejandro D. Somoza, Conor Mc Keever, Michael Lubasch, and Birger Horstmann. Nonlinear dynamics as a ground-state solution on quantum computers. *Phys. Rev. Res.*, 6:033257, 2024.

[210] Jaewoo Joo and Hyungil Moon. Quantum variational PDE solver with machine learning. *arXiv e-prints*, arXiv:2109.09216, 2021.

[211] Sangeeta Yadav. QPDE: Quantum Neural Network Based Stabilization Parameter Prediction for Numerical Solvers for Partial Differential Equations. *AppliedMath*, 3(3):552–562, 2023.

[212] Enrico Rinaldi, Xizhi Han, Mohammad Hassan, Yuan Feng, Franco Nori, Michael McGuigan, and Masanori Hanada. Matrix-Model Simulations Using Quantum Computing, Deep Learning, and Lattice Monte Carlo. *PRX Quantum*, 3:010324, 2022.

[213] André Melo, Nathan Earnest-Noble, and Francesco Tacchino. Pulse-efficient quantum machine learning. *Quantum*, 7:1130, 2023.

[214] Kurudi V Vedavyasa and Ashok Kumar. Classification analysis of transition-metal chalcogenides and oxides using quantum machine learning. *arXiv e-prints*, arXiv:2405.18989, 2024.

[215] Jarrod R. McClean, Sergio Boixo, Vadim N. Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nature Communications*, 9(1), 2018.

[216] Kazuhiro Seki, Tomonori Shirakawa, and Seiji Yunoki. Symmetry-adapted variational quantum eigensolver. *Physical Review A*, 101(5), 2020.

[217] Shouvanik Chakrabarti, Rajiv Krishnakumar, Guglielmo Mazzola, Nikitas Stamatopoulos, Stefan Woerner, and William J. Zeng. A Threshold for Quantum Advantage in Derivative Pricing. *Quantum*, 5:463, 2021.

[218] Héctor Abraham, AduOffei, Rochisha Agarwal, Ismail Yunus Akhalwaya, and Gadi Aleksandrowicz et al. Qiskit: An Open-source Framework for Quantum Computing, 2019.

[219] Zhenyu Cai, Ryan Babbush, Simon C. Benjamin, Suguru Endo, William J. Huggins, Ying Li, Jarrod R. McClean, and Thomas E. O'Brien. Quantum error mitigation. *Rev. Mod. Phys.*, 95:045005, 2023.

[220] Simon J Devitt, William J Munro, and Kae Nemoto. Quantum error correction for beginners. *Reports on Progress in Physics*, 76(7):076001, 2013.

[221] Abhinav Kandala, Kristan Temme, Antonio D. Córcoles, Antonio Mezzacapo, Jerry M. Chow, and Jay M. Gambetta. Error mitigation extends the computational reach of a noisy quantum processor. *Nature*, 567(7749):491–495, 2019.

[222] Ryan Sweke, Frederik Wilde, Johannes Meyer, Maria Schuld, Paul K. Faehrmann, Barthélémy Meynard-Piganeau, and Jens Eisert. Stochastic gradient descent for hybrid quantum-classical optimization. *Quantum*, 4:314, 2020.

[223] Sergi Ramos-Calderer. Efficient quantum interpolation of natural data. *Phys. Rev. A*, 106:062427, 2022.

[224] Mudassir Moosa, Thomas W Watts, Yiyou Chen, Abhijat Sarma, and Peter L McMahon. Linear-depth quantum circuits for loading Fourier approximations of arbitrary functions. *Quantum Science and Technology*, 9(1):015002, 2023.

[225] Chelsea A. Williams, Annie E. Paine, Hsin-Yu Wu, Vincent E. Elfving, and Oleksandr Kyriienko. Quantum Chebyshev Transform: Mapping, Embedding, Learning and Sampling Distributions. *arXiv e-prints*, arXiv:2306.17026, 2023.

[226] Gabriel Marin-Sanchez, Javier Gonzalez-Conde, and Mikel Sanz. Quantum algorithms for approximate function loading. *Phys. Rev. Res.*, 5:033114, Aug 2023.

[227] Ewin Tang. A quantum-inspired classical algorithm for recommendation systems. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, page 217–228, New York, NY, USA, 2019. Association for Computing Machinery.

[228] Ewin Tang. Quantum principal component analysis only achieves an exponential speedup because of its state preparation assumptions. *Phys. Rev. Lett.*, 127:060503, 2021.

[229] András Gilyén, Zhao Song, and Ewin Tang. An improved quantum-inspired algorithm for linear regression. *Quantum*, 6:754, 2022.

[230] András Gilyén, Seth Lloyd, and Ewin Tang. Quantum-inspired low-rank stochastic regression with logarithmic dependence on the dimension. *arXiv e-prints*, arXiv:1811.04909, 2018.

[231] Nai-Hui Chia, András Pal Gilyén, Tongyang Li, Han-Hsuan Lin, Ewin Tang, and Chunhao Wang. Sampling-based sublinear low-rank matrix arithmetic framework for dequantizing quantum machine learning. *J. ACM*, 69(5), 2022.

[232] Fergus Barratt, James Dborin, and Lewis Wright. Improvements to gradient descent methods for quantum tensor network machine learning. *arXiv e-prints*, arXiv:2203.03366, 2022.

[233] Tianyi Hao, Xuxin Huang, Chunjing Jia, and Cheng Peng. A quantum-inspired tensor network algorithm for constrained combinatorial optimization problems. *Frontiers in Physics*, 10, 2022.

[234] Román Orús. Tensor networks for complex quantum systems. *Nature Reviews Physics*, 1(9):538–550, 2019.

[235] Edwin Stoudenmire and David J Schwab. Supervised learning with tensor networks. *Advances in Neural Information Processing Systems*, 29, 2016.

[236] E. Miles Stoudenmire. Learning relevant features of data with multi-scale tensor networks. *Quantum Science and Technology*, 3(3):034003, 2018.

[237] Zhao-Yu Han, Jun Wang, Heng Fan, Lei Wang, and Pan Zhang. Unsupervised generative modeling using matrix product states. *Phys. Rev. X*, 8:031012, 2018.

[238] Ding Liu, Shi-Ju Ran, Peter Wittek, Cheng Peng, Raul Blázquez García, Gang Su, and Maciej Lewenstein. Machine learning by unitary tensor network of hierarchical tree structure. *New Journal of Physics*, 21(7):073059, 2019.

[239] Jing Liu, Sujie Li, Jiang Zhang, and Pan Zhang. Tensor networks for unsupervised machine learning. *Phys. Rev. E*, 107:L012103, 2023.

[240] Giacomo Torlai, Christopher J. Wood, Atithi Acharya, Giuseppe Carleo, Juan Carrasquilla, and Leandro Aolita. Quantum process tomography with unsupervised learning and tensor networks. *Nature Communications*, 14(1):2858, 2023.

[241] Tom Vieijra, Laurens Vanderstraeten, and Frank Verstraete. Generative modeling with projected entangled-pair states. *arXiv preprint arXiv:2202.08177*, 2022.

[242] Samuel T. Wauthier, Bram Vanhecke, Tim Verbelen, and Bart Dhoedt. Learning generative models for active inference using tensor networks. In Christopher L. Buckley, Daniela Cialfi, Pablo Lanillos, Maxwell Ramstead, Noor Sajid, Hideaki Shimazaki, and Tim Verbelen, editors, *Active Inference*, pages 285–297, Cham, 2023. Springer Nature Switzerland.

[243] Friederike Metz and Marin Bukov. Self-correcting quantum many-body control using reinforcement learning with tensor networks. *Nature Machine Intelligence*, 5(7):780–791, 2023.

[244] Sofyan Iblisdir, Jose Ignacio Latorre, and Román Orús. Entropy and exact matrix-product representation of the laughlin wave function. *Phys. Rev. Lett.*, 98:060402, 2007.

[245] Lorenz Richter, Leon Sallandt, and Nikolas Nüsken. Solving high-dimensional parabolic PDEs using the tensor train format. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 8998–9009. PMLR, 2021.

[246] Sergey Dolgov, Boris N. Khoromskij, Alexander Litvinenko, and Hermann G. Matthies. Polynomial Chaos Expansion of random coefficients and the solution of stochastic partial differential equations in the Tensor Train Format. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):1109–1135, 2015.

[247] Martin Eigel, Max Pfeffer, and Reinhold Schneider. Adaptive stochastic Galerkin FEM with hierarchical tensor representations. *Numerische Mathematik*, 136(3):765–803, 2016.

[248] Rui Hong, Ya-Xuan Xiao, Jie Hu, An-Chun Ji, and Shi-Ju Ran. Functional tensor network solving many-body Schrödinger equation. *Phys. Rev. B*, 105:165116, 2022.

[249] Michael P. Zaletel, Roger S. K. Mong, Christoph Karrasch, Joel E. Moore, and Frank Pollmann. Time-evolving a matrix product state with long-ranged interactions. *Phys. Rev. B*, 91(16):165112, 2015.

[250] Ulrich Schollwöck. The density-matrix renormalization group. *Rev. Mod. Phys.*, 77:259–315, 2005.

[251] Jacob C. Bridgeman and Christopher T. Chubb. Hand-waving and interpretive dance: an introductory course on tensor networks. *Journal of Physics A: Mathematical and Theoretical*, 50(22):223001, 2017.

[252] Erwin Fehlberg. Classical fifth-, sixth-, seventh-, and eighth-order Runge-Kutta formulas with stepsize control. *NASA Technical Report*, 287, 1968.

[253] C Leforestier, R.H Bisseling, C Cerjan, M.D Feit, R Friesner, A Guldberg, A Hammerich, G Jolicard, W Karrlein, H.-D Meyer, N Lipkin, O Roncero, and R Kosloff. A comparison of different propagation schemes for the time dependent Schrödinger equation. *Journal of Computational Physics*, 94(1):59–80, 1991.

[254] Alex A. Gorodetsky and John D. Jakeman. Gradient-based optimization for regression in the functional tensor-train format. *Journal of Computational Physics*, 374:1219–1238, 2018.

[255] Jinhui Wang, Chase Roberts, Guifré Vidal, and Stefan Leichenauer. Anomaly detection with tensor networks. *arXiv e-prints*, arXiv:2006.02516, 2020.

[256] R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK Users' Guide*. Society for Industrial and Applied Mathematics, 1998.

[257] W. E. Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quarterly of Applied Mathematics*, 9(1):17–29, 1951.

[258] Sara Pollock and L.Ridgway Scott. Extrapolating the arnoldi algorithm to improve eigenvector convergence. *International Journal of Numerical Analysis and Modeling*, 18(5):712–721, 2021.

[259] Karen A. Hallberg. Density-matrix algorithm for the calculation of dynamical properties of low-dimensional systems. *Phys. Rev. B*, 52:R9827–R9830, 1995.

[260] Piet E. Dargel, Andreas Honecker, Robert Peters, Reinhard M. Noack, and T. Pruschke. Adaptive Lanczos-vector method for dynamic properties within the density matrix renormalization group. *Phys. Rev. B*, 83:161104, 2011.

[261] Piet E. Dargel, Anton Wöllert, Andreas Honecker, Ian P. McCulloch, Ulrich Schollwöck, and T. Pruschke. Lanczos algorithm with matrix product states for dynamical correlation functions. *Phys. Rev. B*, 85:205119, 2012.

[262] Andreas Holzner, Andreas Weichselbaum, Ian P. McCulloch, Ulrich Schollwöck, and Jan von Delft. Chebyshev matrix product state approach for spectral functions. *Phys. Rev. B*, 83:195115, 2011.

[263] Michael Chandross and J. C. Hicks. Density-matrix renormalization-group method for excited states. *Phys. Rev. B*, 59:9699–9702, 1999.

[264] Alberto Baiardi, Anna Klára Kelemen, and Markus Reiher. Excited-State DMRG Made Simple with FEAST. *Journal of Chemical Theory and Computation*, 18(1):415–430, 2022.

[265] Eric Jeckelmann. Dynamical density-matrix renormalization-group method. *Phys. Rev. B*, 66:045114, 2002.

[266] Ulrich Schollwöck. Methods for time dependence in DMRG. In *AIP Conference Proceedings*. AIP, 2006.

[267] The MathWorks Inc. Matlab version r2024a, 2024.

[268] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[269] Andreas Stathopoulos and James R. McCombs. PRIMME: PReconditioned Iterative MultiMethod Eigensolver: Methods and software description. *ACM Transactions on Mathematical Software*, 37(2):21:1–21:30, 2010.

[270] Lingfei Wu, Eloy Romero, and Andreas Stathopoulos. PRIMME_SVDS: A High-Performance Preconditioned SVD Solver for Accurate Large-Scale Computations. *SIAM Journal on Scientific Computing*, 39(5):S248–S271, 2017.

[271] David K. Hoffman and Donald J. Kouri. Distributed approximating function theory: A general, fully quantal approach to wave propagation. *The Journal of Physical Chemistry*, 96(3):1179–1184, 1992.

[272] F. Alexander Wolf, Ian P. McCulloch, Olivier Parcollet, and Ulrich Schollwöck. Chebyshev matrix product state impurity solver for dynamical mean-field theory. *Phys. Rev. B*, 90:115124, 2014.

[273] Thomas Koffel, M. Lewenstein, and Luca Tagliacozzo. Entanglement entropy for the long-range Ising chain in a transverse field. *Phys. Rev. Lett.*, 109:267203, 2012.

[274] Jutho Haegeman, Christian Lubich, Ivan Oseledets, Bart Vandereycken, and Frank Verstraete. Unifying time evolution and optimization with matrix product states. *Phys. Rev. B*, 94:165116, 2016.

[275] Valentin Zauner-Stauber, Laurens Vanderstraeten, Matthew T. Fishman, Frank Verstraete, and Jutho Haegeman. Variational optimization algorithms for uniform matrix product states. *Phys. Rev. B*, 97:045145, 2018.

[276] Sebastián Roca-Jerat, Juan Román-Roche, and David Zueco. Qudit machine learning. *Machine Learning: Science and Technology*, 5(1):015057, 2024.

[277] Levitodynamics ETH, 2024. Accessed: 2024-08-29.

[278] Eric Bonvin, Louisiane Devaud, Massimiliano Rossi, Andrei Militaru, Lorenzo Dania, Dmitry S. Bykov, Oriol Romero-Isart, Tracy E. Northup, Lukas Novotny, and Martin Frimmer. State expansion of a levitated nanoparticle in a dark harmonic potential. *Physical Review Letters*, 132(25):253602, 2024.

[279] Mark Roda-Llordes, Davide Candoli, Piotr T. Grochowski, Andreu Riera-Campeny, Thomas Agrenius, Juan José García-Ripoll, Carlos Gonzalez-Ballestero, and Oriol Romero-Isart. Numerical simulation of large-scale nonlinear open quantum mechanics. *Phys. Rev. Res.*, 6:013262, 2024.

[280] Marc Roda-Llordes, Andreu Riera-Campeny, Davide Candoli, Piotr T. Grochowski, and Oriol Romero-Isart. Macroscopic quantum superpositions via dynamics in a wide double-well potential. *Phys. Rev. Lett.*, 132:023601, 2024.

[281] Levitodynamics ICFO, 2024. Accessed: 2024-08-29.

[282] Chao-Yang Lu, Yuan Cao, Cheng-Zhi Peng, and Jian-Wei Pan. Micius quantum experiments in space. *Rev. Mod. Phys.*, 94:035001, 2022.

[283] Simon Storz, Josua Schär, Anatoly Kulikov, Paul Magnard, Philipp Kurpiers, Janis Lütolf, Theo Walter, Adrian Copetudo, Kevin Reuer, Abdulkadir Akin, Jean-Claude Besse, Mihai Gabureac, Graham J. Norris, Andrés Rosario, Ferran Martin, José Martinez, Waldimar Amaya, Morgan W. Mitchell, Carlos Abellan, Jean-Daniel Bancal, Nicolas Sangouard, Baptiste Royer, Alexandre Blais, and Andreas Wallraff. Loophole-free Bell inequality violation with superconducting circuits. *Nature*, 617(7960):265–270, 2023.

[284] Nancy Aggarwal, George P. Winstone, Mae Teo, Masha Baryakhtar, Shane L. Larson, Vicky Kalogera, and Andrew A. Geraci. Searching for new physics with a levitated-sensor-based gravitational-wave detector. *Phys. Rev. Lett.*, 128:111101, 2022.

[285] David C. Moore, Alexander D. Rider, and Giorgio Gratta. Search for millicharged particles using optically levitated microspheres. *Phys. Rev. Lett.*, 113:251801, 2014.

[286] Alexander D. Rider, David C. Moore, Charles P. Blakemore, Maxime Louis, Marie Lu, and Giorgio Gratta. Search for screened interactions associated with dark energy below the 100 μm length scale. *Phys. Rev. Lett.*, 117:101101, 2016.

[287] James Bateman, Ian McHardy, Alexander Merle, Tim R. Morris, and Hendrik Ulbricht. On the existence of low-mass dark matter and its direct detection. *Scientific Reports*, 5(1):8058, 2015.

[288] Fernando Monteiro, Gadi Afek, Daniel Carney, Gordan Krnjaic, Jiaxiang Wang, and David C. Moore. Search for composite dark matter with optically levitated sensors. *Phys. Rev. Lett.*, 125:181102, 2020.

[289] D Carney, G Krnjaic, D C Moore, C A Regal, G Afek, S Bhave, B Brubaker, T Corbitt, J Cripe, N Crisosto, A Geraci, S Ghosh, J G E Harris, A Hook, E W Kolb, J Kunjummen, R F Lang, T Li, T Lin, Z Liu, J Lykken, L Magrini, J Manley, N Matsumoto, A Monte, F Monteiro, T Purdy, C J Riedel, R Singh, S Singh, K Sinha, J M Taylor, J Qin, D J Wilson, and Y Zhao. Mechanical quantum sensing in the search for dark matter. *Quantum Science and Technology*, 6(2):024002, 2021.

[290] Chiara Marletto and Vlatko Vedral. Gravitationally induced entanglement between two massive particles is sufficient evidence of quantum effects in gravity. *Phys. Rev. Lett.*, 119:240402, 2017.

[291] Sougato Bose, Anupam Mazumdar, Gavin W. Morley, Hendrik Ulbricht, Marko Toroš, Mauro Paternostro, Andrew A. Geraci, Peter F. Barker, M. S. Kim, and Gerard Milburn. Spin entanglement witness for quantum gravity. *Phys. Rev. Lett.*, 119:240401, 2017.

[292] Andrew Geraci and Hart Goldman. Sensing short range forces with a nanosphere matter-wave interferometer. *Phys. Rev. D*, 92:062002, 2015.

[293] Benjamin A Stickler, Birthe Papendell, Stefan Kuhn, Björn Schrinski, James Millen, Markus Arndt, and Klaus Hornberger. Probing macroscopic quantum superpositions with nanorotors. *New Journal of Physics*, 20(12):122001, 2018.

[294] C. M. DeWitt and D. Rickles. *The role of gravitation in physics: report from the 1957 Chapel Hill Conference*, volume 5. Edition Open Access, 2011.

[295] H. Pino, J. Prat-Camps, K. Sinha, B. Prasanna Venkatesh, and O. Romero-Isart. On-chip quantum interference of a superconducting microsphere. *Quantum Science and Technology*, 3(2):025001, 2018.

[296] Oriol Romero-Isart. Coherent inflation for large quantum superpositions of levitated microspheres. *New Journal of Physics*, 19(12):123029, 2017.

[297] Tim Weiss, Marc Roda-Llordes, Erik Torrontegui, Markus Aspelmeyer, and Oriol Romero-Isart. Large quantum delocalization of a levitated nanoparticle using optimal control: Applications for force sensing and entangling via weak forces. *Phys. Rev. Lett.*, 127:023601, 2021.

[298] Oriol Romero-Isart, Anika C. Pflanzer, Florian Blaser, Rainer Kaltenbaek, Nikolai Kiesel, Markus Aspelmeyer, and J. Ignacio Cirac. Large quantum superpositions and interference of massive nanometer-sized objects. *Phys. Rev. Lett.*, 107:020405, 2011.

[299] Oriol Romero-Isart. Quantum superposition of massive objects and collapse models. *Phys. Rev. A*, 84:052121, 2011.

[300] Aditi Mitra. Quantum quench dynamics. *Annual Review of Condensed Matter Physics*, 9(1):245–259, 2018.

[301] Sumit R. Das. Quantum quench and universal scaling, 2020.

[302] Ashley Setter, Jamie Vovrosh, and Hendrik Ulbricht. Characterization of non-linearities through mechanical squeezing in levitated optomechanics. *Applied Physics Letters*, 115(15):153106, 2019.

[303] Lukas Neumeier, Mario A. Ciampini, Oriol Romero-Isart, Markus Aspelmeyer, and Nikolai Kiesel. Fast quantum interference of a nanoparticle via optical potential control. *Proceedings of the National Academy of Sciences*, 121(4):e2306953121, 2024.

[304] David K. Hoffman, Mark Arnold, and Donald J. Kouri. Properties of the optimum distributed approximating function class propagator for discretized and continuous wave packet propagations. *The Journal of Physical Chemistry*, 96(16):6539–6545, 1992.

[305] D. K. Hoffman, T. L. Marchioro, Mark Arnold, Youhong Huang, Wei Zhu, and Donald J. Kouri. Variational derivation and extensions of distributed approximating functionals. *Journal of Mathematical Chemistry*, 20(1):117–140, 1996.

[306] David K. Hoffman and Donald J. Kouri. Distributed approximating functionals: a robust, new approach to computational chemistry and physics. In Antonio Laganà and Antonio Riganelli, editors, *Reaction and Molecular Dynamics*, pages 57–73, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[307] Samuel M. Greene and Victor S. Batista. Tensor-Train Split-Operator Fourier Transform (TT-SOFT) Method: Multidimensional Nonadiabatic Quantum Dynamics. *Journal of Chemical Theory and Computation*, 13(9):4034–4042, 2017.

[308] Ningyi Lyu, Micheline B. Soley, and Victor S. Batista. Tensor-Train Split-Operator KSL (TT-SOKSL) Method for Quantum Dynamics Simulations. *Journal of Chemical Theory and Computation*, 18(6):3327–3346, 2022.

[309] Micheline B. Soley, Paul Bergold, Alex A. Gorodetsky, and Victor S. Batista. Functional Tensor-Train Chebyshev Method for Multidimensional Quantum Dynamics Simulations. *Journal of Chemical Theory and Computation*, 18(1):25–36, 2022.

[310] Sergei V. Dolgov, Boris N. Khoromskij, and Ivan V. Oseledets. Fast Solution of Parabolic Problems in the Tensor Train/Quantized Tensor Train Format with Initial Application to the Fokker–Planck Equation. *SIAM Journal on Scientific Computing*, 34(6):A3016–A3038, 2012.

[311] Erika Ye and Nuno F. G. Loureiro. Quantum-inspired method for solving the Vlasov-Poisson equations. *Phys. Rev. E*, 106:035208, 2022.

[312] Masuo Suzuki. Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations. *Physics Letters A*, 146(6):319–323, 1990.

[313] Masuo Suzuki. General theory of fractal path integrals with applications to many-body theories and statistical physics. *Journal of Mathematical Physics*, 32(2):400–407, 1991.

[314] R. I. McLachlan and P. Atela. The accuracy of symplectic integrators. *Nonlinearity*, 5(2):541–562, 1992.

[315] Pavel Holoborodko. Smooth noise robust differentiators. http://www.holoborodko.com/pavel/numerical-methods/numerical-derivative/smooth-low-noise-differentiators/, 2008.

[316] B. Cano and J. M. Sanz-Serna. Error Growth in the Numerical Integration of Periodic Orbits, with Application to Hamiltonian and Reversible Systems. *SIAM Journal on Numerical Analysis*, 34(4):1391–1417, 1997.

[317] Talitha Weiss and Oriol Romero-Isart. Quantum motional state tomography with nonquadratic potentials and neural networks. *Phys. Rev. Res.*, 1:033157, 2019.

[318] Julien Zylberman and Fabrice Debbasch. Efficient quantum state preparation with Walsh series. *Phys. Rev. A*, 109:042401, 2024.

[319] Boris N. Khoromskij and Ivan Oseledets. Quantics-TT Collocation Approximation of Parameter-Dependent and Stochastic Elliptic PDEs. *Computational Methods in Applied Mathematics*, 10(4):376–394, 2010.

[320] Boris N. Khoromskij and Ivan V. Oseledets. QTT approximation of elliptic solution operators in higher dimensions. *Russian Journal of Numerical Analysis and Mathematical Modelling*, 26(3):303–322, 2011.

[321] Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654, 1973.

[322] Raj G. Patel, Chia-Wei Hsing, Serkan Sahin, Samuel Palmer, Saeed S. Jahromi, Shivam Sharma, Tomas Dominguez, Kris Tziritas, Christophe Michel, Vincent Porte, Mustafa Abid, Stephane Aubert, Pierre Castellani, Samuel Mugel, and Roman Orus. Quantum-inspired tensor neural networks for option pricing. *arXiv e-prints*, arXiv:2212.14076, 2024.

[323] Raj G. Patel, Tomas Dominguez, Mohammad Dib, Samuel Palmer, Andrea Cadarso, Fernando De Lope Contreras, Abdelkader Ratnani, Francisco Gomez Casanova, Senaida Hernández-Santana, Álvaro Díaz-Fernández, Eva Andrés, Jorge Luis-Hita, Escolástico

Sánchez-Martínez, Samuel Mugel, and Roman Orus. Application of Tensor Neural Networks to Pricing Bermudan Swaptions. *arXiv e-prints*, arXiv:2304.09750, 2024.

[324] Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654, 1973.

[325] S. Moon-Ho Song, S. Napel, N.J. Pelc, and G.H. Glover. Phase unwrapping of mr phase images using Poisson equation. *IEEE Transactions on Image Processing*, 4(5):667–676, 1995.

Acronyms

ODE: ordinary differential equation

PDE: partial differential equation

FFT: Fast Fourier transform

QFT: Quantum Fourier transform

VQA: variational quantum algorithm

VQE: variational quantum eigensolver

MPS: matrix product state

MPO: matrix product operator

PEPS: projected entangled pair states

TTN: tree tensor network

MERA: multi-scale entanglement renormalization ansatz

DMRG: density matrix renormalization group

TEBD: time-evolving block decimation

HDAF: Hermite distributed approximating functional