UNIVERSITÀ
DEGLI STUDI
DI TRIESTE

UNIVERSITÀ DEGLI STUDI DI TRIESTE

XXXVIII CICLO DEL DOTTORATO DI RICERCA IN

Uso del Quantum Computing per simulazioni numeriche in Cosmologia

Finanziato dall'Unione europea - NextGenerationEU
Funded by the European Union - NextGenerationEU

Fondazione ICSC National Recovery and Resilience Plan (PNRR); Project ID CN-00000013 "Italian Research Center on High-Performance Computing, Big Data and Quantum Computing"

QUANTUM ALGORITHMS FOR COSMOLOGICAL SIMULATIONS

Settore scientifico-disciplinare: FIS/05

DOTTORANDO / A
LUCA CAPPELLI

COORDINATORE
PROF. ANGELO BASSI

SUPERVISORE DI TESI
DR. GIUSEPPE MURANTE

CO-SUPERVISORE DI TESI
PROF. STEFANO BORGANI

ANNO ACCADEMICO 2024/2025

I dedicate this thesis to the memory of Cappelli Giovanni (12/07/1931 – 08/10/2023),
– *grateful for the opportunities that you always valued so deeply.* –

Acknowledgements

I would like to thank my supervisors, Prof.Stefano Borgani and Dr. Giuseppe Murante, for their guidance and patience. This work has been supported by the Fondazione ICSC National Recovery and Resilience Plan (PNRR) Project ID CN-00000013 "Italian Research Center on High-Performance Computing, Big Data and Quantum Computing" funded by MUR Missione 4 Componente 2 Investimento 1.4: "Potenziamento strutture di ricerca e creazione di "campioni nazionali di R&S (M4C2-19)" - Next GenerationEU (NGEU).

Contents

Dedication i

Acknowledgements ii

Introduction 1

1 Background of Cosmic Structure Formation 7
1.1 Cosmic Density Perturbations . . . 7
1.1.1 Dark Matter and Large Scale Structures . . . 9
1.1.2 The Vlasov-Poisson equation . . . 11
1.1.3 Jeans Equations . . . 12
1.2 Numerical methods . . . 14
1.2.1 Eulerian Vlasov-Poisson solvers . . . 14
1.2.2 N-body methods . . . 15
1.2.3 The Schrödinger method . . . 18
1.3 Summary . . . 21

2 Introduction to Quantum Computing 22
2.1 Classical Computation and Its Limitations . . . 23
2.1.1 Turing Machines . . . 23
2.1.2 Boolean Circuits and Binary Information . . . 23
2.1.3 Physical Limits, and the Need for New Computational Paradigms . 24
2.2 Quantum Computation: Information and States . . . 25
2.2.1 Qubits: The Basic Unit of Quantum Information . . . 25
2.2.2 Quantum States, Superposition, and Entanglement . . . 26
2.3 Quantum Operations and the Circuit Formalism . . . 29
2.3.1 Single-Qubit Gates . . . 31
2.3.2 Two-Qubit and Multi-Qubit Gates . . . 32
2.3.3 Circuit Cost, Gate Decomposition, and Ancilla Qubits . . . 34
2.4 Noise and Decoherence . . . 36
2.4.1 NISQ and Fault-Tolerant Quantum Computing . . . 37
2.5 Measurements in Quantum Computing . . . 38
2.6 Summary . . . 39

3 Comparison of Vlasov Solvers 41
3.1 Numerical solvers . . . 41
3.1.1 Eulerian Vlasov-Poisson Solver . . . 42

3.1.2 Spectral method for Schrödinger-Poisson . . . 47
3.1.3 Particle mesh . . . 48
3.1.4 Poisson Solver . . . 50
3.2 Numerical Test of the Solvers . . . 50
3.2.1 Details on numerical implementation . . . 50
3.2.2 Timestep . . . 51
3.2.3 Maxwellian initial condition . . . 51
3.2.4 CDM Initial condition . . . 51
3.2.5 Smoothed initial condition . . . 55
3.2.6 The right smoothing choice . . . 57
3.2.7 Summing up . . . 59
3.3 Numerical Comparison . . . 59
3.3.1 One-dimensional sinusoidal perturbation . . . 59
3.3.2 Two dimensional collapse . . . 67
3.4 Summary . . . 68

4 A Quantum Algorithm for Schrödinger-Poisson 71
4.1 The nonlinear SP equation on quantum computers . . . 72
4.1.1 The quantum computing approach to the SP equation . . . 73
4.2 The Algorithm . . . 77
4.2.1 Circuit implementation . . . 77
4.3 Results and Discussion . . . 80
4.3.1 Numerical simulations . . . 81
4.3.2 Interpretation of the SP results . . . 83
4.3.3 Scaling of required resources . . . 84
4.4 Summary . . . 89

5 Towards a Schrödinger-Poisson algorithm for large scale simulations 91
5.1 Multidimensional approach . . . 91
5.2 Improvements . . . 92
5.2.1 Reference Timestep . . . 93
5.2.2 Ansatz . . . 93
5.3 Scaling Challenges . . . 94
5.3.1 Regularization problem . . . 94
5.3.2 Measurement precision . . . 102
5.3.3 Timestep . . . 103
5.4 Linearization Approach . . . 103
5.4.1 Carleman Embedding . . . 103
5.4.2 Classical solutions . . . 104
5.4.3 Second order Carleman Embedding . . . 106
5.4.4 Third order Carleman Embedding . . . 108
5.5 Numerical Simulations . . . 108
5.6 Conclusions . . . 108

6 QFRANS: Quantum Fixed RAdius Neighbor Search 111
6.1 Introduction . . . 111
6.2 The fixed-radius neighbor search . . . 114
6.3 Quantum search algorithms . . . 116
6.3.1 FPS algorithm . . . 117
6.3.2 OAA algorithm . . . 121
6.4 Quantum FRANS algorithm . . . 121
6.4.1 State preparation . . . 123
6.4.2 The oracle $\hat{O}$ . . . 126
6.4.3 The reflection . . . 129
6.4.4 The readout measurement . . . 129
6.5 Numerical Tests . . . 131
6.6 Analysis of noise resilience . . . 132
6.7 Discussion . . . 135

Conclusions 138

A Supplement material for the Schrödinger-Poisson quantum algorithm 142
A.1 Dimension Re-Scaling of Schrödinger-Poisson equation . . . 142
A.2 Potential cost function . . . 143
A.3 Adder circuit . . . 144
A.4 Circuit proofs . . . 144

B Supplement material to the Carleman linearization 148
B.1 Second order Carleman-SP system . . . 148
B.2 Third order Carleman-SP system . . . 149

References 151

Introduction

Numerical simulations have established themselves as the third pillar of modern cosmology, bridging the gap between theoretical models and increasingly precise astronomical observations. Today, our understanding of the cosmos relies on the systematic comparison between these models and observational data. Numerical simulations play a central role in this process by tracking the evolution of initial density fluctuations generated during the earliest phases of cosmic evolution, within a specified cosmological model [1], [2]. Initially amplified during cosmic inflation and subsequently shaped by gravitational interactions over billions of years, these fluctuations give rise to the complex large-scale structures observed today [3]. The filamentary cosmic web connecting galaxy clusters, the dark matter haloes hosting individual galaxies, and the fine substructure revealed through gravitational lensing all provide critical tests of cosmological models when confronted with simulation results [4]. Figure 1 shows a representative slice of the Millennium Simulation [5], included here as an illustrative example.

Such comparisons are essential in regimes where gravitational clustering has progressed far beyond the linear approximation [2]. In this non-linear domain, numerical simulations become the only viable means of predicting the detailed evolution of cosmic structures, as no analytic method can reliably capture the detailed morphology of dark matter haloes or the emergence of satellite substructure [1]. As a result, numerical simulations are not merely complementary to theoretical analysis, but constitute a fundamental tool for validating cosmological models and interpreting observations across a wide range of scales [6].

Over the years, different cosmological models have been proposed to explain the origin and evolution of the Universe. However, a number of astrophysical and cosmological observations consistently support the so-called standard cosmological model [7]. These include measurements of the cosmic microwave background (CMB), observations of type Ia supernovae [8], [9], large-scale galaxy surveys [10], and gravitational lensing [11]. In this model, the mass-energy content of the Universe is composed of about 70% of an unknown form of Dark Energy (DE), which accounts for the accelerated cosmic expansion, by about 25% of an unknown form of collisionless non-baryonic Dark Matter (DM), while only the remaining $\sim 5\%$ is made of ordinary matter, called baryonic by astrophysicists. The backbone of cosmological simulations is the dark matter distribution. Dark matter is treated as a collisionless fluid governed by the Vlasov–Poisson equations [12] and is typically represented by particles that move under their mutual gravitational attraction. A small initial density perturbation is then evolved forward in time until the desired epoch, measured in terms of redshift. The resulting distribution maps the gravitational potential wells that form the scaffolding upon which the visible structure is assembled.

Baryonic matter accounts for most of the observable electromagnetic emission asso-

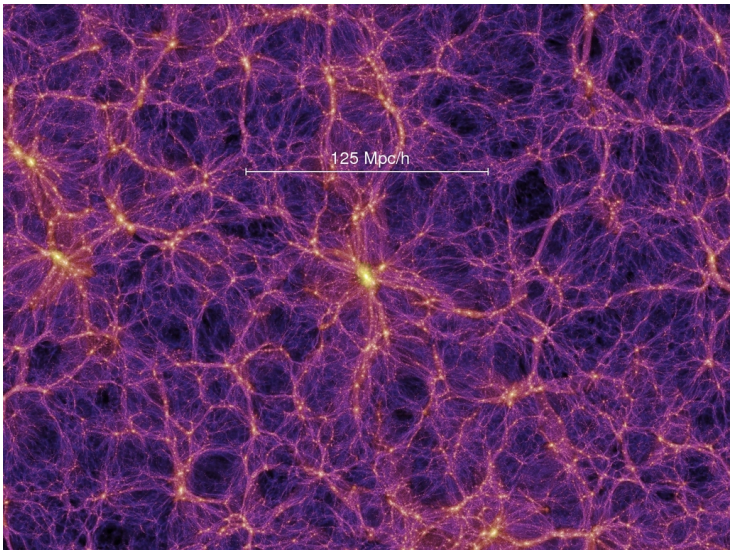


Figure 1: Slice of the Millennium Simulation [5] showing the simulated cosmic web of DM distribution. The bright yellow regions are the high density clusters that form at the intersection of filaments. https://wwwmpa.mpa-garching.mpg.de/galform/virgo/millennium/

ciated with galactic systems, galaxy clusters, and the intergalactic medium [12]. Unlike dark matter, baryons interact electromagnetically and, therefore, undergo a rich variety of physical processes. They radiate energy, are heated by shocks during gravitational collapse, cool through atomic and molecular transitions, fragment into dense structures, and ultimately fuel star formation. Stellar evolution, supernova explosions, and accretion onto active galactic nuclei further inject energy and momentum into the surrounding medium [13], [14].

Modern hydrodynamical simulations attempt to model this complexity by solving the equations of fluid dynamics alongside gravity [15]. However, many relevant physical mechanisms operate below the resolution limit of even the most advanced simulations. As a consequence, additional sub-resolution models are required to describe processes such as star formation, chemical enrichment, and feedback from supernovae and active galactic nuclei, introducing further computational and modeling challenges [6].

Modern cosmology places increasingly stringent demands on numerical simulations. High resolution, both in space and time, is needed to resolve the detailed internal structure of galaxies and clusters, capturing both the dark matter framework and the dynamics of baryonic matter, while simulations must also cover large volumes to match the scale of contemporary surveys, which now map billions of galaxies across the sky [6]. To explore the variety of possible cosmic histories and constrain theoretical uncertainties, researchers run ensembles of simulations in which the underlying cosmological and astrophysical assumptions are systematically varied [1]. These ensembles quantify cosmic variance and probe parameter degeneracies, situations in which different combinations of inputs can produce similar large-scale structures, allowing researchers to identify which sets of assumptions are most consistent with observations.

Building on this statistical approach, cosmologists have recently begun creating *digital twins* of specific observed regions of the Universe [16]. The term, borrowed from

meteorology, refers to detailed virtual replicas that aim to reproduce particular cosmic structures rather than exploring parameter space broadly. In cosmology, digital twins start from observationally constrained initial conditions and systematically vary cosmological parameters to generate multiple evolutionary scenarios for the same region [17]. By comparing this catalog of virtual realizations to real observations, researchers can determine which parameter combinations best reproduce the observed structures. In this way, digital twins establish a direct bridge between theoretical models and the actual local Universe, providing a powerful tool for testing cosmological theories and interpreting complex observational data.

Despite advances in classical and hybrid simulation techniques, the combined requirements of high resolution, large volumes, and multiple realizations continue to push conventional methods to their limits. In response, researchers are exploring more radical solutions, including specialized hardware accelerators and quantum computing frameworks.

Quantum computing remains largely in its developmental stage, yet it offers fundamentally new ways to represent and process information. In a quantum computer, information is stored in quantum states whose basic unit is the qubit, which can exploit superposition and entanglement to represent complex, highly correlated datasets [18]. This unique computational paradigm may, in principle, reduce the resources and execution time required for high-resolution cosmological simulations, and facilitate the efficient generation of multiple realizations of the same cosmic region—crucial for exploring parameter space, quantifying uncertainties, and constructing digital twins [19].

Current devices operate in the noisy intermediate-scale quantum (NISQ) era [20], with a moderate number of qubits, currently on the order of hundreds, and substantial error rates. Nevertheless, these limitations should not discourage exploration: the development of algorithms that leverage the scaling advantages of qubits, superposition, and entanglement remains a key priority. Once fully fault-tolerant quantum computers are realized, such algorithms could provide transformative improvements in simulation speed and computational capacity.

Quantum computing has already demonstrated its potential in other fields, such as quantum chemistry [21] where it has been used to simulate molecular systems that are intractable for classical methods. These successes suggest that similar advantages could be realized in cosmological applications, motivating the systematic exploration of quantum algorithms for high-resolution simulations.

Building on these considerations, the aim of this Thesis is to explore the applicability of quantum computing to cosmological simulations. Specifically, we investigate whether quantum algorithms can offer computational speedups in the evolution of dark matter perturbations, either through direct numerical integration or by acting as an accelerator in the most computationally demanding parts of classical simulation codes.

This Thesis is organized as follows: Chapter 1 establishes the physical and computational framework for the formation of cosmological structures. It begins with the evolution of density perturbations in the early Universe, examining their origin and their role in shaping the large-scale distribution of matter. A concise overview of the main classes of dark matter follows, with particular emphasis on the observational and theoretical motivations for adopting cold dark matter (CDM) as the basis for this work.

This chapter then introduces the Vlasov-Poisson (VP) equations, which govern the evolution of CDM perturbations. The distinction between the linear regime of small perturbations and the fully nonlinear regime is presented, highlighting the necessity of numerical methods where analytical solutions are insufficient. A succinct survey of the principal numerical approaches is provided, including direct integration of the VP system, N-body methods, and the Schrödinger method. The emphasis is placed on the latter, as it forms the foundation for the development of quantum algorithms intended to accelerate simulations of collisionless dark matter.

Together, these elements define a coherent framework for understanding structure formation, the rationale for focusing on CDM, and the computational strategies employed to model its nonlinear evolution. This foundation supports the analysis and comparisons presented in subsequent chapters, including the exploration of quantum computing approaches to cosmological simulations.

The second Chapter 2 provides an overview of the basic concepts of quantum computing that are relevant for the purposes of this Thesis. We begin by recalling the classical paradigm of computation, using the Turing machine as a reference, and then outline how the principles of quantum mechanics modify the fundamental components of an algorithm. In particular, we discuss how the representation of information, the implementation of logical operations, and the readout of results differ from their classical counterparts when computation is carried out using quantum states.

The discussion resumes by introducing the circuit model of quantum computation, which constitutes the standard framework for the formulation of quantum algorithms. Hereafter, this model will be employed throughout. Within this setting, we introduce the essential elements required in the remainder of this work, including qubits, unitary evolution, quantum gates, measurement processes, and the impact of noise in realistic devices. Explicit examples of qubit implementations on existing quantum hardware are also presented, providing a concrete link between the abstract formalism and current technological realizations. The chapter is intentionally focused on these core concepts, which form the conceptual basis for the developments presented in the subsequent chapters.

Chapter 3 examines the extent to which the Schrödinger method can faithfully reproduce the dynamics of the Vlasov–Poisson (VP) system. This analysis is pivotal to our broader objective: exploiting the inherently quantum nature of the Schrödinger-Poisson (SP) formulation to develop algorithms capable of efficiently evolving collisionless dark matter. By establishing the regimes in which SP accurately captures VP dynamics, we lay the groundwork for quantum-inspired or quantum-native simulations of cosmological structure formation.

To this end, we develop a set of numerical solvers, including direct integrations of the VP system, a particle-mesh N-body method, and a spectral solver for SP, tailored to efficiently handle the challenges posed by cold dark matter initial conditions. In particular, we introduce a novel regularization strategy that allows the direct VP solvers to operate reliably even in the presence of stiff initial velocity fields, overcoming a major limitation of previous methods. These results have been published in [22]. Through comparative tests, we assess whether SP successfully reproduces the key features of VP evolution in the case of a simple one-dimensional sinusoidal perturbation and a two-dimensional gaussian collapse. We also verify if a direct VP solver provides additional information in these scenarios, compared to a N-body approach. The results of these comparisons are part of

a scientific article currently in production.

Chapter 4 introduces a first quantum algorithm for the solution of the SP equation, based on a variational approach. The SP equation governs the dynamics of self-gravitating cold dark matter, introducing nonlinearities through the gravitational potential. Mapping this nonlinear, self-consistent problem onto a quantum device requires specialized algorithmic strategies. A classical-hybrid variational framework, inspired by Lubasch et al. [23], is adapted here to the nonlinear context of cosmological simulations.

The evolution of the wavefunction is implemented using a variational time evolution (VTE) method, tailored for grid-based problems. This approach allows for an exponential reduction in memory requirements, encoding N spatial grid points into $\log_2(N)$ qubits. Based on Ollitrault et al. [24], the VTE algorithm is further extended to cases where the potential is represented by a variational ansatz. Explicit quantum circuits are proposed to evaluate the necessary matrix elements, with a circuit depth scaling polynomially in the number of qubits and the number of measurements required for a given accuracy scaling polynomially with the number of grid points N.

The chapter examines the behavior of spatial resolution as the SP dynamics approach the classical limit ($\hbar/m \to 0$), revealing an empirical logarithmic relationship between the number of qubits and the SP scale parameter $\hbar/m$.This analysis offers insight into the resources required for SP simulations to reproduce classical behavior with acceptable accuracy. These results have been published in [25].

This chapter establishes a variational quantum algorithm for the Schrödinger–Poisson equation, providing a concrete framework for quantum simulations and representing a first step toward developing quantum algorithms for the formation of cosmological structures.

Chapter 5 is devoted to the development of a quantum algorithm for the solution of SP in the case of large systems. Building on the encouraging results obtained in the previous chapter, we first focus on improving the VTE approach for SP, aiming to overcome the primary limitations of the existing algorithm related to temporal evolution. In this context, we identify the key challenges in scaling the algorithm to larger systems, highlighting both computational bottlenecks and sources of numerical error.

In addition to refining the VTE method, we explore an alternative strategy that takes advantage of the inherent linearity of quantum mechanics to solve the SP problem after linearization. This approach provides a complementary perspective and serves as a benchmark for assessing the advantages and limitations of variational schemes. For both strategies, our analysis is conducted with a critical perspective, emphasizing not only their potential but also the intrinsic limitations and practical considerations that arise when attempting to extend these methods to large-scale quantum simulations.

These investigations lay the groundwork for large-scale SP simulations, highlighting the challenges that arise in developing a scalable quantum algorithm for SP.

Chapter 6 explores an alternative avenue for applying quantum computing to cosmological simulations of dark matter. Rather than directly simulating with a quantum approach the gravitational instability of dark matter density perturbations, we focus on leveraging quantum computing to accelerate the most computationally demanding portions of classical codes. In this approach, the quantum device functions as a high-performance accelerator, complementing classical computation.

We center our attention on the problem of neighbor search, a recurring and computationally intensive task in cosmological simulations. The computational cost is not

associated with the complexity of individual operations, but with the sheer volume of data that must be processed.

The specific task we address is the Fixed Radius Neighbor Search (FRANS), where all neighbors within a given fixed radius must be identified. Although this does not exactly replicate the full cosmological scenario, it captures a representative and practically relevant computational challenge. Our approach employs amplitude amplification techniques to accelerate this search.

We present a detailed design of the quantum circuit for the neighbor search, analyzing the implementation and computational cost of each component. Moreover, we proposed a strategy to determine the optimal number of quantum measurements. We validate this approach through numerical simulations using a toy model. Finally, we propose a strategy to mitigate noise in the readout measurement process. The results are currently under revision for publications.

This work establishes a conceptual foundation for integrating quantum computation into selected parts of classical cosmological simulations, highlighting practical issues and presenting the problem from a perspective that remains largely unexplored.

We conclude the Thesis with Chapter 6.7, where we reflect on the potential role of quantum computing in cosmological simulations. In particular, we analyze the limitations and possible developments for two alternative approaches: quantum computing as a direct solver for the evolution of cosmological perturbations and quantum computing as an accelerator for the computationally demanding components of classical simulation codes. We examine the strengths and weaknesses of both approaches and provide what we consider promising future developments.

Chapter 1

Background of Cosmic Structure Formation

This chapter introduces the landscape of cosmological simulations of the Universe in the ΛCDM model. We propose a quick review on the evolution of Dark Matter (DM) density perturbations and their role in galaxy formation, describing the equations that govern the process and the numerical methods used to solve it.

This chapter is organized into two main sections. In Sec. 1.1, we introduce the context of the ΛCDM model, explain the nature and evolution of cosmological perturbations, and discuss how the standard cosmological model is tested observationally. We also present the fundamental equation governing the evolution of these perturbations in the Universe: the Vlasov–Poisson (VP) equation.

In Sec. 1.2, we review the numerical methods commonly used to solve the VP equation, ranging from direct discretization to N-body techniques, and finally the Schrödinger method. This latter approach will provide the foundation for developing our first quantum algorithm.

1.1 Cosmic Density Perturbations

During the years, different cosmological models have been proposed to explain the origin and evolution of the Universe. However, a number of astrophysical and cosmological observations consistently point toward the definition of the so-called standard cosmological model [7]. In this model, the mass-energy content of the Universe is made of about 70% of an unknown form of Dark Energy (DE), which accounts for the accelerated cosmic expansion, by about 25% of an unknown form of collisionless non-baryonic Dark Matter (DM), while only the remaining $\sim 5\%$ is made of baryonic matter. In astrophysics, the term *baryonic* refers to all ordinary matter that participates in the formation of stars, planets, gas clouds, and other cosmic structures [26]. The inclusion of the electrons is due to the fact that their contribution in mass is negligible respect to the total. It is used primarily to distinguish DM from this visible, interacting matter. Viable models of galaxy formation require dark matter to be cold (CDM), i.e., non-relativistic with negligible streaming velocities compared to the speed of light. With further observational evidence for DE to be consistent with a cosmological constant term (Λ) in the Einstein field equations, all this leads to the definition of the standard ΛCDM cosmological model

[27]. While the exact nature of cosmic dark constituents remains so far elusive, it is widely accepted that the gravitational instability of the tiny CDM density perturbations imprinted in the primordial Universe drive the formation of cosmic structures, from the kiloparsec (kpc) scales of the global cosmic web, to Gigaparsec (Gpc) scales relevant for galaxy formation [12].

The prevailing hypoThesis for the origin of cosmological perturbations, which has found strong support in observational data, posits quantum fluctuations of the inflation field as the primordial source. Inflation is a hypothesized epoch in the very early Universe during which the expansion rate was extraordinarily large and nearly exponential, resolving several conceptual challenges of the standard Big Bang model, such as the observed homogeneity of causally disconnected regions. During this phase, small energy variations arise spontaneously in the vacuum field according to the Heisenberg uncertainty principle and are subsequently amplified by the rapid expansion, seeding the primordial density perturbations that drive the formation of cosmic structures. As space expands exponentially, each microscopic fluctuation is stretched to cosmic scales and crosses beyond the causal horizon, where it becomes inaccessible to local physical processes. At this juncture, the amplitude of each mode surpasses the maximum distance over which physical information can propagate, causing its evolution to effectively freeze. Since the inflationary background evolves slowly, perturbations generated at different epochs acquire nearly identical amplitudes, yielding an almost scale-invariant power spectrum. After the end of inflation, the field decays during reheating, converting its energy into standard model particles. These frozen perturbations then manifest in the matter and radiation fields. Subsequently, as the cosmological horizon expands and these modes re-enter the causal domain, gravitational instability amplifies overdensities while suppressing underdensities. The resulting perturbation field can be considered Gaussian in nature due to the statistically independent quantum modes producing uncorrelated random amplitudes.

A better characterization of the perturbation is given by defining the *dimensionless density contrast*

$$\delta(\mathbf{x},t) \equiv \frac{\rho(\mathbf{x},t)-\bar{\rho}(t)}{\bar{\rho}(t)}, \tag{1.1}$$

where $\rho(\mathbf{x},t)$ is the local matter density and $\bar{\rho}(t)$ is the cosmic mean density at time t, both defined in comoving coordinates $\mathbf{r} = a(t)\mathbf{x}$, where a accounts for the cosmic expansion. It is possible to identify over-dense and under-dense regions of the cosmos by the sign of δ. The initial perturbation is modeled as a Gaussian random field with zero mean, and in Fourier space is characterized by the Fourier modes

$$\delta(\mathbf{k},t) = \int d^3x\, \delta(\mathbf{x},t) e^{-i\mathbf{k}\cdot\mathbf{x}} \tag{1.2}$$

Its statistical properties are completely described by the power spectrum $P(k)$, defined via the Fourier transform as

$$\langle \delta(\mathbf{k},t)\, \delta^*(\mathbf{k}',t)\rangle = (2\pi)^3\, \delta_D(\mathbf{k}-\mathbf{k}')\, P(k,t),$$

where δ_D is the Dirac delta function and $\langle\cdot\rangle$ denotes an ensemble average. Intuitively, $P(k,t)$ measures the variance of fluctuations at a given scale k, with large k corresponding to small-scale structure and small k to large-scale structure. Because the density field is

Gaussian, it is fully specified by this variance together with independent random phases for each Fourier mode; all higher-order connected moments vanish [12]. This property allows the field to be naturally decomposed into independent Fourier modes,

$$\delta(\mathbf{x}, t) = \int \frac{d^3k}{(2\pi)^3}\, \delta(\mathbf{k}, t)\, e^{i\mathbf{k}\cdot\mathbf{x}}\,.$$

In the early universe, the amplitude of these perturbations is small, $|\delta| \ll 1$. In this linear regime, each mode evolves independently, growing proportionally to the linear growth factor $D(t)$, whose exact shape depends on the details of the cosmological model. Linear theory thus allows precise predictions for both the evolution of large-scale structure and the statistical properties of the density field.

Once $\delta \sim 1$, linear theory breaks down and the evolution enters the nonlinear regime [28]. Here, Fourier modes are no longer independent, and gravitational collapse leads to the formation of bound structures [12], [29]. Over-dense regions undergo shell-crossing and collapse into nodes and filaments, forming the sites where galaxies and clusters emerge, while under-dense regions expand into cosmic voids [12]. These nonlinear interactions produce the cosmic web of filaments and sheets observed in large-scale surveys and simulations [30], [31].
This regime is highly complex and is typically studied using N-body simulations, which capture the full gravitational dynamics and mode coupling. By combining linear theory on large scales and at early times with nonlinear modeling on small scales and at late times, it becomes possible to describe in detail the evolution of DM perturbations within a DE-dominated expanding background. Comparing these theoretical predictions to observational data is crucial for shedding light on the nature of both DM and DE [32], [33].

1.1.1 Dark Matter and Large Scale Structures

Dark matter can be classified according to how fast its particles move in the early Universe, which affects how small structures can form. Three main categories are usually considered: *hot*, *warm*, and *cold* dark matter.
CDM is composed of particles that are non-relativistic at early times. Their slow motion allows even the smallest density fluctuations to grow, so that small clumps of matter collapse first and gradually merge into larger galaxies and clusters. This hierarchical bottom-up sequence of halo formation reproduces what is observed in the Universe.

The ΛCDM model, with cold collisionless dark matter, provides a successful framework to understand the large–scale structure (LSS) of the Universe: it reproduces the detailed pattern of the cosmic microwave background (CMB, the relic radiation from the epoch of recombination), predicts the baryon acoustic oscillation (BAO) feature (a preferred scale left by sound waves in the early plasma) observed in galaxy clustering, and matches measurements of weak gravitational lensing (the small distortions of background galaxy shapes used to map the projected mass distribution). Moreover, the hierarchical or *bottom–up* growth of the structure produces a halo mass function (the number density of dark haloes as a function of mass) that agrees with the observed abundance of galaxy groups and clusters when baryonic effects are included. These successes, verified by CMB, large–scale surveys and N-body simulations, make ΛCDM the standard reference model

for studies of cosmic structure formation. [7], [12], [27].
However, despite its successes on large scales, cold dark matter faces several challenges on smaller scales [34]. For example, the *core–cusp problem* [35] refers to the discrepancy between the steep central density "cusps" predicted by numerical simulations and the shallower cores often inferred from observation of dwarf galaxies. The *missing satellites problem* [36] arises because simulations produce more low-mass subhaloes than the observed number of satellite galaxies, whereas the *too-big-to-fail problem* [37] points out that the massive simulated subhaloes appear denser than the brightest observed satellites. Many of these tensions can be partially mitigated by complex baryonic processes, such as feedback from supernovae or active galactic nuclei, which can redistribute matter and suppress star formation [34], [38], [39], although some discrepancies, particularly the inner density structure of massive subhaloes, may persist even after accounting for such effects [34], [35]. This motivates the consideration of alternative dark matter models, such as fuzzy dark matter [40].

Beyond small scales, tensions also emerge when comparing different cosmological probes. One prominent example is the *Hubble tension*: the value of the Hubble constant estimated from the early Universe (via the CMB under ΛCDM assumptions) disagrees with late-Universe direct measurements from supernovae [7], [41]. Similarly, a milder but persistent discrepancy, sometimes called S_8 *tension*, appears between the clustering amplitude predicted from the CMB and that inferred from weak-lensing surveys [42]. These tensions may point to unaccounted systematics, more complex baryonic physics, or the need for extensions to the standard cosmological model. Finally, the identity of particles forming the dark matter remains unknown, as direct and indirect searches have yet to produce a conclusive detection [43].

Beyond the cold dark matter paradigm discussed above, alternative scenarios have been proposed based on the thermal properties of DM candidates. Hot dark matter (HDM), for instance, consists of particles that remain relativistic when cosmologically relevant modes cross the horizon. Standard Model neutrinos are a classic example. Due to their high velocities, HDM particles tend to erase small-scale density fluctuations, suppressing the growth of structure on these scales. This leads to a *top-down* scenario in which only very large structures collapse first, with smaller structures forming later through fragmentation [28], [44]. In contrast, cold dark matter allows small perturbations to collapse first, leading to a *bottom-up* hierarchical formation of structures [45], [46]. Observations of galaxies and the cosmic microwave background indicate that HDM cannot constitute the main form of dark matter.
Warm dark matter WDM corresponds to particles with intermediate thermal velocities that erase perturbations below a characteristic cutoff scale. This reduces the number of tiny galaxies compared to the cold dark matter case. A well-known candidate for warm dark matter is the sterile neutrino with a mass of a few keV. WDM has been proposed as an alternative to the CDM paradigm because it suppresses the formation of low-mass galaxies while still matching observations on large scales [47]. In addition to WDM, another class of dark matter models, self-interacting dark matter (SIDM), has been suggested to address small-scale tensions. SIDM introduces non-gravitational scattering between dark matter particles, which can flatten central density profiles and modify halo structures [48].
We focus on cold dark matter for several reasons. First, it is the best-tested model on

large scales and agrees well with most cosmological observations. Second, most theoretical predictions and numerical simulations are designed for CDM, which allows for direct comparisons with existing studies. Third, while alternative models (like WDM [49] or SIDM [50]) can change the properties of very small galaxies, these effects are beyond the focus of this work. Therefore, assuming dark matter is cold and collisionless provides a solid and widely accepted framework for studying the formation of cosmic structures.

1.1.2 The Vlasov-Poisson equation

The focus of this Thesis is the nonlinear regime of structure formation; thus, it is necessary to introduce the equations that formally govern the evolution of DM in this regime.

Under the fluid assumption, DM is described as a collisionless, self-gravitating fluid. The phase-space distribution of massive fluid elements at time t is characterized by the *distribution function* $f(\mathbf{r}, \mathbf{v}, t)$, where $\mathbf{r}, \mathbf{v} \in \mathbb{R}^3$ represent the positions and velocities of the fluid elements. The phase-space density $f\, d\mathbf{r}\, d\mathbf{v}$ within the 6D volume element $d\mathbf{r}\, d\mathbf{v}$ represents the probability of finding a fluid element at position $\mathbf{r}$ with velocity $\mathbf{v}$. The density field in configuration space is obtained by integrating over velocity space: $\rho(\mathbf{r}, t) = \int f(\mathbf{r}, \mathbf{v}, t)\, d\mathbf{v}$.

The evolution of this collisionless fluid obeys a continuity equation in phase space, $df/dt = 0$, expressed by the collisionless Boltzmann equation:

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{r}} f - \nabla_{\mathbf{r}} \phi \cdot \nabla_{\mathbf{v}} f = 0, \tag{1.3}$$

where $\nabla_{\mathbf{r}} = (\partial_x, \partial_y, \partial_z)$ and $\nabla_{\mathbf{v}} = (\partial_{v_x}, \partial_{v_y}, \partial_{v_z})$ are respectively the gradient in configuration and in velocity space. The gravitational potential ϕ is determined by the Poisson equation [51]:

$$\nabla^2_{\mathbf{r}} \phi(\mathbf{r}, t) = 4\pi G \int f(\mathbf{r}, \mathbf{v}, t)\, d\mathbf{v} = 4\pi G \rho(\mathbf{r}, t). \tag{1.4}$$

Direct discretization of the Poisson equation for the absolute mass density introduces numerical ambiguities related to boundary conditions and the treatment of the homogeneous background density [52]. In practice, Fourier-based methods solve the Poisson equation for the density contrast, which naturally arises when the equations are expressed in comoving coordinates. This formulation removes the contribution of the uniform background density and leads to a well-defined discretization of the gravitational potential.

In an expanding universe, it is convenient to adopt comoving coordinates $\mathbf{x} = \mathbf{r}/a(t)$ and peculiar velocities $\mathbf{v} = a(t)\dot{\mathbf{x}}$, where $a(t)$ accounts for cosmic expansion. In this framework, the distribution function evolves according to

$$\frac{\partial f}{\partial t} + \frac{\mathbf{v}}{a} \cdot \nabla_{\mathbf{x}} f - \nabla_{\mathbf{v}} f \cdot \nabla_{\mathbf{x}} \Phi = 0, \tag{1.5}$$

where we have introduced $\Phi(\mathbf{x}, t)$, the *peculiar gravitational potential*, defined as the deviation from the homogeneous background potential. This potential satisfies the modified Poisson equation in comoving coordinates:

$$\nabla^2_{\mathbf{x}} \Phi = 4\pi G a^2 \left(\rho(\mathbf{x}, t) - \bar{\rho}(t)\right) = 4\pi G a^2 \bar{\rho}\, \delta(\mathbf{x}, t), \tag{1.6}$$

where the source term is expressed in terms of the density perturbation $\delta(\mathbf{x}, t)$. The factors of $a(t)$ naturally account for the dilution of density and the scaling of velocities due to cosmic expansion.

However, since the aim of this Thesis is not to produce accurate cosmological simulations but rather to develop quantum algorithms, we adopt a simplified static universe approximation with $a = 1$. Within this approximation, density fluctuations grow slightly faster, remain higher than in an expanding universe, and velocities are not damped by cosmic expansion. In contrast, in an expanding universe, the growth of structures is slowed, matter densities are diluted over time, and physical velocities are reduced due to Hubble friction. For the purposes of this Thesis, focused on algorithm development rather than precise cosmological predictions, a static universe is a reasonable approximation, though proper expansion should be accounted for in accurate cosmological simulations

Consequently, Eqs. (1.5) and (1.6) reduce to the following system:

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f - \nabla_{\mathbf{x}} \phi \cdot \nabla_{\mathbf{v}} f = 0, \tag{1.7}$$

$$\nabla^2_{\mathbf{x}} \phi = 4\pi G \bar{\rho} \left(\frac{\rho(\mathbf{x}, t)}{\bar{\rho}} - 1 \right) = 4\pi G \bar{\rho}\, \delta(\mathbf{x}, t). \tag{1.8}$$

These equations serve as the foundational reference throughout this Thesis for describing DM evolution and will collectively be referred to as the Vlasov-Poisson (VP) system.

1.1.3 Jeans Equations

Fluid-like equations can be obtained from VP leading to the so-called Jeans equations [12], [53]. Integrating Eq. (1.7) in velocity space we obtain the continuity equation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \tag{1.9}$$

where the zeroth velocity moment defines the mass density

$$\rho(\mathbf{x}, t) = \int f(\mathbf{x}, \mathbf{v}, t)\, d^3 v, \tag{1.10}$$

and the bulk velocity field is given by

$$\mathbf{u}(\mathbf{x}, t) = \frac{1}{\rho} \int \mathbf{v} f(\mathbf{x}, \mathbf{v}, t)\, d^3 v. \tag{1.11}$$

The bulk velocity $\mathbf{u}(\mathbf{x}, t)$ is defined as the mass-weighted average velocity at fixed position, obtained by dividing the momentum density by the mass density, ensuring that $\mathbf{u}(\mathbf{x}, t)$ represents the velocity of the local center of mass. The continuity equation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0\,, \tag{1.12}$$

expresses local conservation of mass and is an exact consequence of the Vlasov equation. The first velocity moment (i.e, multiplying by $\mathbf{v}$, and integrating over velocities $\int d\mathbf{v}$) yields

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -\rho \nabla V - \nabla \cdot \mathbf{P}_J, \tag{1.13}$$

where $\mathbf{P}_J$ is the Jeans stress tensor,

$$(P_J)_{ij} = \int (v_i - u_i)(v_j - u_j) f(\mathbf{x}, \mathbf{v}, t)\, d^3v = \rho \left(\langle v_i v_j \rangle - \langle v_i \rangle \langle v_j \rangle \right). \tag{1.14}$$

The stress tensor encodes the velocity dispersion about the mean flow and acts as an effective pressure term. The Jeans equations form an infinite hierarchy, since the evolution of $\mathbf{P}_J$ depends on higher-order velocity moments. In general, this hierarchy cannot be closed without additional assumptions.

Euler equation

A particular class of exact solutions of the VP system is obtained by considering a single-speed distribution, where to each fluid element is associated only one velocity value.

$$f(\mathbf{x}, \mathbf{v}, t) = \rho(\mathbf{x}, t)\, \delta(\mathbf{v} - \mathbf{u}(\mathbf{x}, t)). \tag{1.15}$$

In this case, all particles at a given position share the same velocity and the velocity dispersion vanishes identically, implying $\mathbf{P}_J = 0$. Substituting this ansatz into Eq. (1.13) equations yields the pressureless Euler-Poisson (EP) system,

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \tag{1.16}$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla \Phi, \tag{1.17}$$

$$\nabla^2 \Phi = 4\pi G(\rho - \bar{\rho}). \tag{1.18}$$

These equations are exact as long as the distribution remains single-valued in velocity space. This regime is commonly referred to as the *dust* or *cold* limit of the Vlasov-Poisson system, and is valid until *shell-crossing* occurs [28].

Shell-crossing

The term *shell-crossing* can be better understood in the context of the collapse of a spherical density perturbation in a self-gravitating collisionless fluid [54]. As the Universe expands and matter accretes, spherical shells collapse under gravity until they intersect, leading to the formation of a singularity. The concept has since been applied to dark matter [55], [56], whose collisionless nature makes such intersections particularly relevant.

More generally, shell-crossing occurs whenever the orbits of two or more fluid elements intersect. After this phenomenon, the simple one-to-one correspondence between a fluid element's initial position and its velocity no longer holds, because multiple elements can occupy the same location with different velocities. This breakdown of the mapping between position and velocity is a hallmark of multi-stream regions in collisionless systems, and it complicates the description of the dynamics, since the phase-space distribution becomes multi-valued at a given spatial point.

Isotropic Closure of Jeans equations

However, for generic initial conditions, the Vlasov evolution leads to multi-streaming as particle trajectories cross in configuration space. This process results in the formation

of caustics, where the density formally diverges, and the velocity field becomes multi-valued. Beyond the first shell-crossing event, the single-speed ansatz breaks down, the stress tensor $\mathbf{P}_J$ becomes nonzero, and the pressureless EP description is no longer valid. In this regime, one must revert to the full VP equations or to the unclosed Jeans hierarchy.

A commonly employed phenomenological approach consists in introducing an effective pressure term in the momentum equation,

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\nabla V - \frac{1}{\rho}\nabla P(\rho), \tag{1.19}$$

which corresponds to closing the Jeans hierarchy by assuming an isotropic stress tensor of the form $\mathbf{P}_J = P(\rho)\mathbf{I}$. While this closure can regularize shell-crossing singularities, it provides only an approximate description of velocity dispersion and does not fully capture the underlying phase-space dynamics.

1.2 Numerical methods

In the nonlinear regime, when $|\delta| \sim 1$ the evolution of cosmic density fluctuations becomes strongly nonlinear, making exact analytical treatments impossible. Numerical simulations, therefore, play a central role. Within the ΛCDM model this translates into finding the solution of Eqs. (1.7) and (1.8), once initial conditions on position and velocity of the fluid-elements $f(\mathbf{x}, \mathbf{v}, t_0)$ are assigned to represent an ensemble realization of a given cosmological model [57]. As such, the VP equations must be solved in $6D + 1$ dimensions. Moreover, the dependence of the distribution function on its current shape allows for the insurgence of strongly nonlinear dynamical effects, requiring an adequate resolution in phase space. The combinations of these two factors imply that a direct numerical integration of Eq. (1.7) requires an enormous amount of computational resources.

1.2.1 Eulerian Vlasov-Poisson solvers

Despite these challenges, numerous grid-based approaches have been developed to simulate self-gravitating system dynamics. Notable examples include the pioneering splitting technique [58], the more versatile semi-Lagrangian schemes [59], and finite volume approaches [60]. A comprehensive description of these techniques will be presented in Chapter 3; here we confine our attention to the fundamental principles underlying these simulations.

In a Eulerian Vlasov–Poisson solver, the complete phase space is discretized using a mesh structure. The spatial domain typically consists of a box of dimension L, partitioned into equispaced N^3 cells representing a region of the universe. Periodic boundary conditions are imposed to eliminate boundary effects and to treat the simulated domain as a representative volume embedded in an effectively unbounded, statistically homogeneous Universe. The velocity domain is discretized similarly into M^3 velocity bins spanning the range $V_{\min}$ to $V_{\max}$, with vanishing boundary conditions applied.

The principal strength of this framework lies in its essentially noise-free nature: because no particle sampling is required, the scheme preserves intricate phase-space structures that would otherwise be obscured by discreteness effects inherent to N-body simulations. However, they are substantially resolution-limited: the computational cost grows

rapidly with the number of grid points in both position and velocity space, which restricts the smallest scales that can be accurately resolved [61].

Most contemporary implementations follow the splitting scheme proposed by Cheng and Knorr [58], where the Vlasov–Poisson system is treated as a sequence of advection equations in configuration and velocity spaces:

$$\frac{\partial f(\mathbf{x},\mathbf{v},t)}{\partial t} + \mathbf{v}\cdot\boldsymbol{\nabla}_{\mathbf{x}} f(\mathbf{x},\mathbf{v},t) = 0\,, \tag{1.20}$$

$$\frac{\partial f(\mathbf{x},\mathbf{v},t)}{\partial t} - \boldsymbol{\nabla}\phi(\mathbf{x},t)\cdot\boldsymbol{\nabla}_{\mathbf{v}} f(\mathbf{x},\mathbf{v},t) = 0\,. \tag{1.21}$$

The advection coefficients in Eq. (1.21) are determined by the solution of Poisson's equation (1.8). Obtaining this solution requires computing the density field $\rho(\mathbf{x},t)$ via the velocity-space integration of f.

For periodic domains, the spectral method provides the most efficient solution strategy for Poisson's equation. The gravitational potential $\phi(\mathbf{x})$ is obtained by Fourier transforming the density field $\rho(\mathbf{x}) \mapsto \widehat{\rho}(\mathbf{k})$, applying multiplication by a discrete Green's function $\widehat{G}$,

$$\widehat{\phi}(\mathbf{k}) \;=\; \widehat{G}(\mathbf{k})\,\widehat{\rho}(\mathbf{k}),$$

and perform an inverse transform $\widehat{\phi}(\mathbf{k}) \mapsto \phi(\mathbf{x})$. The spectral approach proves particularly well-suited for Poisson's equation, as the Laplacian operator $\boldsymbol{\nabla}^2$ yields the simple-man Green Function $\widehat{G} \propto -1/k^2$ in Fourier space. However, in finite and discontinuous domains the form of the Green's function is different [52].

The foremost limitation remains the prohibitive memory requirements. As the mesh must encompass the full six-dimensional phase space, the grid cell count scales as $N_x^3 N_v^3$, rapidly becoming computationally intractable for cosmological applications. Even with semi-Lagrangian techniques that permit relatively coarse velocity discretizations, Vlasov solvers are typically confined to lower-dimensional configurations (e.g., one or two spatial dimensions) or to idealized problems where fine phase-space resolution proves indispensable.

Notwithstanding these constraints, Eulerian Vlasov–Poisson methods constitute an invaluable tool for investigating regimes where phase-space architecture plays a critical role and particle noise must be suppressed. Their application to fully three-dimensional cosmologically relevant scenarios remains computationally challenging, although proof-of-concept studies have been performed [61].

1.2.2 N-body methods

A widely used approach to reduce the dimensionality of the problem is to model the initial DM distribution as an ensemble of collisionless massive fluid-elements interacting only through self-gravity. This set formally obeys the *Euler-Poisson* (EP) equations, a closure of the VP equations obtained by asking the distribution function to be single-valued in space[62]–[64]. These particles are sampled from a starting distribution describing the cosmological initial conditions. Taking advantage from the isotropy and uniformity of the Universe at large scales, one represents a specific region of it as a periodic cubic volume populated by N_p massive particles. The goal is to model the formation and evolution of cosmic structures under the action of gravity, in an expanding background. To ensure

that this volume captures the relevant statistical properties of the Universe, its size must exceed the scale at which the matter distribution becomes homogeneous. Simulations typically impose periodic boundary conditions so that the finite domain mimics an infinite cosmos and so that Fourier-based techniques can be used efficiently.
Modern N-body methods differ primarily in how these forces are evaluated. In the following, we provide an introduction to the most widely known approaches.

Direct Summation

The simplest way to follow the nonlinear evolution of a cosmological fluid is to represent it as a discrete set of particles and compute the gravitational forces between all pairs directly. These are known as *particle-particle* (PP) methods. By advancing particle velocities and positions in small timesteps using the computed accelerations, one can iteratively update the system.

It is important to note that the particles do not represent actual discrete objects; they are a numerical sampling of a continuous fluid. A practical issue with PP calculations arises because the Newtonian force diverges as two particles approach each other. Resolving such close encounters would require prohibitively small timesteps and immense computational resources, and formal divergences cannot be handled numerically.

This problem is commonly addressed by introducing a softening length h, which effectively replaces each particle by an extended mass distribution of size $\sim h$. The modified force between particles at positions $\mathbf{q}_i$ and $\mathbf{q}_j$ with equal mass m becomes

$$\mathbf{F}_{ij} = Gm^2 \frac{\mathbf{q}_j - \mathbf{q}_i}{(h^2 + |\mathbf{q}_i - \mathbf{q}_j|^2)^{3/2}}. \tag{1.22}$$

This prevents infinite forces at zero separation and suppresses unphysical two-body interactions on scales below h. Consequently, the particle distribution cannot be trusted on scales $\lesssim h$, but this is consistent with the goal of simulating a smooth fluid rather than individual point masses.

For a simulation with N_p particles, direct summation requires evaluating pairwise forces $N_p(N_p - 1)/2$ at each time step. This $\mathcal{O}(N_p^2)$ scaling is the primary limitation of PP methods: they become prohibitively slow for $N_p \gtrsim 10^6$, which is insufficient for realistic large-scale structure simulations. This method is in fact not used anymore for cosmological simulations, due to the unfavorable scaling.

Particle-Mesh and P^{3}M methods

A widely adopted alternative to direct pairwise summation is the *particle-mesh* (PM) approach, in which particle masses are first assigned to a regular grid in order to produce a density $\rho(\mathbf{x})$ that will be used to solve the Poisson's equation on that mesh. Subsequently, the acceleration on each particle is evaluated from the expression of the force on the grid, via the same interpolation scheme used to assign the density values and their positions are updated accordingly. Even in this scenario, before anything is done, the particles are sampled from a distribution that represents the cosmological initial condition.

A compact description of the PM pipeline is as follows:

1. **Mass assignment.** Particles at positions $\mathbf{q}_i$ are deposited on the mesh using a window function W, producing the grid density $\rho(\mathbf{x})$. Let us define with N_p the number of particles in box of dimension L^3 and N^3 gridpoints. N represent the number of gridpoints used to discretize one side of the cosmological box and $\Delta x = L/N$ is the resolution along one axis. Consequently the discrete vector position can be written as $\mathbf{x} = \mathbf{n}\, L/N$. The expression of the density is given by

$$\rho(\mathbf{x}) \;=\; M^3 \,\frac{1}{N} \sum_{i=1}^{N} W(\mathbf{q}_i - \mathbf{x}).$$

   Common choices for W are Nearest Grid Point (NGP), where each particle mass is assigned to the cell where it resides; Cloud In Cell (CIC) where the kernel W is a top-hat function; and Triangular Shaped Cloud (TSC) as suggested by the name consist in a triangular shaped kernel.

2. **Poisson solver** The next step consists of solving Eq. (1.8) using the density field $\rho(\mathbf{x})$ obtained in the previous step. A typical approach employs a spectral method, as discussed for Eulerian VP solvers. This yields the gravitational potential in position space, defined on the same grid as $\rho(\mathbf{x})$.

3. **Force evaluation and interpolation.** Grid forces $\mathbf{F}(\mathbf{x})$ follow from finite-differencing or spectral differentiation of ϕ,

$$\mathbf{F}(\mathbf{x}) \;=\; -\,\mathcal{D}\,\phi(\mathbf{x}),$$

   and the forces on particles are obtained by interpolating the grid field back to particle positions with the same kernel W:

$$\mathbf{F}(\mathbf{q}_i) \;=\; \sum_{\mathbf{x}} W(\mathbf{q}_i - \mathbf{x})\, \mathbf{F}(\mathbf{x}).$$

4. **Position update** Knowing the force on each particle, each position is updated accordingly to the chosen time integration scheme. The most naive way is to update the velocity of each particle first $\mathbf{v}_i \mapsto \mathbf{v}_i' = \mathbf{F}(\mathbf{q}_i)\,\Delta t/m$ and then update its position $\mathbf{q}_i \mapsto \mathbf{q}_i' = \mathbf{q}_i + \mathbf{v}_i\,\Delta t$.

Because FFTs cost $\mathcal{O}(N_g \log N_g)$ for $N_g = N^3$ grid points, PM is vastly faster than $\mathcal{O}(N_p^2)$ direct summation for large particle numbers. The trade-off is spatial resolution: the mesh enforces a force resolution on scales comparable to a few grid cells, so PM alone is inadequate when accurate short-range forces are required.

The *particle-particle-particle-mesh* ($\mathrm{P^3M}$) hybrid remedies this by combining the PM long-range solver with an explicit short-range correction. One defines a short-range radius r_s (typically a few mesh units) and adds a direct pairwise correction for particle pairs with separation $|\mathbf{q}_i - \mathbf{q}_j| < r_s$. To avoid double counting the correction is constructed to cancel the smoothed mesh contribution at small separations, or equivalently one can adopt a modified short-range force law (analogous to gravitational softening). Because computing many short-range pairs can be costly in clustered regions, a common optimization is to precompute the short-range force as a function of radius on a finely spaced grid and obtain

pair forces by interpolation from this lookup table. The long-range field is still obtained by the PM/FFT procedure, so the scheme retains the PM scaling for the majority of the computational work while recovering near-pair accuracy where it matters.

PM and P^3M variants, with differing choices of assignment window, softening scale and short-range cutoff, form the backbone of contemporary cosmological N-body simulations, offering a flexible compromise between speed and small-scale fidelity.

Tree codes

A different strategy for improving force resolution while keeping computational costs manageable is to employ a hierarchical decomposition of the simulation volume. Methods based on this principle are collectively known as *tree codes*. The central concept is to approximate the gravitational influence of distant particle groups by replacing them with aggregate "pseudo-particles" that represent their collective mass distribution.
The standard implementation constructs a hierarchy of nested cells, known as the Barnes-Hut tree [65]: the simulation domain is recursively subdivided into eight subcells (forming an octree structure) whenever a cell contains more than one particle. Subdivision continues until each leaf node contains at most a single particle. With this hierarchy established, forces on a given particle can be computed as efficiently as $\mathcal{O}(N_p \log N_p)$ by modeling distant regions using coarse, high-level cells, while resolving nearby particles at finer tree levels.

Although conceptually straightforward, tree codes introduce practical complexities, particularly in maintaining consistent particle information throughout the tree structure. They also require substantially more memory than particle-mesh approaches at comparable resolution. The main obstacle to these implementations is again the computational cost, particularly memory requirements.

1.2.3 The Schrödinger method

An alternative to N-body methods is provided by *Schrödinger method* (SM), where the Schrödinger-Poisson (SP) equation is solved instead of the VP system. In the limit where the dynamic scale of the problem, $\hbar/m$, becomes very small, the VP results can be recovered. This method was proposed by Widrow and Kaiser in the 1990s [66], but it did not attract widespread attention at the time, as their results were limited to one-dimensional implementations. However, only recently, following a work by Mocz et *al.* [53] that showed numerical correspondence between the $6D{+}1$ VP and the $3D{+}1$ SP equations for cosmological simulations revived the interest in simulating and studying various forms of dark matter, which can be modeled by the SP equation [1], [32], [67].

The SM maps the initial classical distribution $f(\mathbf{x}, \mathbf{v}, t_0)$ to the quantum wavefunction $\Psi(\mathbf{x}, t_0)$ by means of a nonlocal operation. Details about this procedure are given in [53]; here we just provide a brief overview of the method. We consider two primary scenarios. In instances where the initial distribution function is characterized by a cold or single-valued stream, meaning that a unique velocity corresponds to each point, it is possible to directly reconstruct the phase S of the quantum wavefunction using the Madelung representation

$$\Psi(\mathbf{x}, t) = \sqrt{\rho(\mathbf{x}, t)}\, \exp\{iS(\mathbf{x}, t)/\hbar\}\,, \tag{1.23}$$

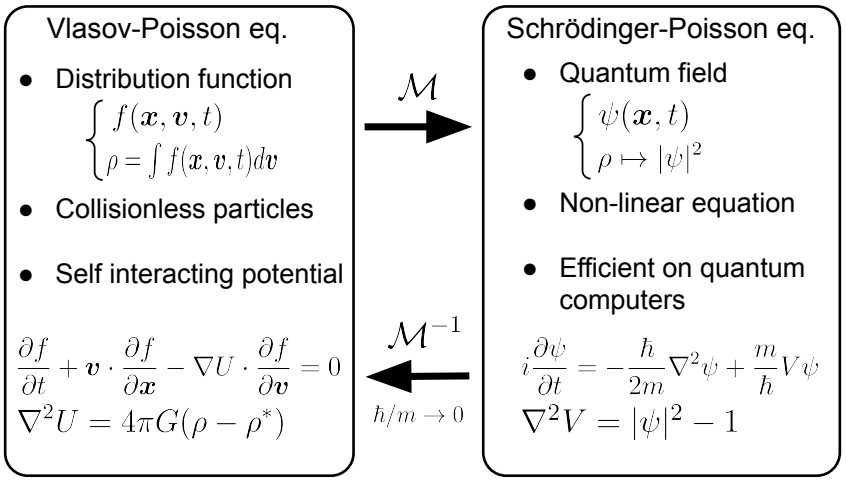


Figure 1.1: Mapping, $\mathcal{M}$, of the classical $N-$body Vlasov problem into the corresponding quantum Schrödinger Poisson formulation obtained through nonlocal manipulation (e.g, Husimi smoothing [68]). Detail on the mapping $\mathcal{M}$ and its inverse are given in [53], [66].

where $\rho = \int d\boldsymbol{v} f(\boldsymbol{x},\boldsymbol{v},t)$ through the solution of the Poisson problem,

$$\boldsymbol{\nabla}\cdot\boldsymbol{v} = \nabla^2 S/m\,, \tag{1.24}$$

where the scale $\hbar/m$ emerges as an effect of quantization.

In the scenario involving multi-streams or warm initial conditions, where a single grid point may correspond to two or more different velocity values, the situation becomes more complex. This complexity arises because the densities do not precisely coincide, and the quantum wavefunction incorporates interference patterns. In this case, the mapping from phase-space distribution function to wavefunction reads

$$\Psi(x) \propto \sum_{\boldsymbol{v}} \sqrt{f(\boldsymbol{x},\boldsymbol{v})} e^{\frac{i}{\hbar} m\,\boldsymbol{x}\cdot\boldsymbol{v} + 2\pi i \phi_{rand,\boldsymbol{v}}} \Delta\boldsymbol{v}\,, \tag{1.25}$$

where, we sum over sampled velocities $\boldsymbol{v}$, each with an associated random phase $\phi_{rand,\boldsymbol{v}} \in [0, 2\pi)$ to ensure uncorrelated phases for each fluid velocity [53].

The wavefunction then evolves according to the SP system of equations

$$i\hbar\frac{\partial\Psi}{\partial t} = -\frac{\hbar^2}{2m}\nabla^2\Psi + mU\Psi\,; \tag{1.26}$$

$$\nabla^2 U = 4\pi G(\rho - \rho^*)\,. \tag{1.27}$$

Here we have chosen to use a density contrast $\rho - \rho^*$ as the source of the gravitational potential, where ρ^* represents the average density over the volume considered.

The correspondence with the classical fluid description becomes explicit in the moment we write the Eqs. (1.26) for the wavefunction in Madelung representation (1.23). The imaginary part of the equation return the continuity equation for ρ

$$\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\mathbf{v}) = 0, \tag{1.28}$$

where $\mathbf{v}$ is emerges naturally as defined in Eq. (1.24). The real part gives a modified Hamilton–Jacobi equation for the phase S:

$$\frac{\partial S}{\partial t} + \frac{(\nabla S)^2}{2m} + m\Phi - \frac{\hbar^2}{2m}\frac{\nabla^2\sqrt{\rho}}{\sqrt{\rho}} = 0, \tag{1.29}$$

where the last term represents the quantum fluctuations of the amplitude of the wavefunction

$$Q = -\frac{\hbar^2}{2m}\frac{\nabla^2\sqrt{\rho}}{\sqrt{\rho}}\,. \tag{1.30}$$

By taking the gradient of this equation, one obtains the evolution equation for the velocity field:

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v}\cdot\nabla)\mathbf{v} = -\nabla\Phi - \nabla Q, \tag{1.31}$$

where $-\nabla Q$, acts as a repulsive force in regions of high density, counteracting gravitational collapse, in a similar way of classical pressure, as shown in Eq. (1.19). For this reason, Q is commonly referred to as the quantum pressure term, although its origin lies not in thermal fluctuations but in the quantum mechanical nature of the system itself. In the limit $\hbar \to 0$, the quantum term vanishes, and the EP equation (1.17) is recovered.

We note that in this approach the aim of Eq. (1.26) is to describe a density field, not a particle's wavefunction. Note also that the constant $\hbar$ and m are not the Planck constant and the mass of the particle but are related respectively to the quantum and classical effects (see discussion below in Sect. 4.1 and in the Appendix A.1 for details about the scale of the equation). In fact, by defining the quantum diffusion coefficient $\lambda = \hbar/m$ as the ratio between the quantum effects ad classical charge m we can rewrite the Schrödinger Equation in (1.26) in a manner where the scale dependence is more explicit

$$i\frac{\partial \Psi}{\partial t} = \left(-\frac{\lambda}{2}\nabla^2 + \frac{1}{\lambda}U\right)\Psi\,. \tag{1.32}$$

Once the Schrödinger-Poisson evolution is completed, the distribution function can be extracted from the final wavefunction using the Husimi procedure, which is a smoothed version of the Wigner quasi-probability distribution. A similar approach can be found in [69] for the solution of the Vlasov equation with electromagnetic fields in plasma physics applications. In a $3D$ context, this operation can be seen as the spatial smoothing of the wavefunction Ψ with a Gaussian filter of width η. Additionally, it involves a Fourier-like transformation to extract momentum information,

$$\begin{aligned}\tilde{\Psi}(\boldsymbol{x},\boldsymbol{p},t;\eta) = &\left(\frac{1}{2\pi\hbar}\right)^{3/2}\left(\frac{1}{\pi\eta^2}\right)^{3/4} \\ &\times \int d^3r\,\Psi(\boldsymbol{r},t)\exp\left\{\left(-\frac{(\boldsymbol{x}-\boldsymbol{r})^2}{2\eta^2} - i\frac{\boldsymbol{p}\cdot(\boldsymbol{r}-\boldsymbol{x}/2)}{\hbar}\right)\right\}.\end{aligned} \tag{1.33}$$

The squared module of the wavefunction in Eq. (1.33) yields a result that closely approximates the desired distribution function [53], [68].

In Fig. 1.1 a summary of what has been said in this subsection is presented, presenting the SM method as an alternative approach to VP.

As a side note, we remind that the SP equations have already been used in the numerical study of cosmic structure formation to study the dynamics of the Fuzzy Dark Matter (FDM) perturbations [70], [71]. This class of DM candidates emerges as the ultra-light mass limit of a scalar bosonic field, whose particles are known as axions. In this case $\hbar$ represents, in fact, the actual Planck constant and m the mass of the axion-like particles. The characteristic scale of the problem is the ratio $\hbar/m$: at smaller scales, the dynamics is influenced by quantum effects such as quantum pressure, while at larger scales, this effect becomes negligible and the classical Cold Dark Matter (CDM) limit is recovered.

1.3 Summary

In this chapter, we introduced the VP equation as the main model used to describe the evolution of perturbations in the distribution of DM during the evolution of the Universe. We highlighted the importance of numerical simulations of DM distribution as the backbone of cosmological simulations. We also introduced the three main methods used for the numerical solution of VP: The direct approach, the N-body method and the Schrödinger method. We delved into the SM, noting how it should be possible to recover VP in the small scales regime. However, the SM does not specify the exact scale on which the correspondence is correct nor gives a specific error scaling of the approximation as a function of $\hbar/m$. In the next chapter, we aim to investigate this specific topic: how accurate SP approximate VP, and how the exact scale $\hbar/m$ that enables this corresponce to be correct. Moreover, our aim is to inquire about the accuracy of the N-body method with respect to direct VP integration. In order to do so, we will develop a numerical solver for all the three different methods and compare the results in different scenarios.

Chapter 2

Introduction to Quantum Computing

This chapter introduces the conceptual and physical foundations of quantum computation that will be used throughout the rest of the Thesis. The goal is not to provide an exhaustive treatment but to establish a clear and consistent framework for understanding how quantum algorithms encode information, manipulate it, and extract computational results. We aim to explain how quantum mechanics influences the three main parts of computation: data encoding, operations, and data decoding.

We begin in Section 2.1 by recalling the classical model of computation. Starting from Turing machines and Boolean circuits, we emphasize the input–output structure of algorithms and the role of bits as carriers of information. This discussion motivates the search for alternative computational paradigms and sets the stage for quantum computation by highlighting the physical nature of information processing.

In Section 2.2, we introduce quantum information and the qubit as the fundamental unit of quantum computation. We discuss superposition and explain how quantum registers store information in quantum states whose dimension grows exponentially with the number of qubits. The section then introduces entanglement as a uniquely quantum resource, showing how correlations between qubits arise and why they play a central role in quantum algorithms.

Section 2.3 presents the circuit model of quantum computation. We describe how quantum operations are represented as unitary gates acting on qubits, drawing a direct analogy with classical circuits. Single- and multi-qubit gates are introduced, together with the notion of a quantum register and ancillary qubits. Particular emphasis is placed on the decomposition of complex operations into a set of native gates, which provides a practical way to analyze circuit cost in terms of gate count and depth.

In Section 2.4, we move from the abstract circuit model to its physical limitations. We introduce noise and decoherence as unavoidable consequences of imperfect isolation and control, and explain how they constrain the depth and reliability of quantum circuits. This discussion naturally leads to the distinction between near-term noisy devices and fully fault-tolerant quantum computers. We also introduce the two main computational regimes, related to state of hardware development. We discuss how the presence or absence of quantum error correction influences algorithm design, resource requirements, and the practical relevance of different classes of quantum algorithms.

Finally, Section 2.5 completes the computational cycle by addressing quantum measurement. We explain how classical information is extracted from a quantum register,

emphasizing the probabilistic nature of measurement, the collapse of the quantum state, and the need for repeated circuit executions to estimate expectation values. This section establishes a direct parallel with the output stage of a classical Turing machine and closes the conceptual loop of quantum computation: initialization, evolution, and readout.

Together, these sections provide the minimal but complete framework required to analyze quantum algorithms and their implementation, which will be the focus of the following chapters.

2.1 Classical Computation and Its Limitations

Modern computer science is built on the precise and abstract notion of computation. Before introducing quantum computation, it is essential to understand the classical computational framework and the reasons that motivate the search for alternative models.

2.1.1 Turing Machines

At the heart of classical computation lies a simple but fundamental idea: a computation is a process that transforms an *input* into an *output* by following a finite set of well-defined instructions. This idea is formalized by the *Turing machine* model [72], which provides a precise definition of what it means to execute an algorithm.

A Turing machine consists of three essential components. First, it has a tape that serves as memory and initially contains the input of the computation, encoded as a finite sequence of symbols. Second, it has a read–write head that can inspect one tape cell at a time and modify its content. Third, it has a finite control, which stores a fixed set of rules—also called the machine's program or the instruction table.

The computation proceeds step by step. At each step, the machine reads the current symbol on the tape and, based on this symbol and its internal state, applies one instruction from its program. This instruction specifies three actions: which symbol to write on the tape, how to move the head (left or right), and which internal state to transition to next. Through this process, the Turing machine deterministically follows its instructions, transforming the input into an output encoded on the tape.

From this perspective, a Turing machine captures the essential structure of any algorithm: an algorithm is a finite list of instructions that, when applied systematically to an input, produces an output. The specific physical realization of the machine is irrelevant; what matters is the abstract notion of computation as rule-based information processing.

Despite its simplicity, the Turing machine is powerful enough to model any classical algorithm. This observation underlies the *Church–Turing Thesis*, which asserts that every function that can be effectively computed by an algorithm can be computed by some Turing machine. As a result, the Turing machine serves as the foundational reference model for classical computation and the theory of algorithms.

2.1.2 Boolean Circuits and Binary Information

While the Turing machine provides a fundamental and abstract description of computation, it is often useful to adopt a model that more closely resembles how computations

are carried out in real hardware. This naturally leads to the *Boolean circuit* model of classical computation.

In the circuit model, information is represented using *bits*, the basic units of classical information. A bit can take only one of two possible states, commonly denoted as 0 and 1, which can also be interpreted as *OFF* and *ON*. Input data are encoded as a finite string of bits, typically written in binary form as a sequence such as 0101 .... This binary encoding provides a uniform way to represent numbers, characters, and more complex data structures.

A Boolean circuit operates on this input string by applying a finite sequence of *logical operations*. These operations are implemented by logical gates, such as AND, OR, and NOT, each of which computes a simple Boolean function on one or more input bits [73]. For example, an AND gate outputs 1 if and only if both of its input bits are equal to 1, while a NOT gate flips the value of its input bit. By combining these elementary gates into larger circuits, it is possible to compute arbitrarily complex functions.

The computation performed by a Boolean circuit follows a clear structure: an input encoded in binary form is processed by a network of gates, and the result of the computation is produced as an output string of bits. Unlike the step-by-step evolution of a Turing machine, a circuit computes its output through the parallel evaluation of its gates, which makes the circuit model particularly useful for analyzing efficiency and resource usage.

From a physical perspective, Boolean circuits are realized in modern microprocessors using electronic components such as transistors. At a very high level, a transistor can act as an electrically controlled switch, allowing or preventing the flow of current depending on the voltage applied to its control terminal. By arranging transistors into specific configurations, it is possible to implement the basic logical gates of Boolean logic [74].

In this way, abstract logical operations are directly mapped onto physical processes: voltage levels encode binary values 0 and 1, transistors implement logical rules, and the coordinated activity of billions of such components enables classical computation at large scales. Despite the enormous complexity of modern processors, their fundamental operation remains grounded in the manipulation of binary information through classical logical gates.

Classical computation is fundamentally deterministic. Given a fixed input and a fixed set of instructions, a classical algorithm always produces the same output. Whether the computation is described in terms of a Turing machine or a Boolean circuit, the evolution of the computational state is entirely determined by classical rules. Even when randomness is introduced through probabilistic algorithms, this randomness reflects uncertainty or external sampling, not a change in the underlying deterministic nature of the computational model.

2.1.3 Physical Limits, and the Need for New Computational Paradigms

At the heart of boolean computation lies classical physics. Bits are stored using macroscopic physical quantities, such as voltage levels, and logical operations are implemented by electronic components whose behavior is accurately described by classical electrodynamics. For many decades, this classical description has been sufficient for designing faster and more powerful computing devices.

As transistors have grown smaller and smaller, quantum mechanics has become unavoidable in understanding and engineering how devices behave. Phenomena such as tunneling, discrete energy levels, and atomic-scale variability now shape the very design of modern chips. Even so, these effects do not change the fact that classical computers operate on classical bits. Quantum behavior in the devices is carefully managed so that processors continue to run familiar algorithms reliably. Meanwhile, the historic pace of performance growth captured by Moore's law, which observed that transistor counts roughly doubled every two years, is slowing down. Further miniaturization is becoming increasingly difficult, costly, and energy intensive, and simply making transistors smaller can no longer guarantee faster or more powerful computers. Quantum computing offers a different approach. Instead of relying on smaller components, it uses the principles of quantum mechanics to store and manipulate information in ways that classical devices cannot, opening the door to fundamentally new forms of computation.

From this perspective, quantum computation can be understood as a modification of the classical computational framework at two fundamental levels. First, the way information is stored must change: classical bits are replaced by quantum states, which can exist in superpositions of classical values. Second, the operations that process information must also change: the classical transition rules of a Turing machine or the logical gates of a Boolean circuit are replaced by quantum operations that evolve quantum states according to the principles of quantum mechanics.

These two changes—quantum information storage and quantum information processing—define the essential departure from classical computation and open the door to computational behaviors that do not have a classical counterpart.

2.2 Quantum Computation: Information and States

As discussed in the previous section, while classical computation relies on deterministic bits and classical logic, quantum computation redefines both the fundamental units of information and the operations that act on them, drawing directly from the principles of quantum mechanics.

2.2.1 Qubits: The Basic Unit of Quantum Information

In quantum computation, the classical bit is replaced by the *quantum bit*, or *qubit*. Unlike a classical bit, which can only take the values 0 or 1, a qubit can exist in a *superposition* of both states simultaneously. Formally, the state of a qubit is written as

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle,$$

where α and β are complex numbers that satisfy $|\alpha|^2 + |\beta|^2 = 1$. Upon measurement on the computational basis, the qubit is found in state $|0\rangle$ with probability $|\alpha|^2$ and in state $|1\rangle$ with probability $|\beta|^2$. This probabilistic nature is intrinsic and reflects the fundamental difference between classical and quantum information.

Importantly, qubits are not static objects: they can be *manipulated* using carefully designed operations that change their quantum state. These single-qubit operations allow a qubit to move from one superposition to another or change its relative phase.

A qubit can be physically realized in many different systems. Some common examples of single qubits, together with how they are manipulated, include:

- **Photon polarization**: The logical states $|0\rangle$ and $|1\rangle$ are represented by horizontal and vertical polarization. Single-qubit operations are implemented using waveplates or polarizers, which rotate the polarization state and create superpositions [18], [75].
- **Electron spin**: A single electron trapped in a quantum dot or atom encodes the qubit in its spin state. Magnetic fields and microwave pulses are used to rotate the spin, preparing arbitrary superpositions of $|0\rangle$ (spin-up) and $|1\rangle$ (spin-down) [76].
- **Atomic or ionic energy levels**: Two energy levels of a trapped ion or a neutral atom define the qubit. Laser pulses induce transitions between these levels, allowing for precise control of the qubit state and the preparation of superpositions [77].
- **Superconducting circuits**: In devices such as transmon qubits, the two lowest energy levels of a nonlinear superconducting oscillator represent $|0\rangle$ and $|1\rangle$. Microwave pulses are applied to perform rotations and phase shifts, enabling arbitrary single-qubit operations [78].

Each of these systems allows a single qubit to be initialized, manipulated, and measured. Although a single qubit is the basic component of a quantum computer, it already exhibits uniquely quantum behaviors such as superposition, which have no classical analog.

It is important to note that a single qubit is just a *building block* of the full information storage unit, much like a classical bit is a component of a classical register in a circuit. By combining multiple qubits, we can construct quantum registers that store and process quantum information in ways that extend beyond the classical possibilities. This naturally leads to the concept of multi-qubit *quantum states*, which will be discussed in the next subsection.

2.2.2 Quantum States, Superposition, and Entanglement

A single qubit, as we have seen, can exist in a superposition of the classical basis states $|0\rangle$ and $|1\rangle$, and can be manipulated through single-qubit operations. Although this represents a departure from classical bits, the full power of quantum computation emerges when multiple qubits are considered together.

When two or more qubits are combined, they form a *quantum register* whose overall state is described by a vector in a Hilbert space of dimension 2^n for n qubits. Given two single qubits in the state $|0\rangle$, we denote by the tensor product the global state

$$|0\rangle \otimes |0\rangle = |00\rangle \tag{2.1}$$

The general state can be written as

$$|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle,$$

where the coefficients α_i are complex numbers called *probability amplitudes*, satisfying $\sum_i |\alpha_i|^2 = 1$. Each basis state $|i\rangle$ corresponds to a distinct combination of classical bit values. Actually, $|i\rangle$ is the binary representation of the integer i expressed on a n qubit register. The generic expression for a two-qubit state is

$$|\psi\rangle = \alpha_0 |00\rangle + \alpha_1 |01\rangle + \alpha_2 |10\rangle + \alpha_3 |11\rangle \ , \tag{2.2}$$

Recall that this is a generic expression and the coefficient α_i could assume any value, provided that they respect the normalization rule $\sum_i |\alpha_i|^2 = 1$.

Superposition as a Quantum Memory Resource

One of the most immediate consequences of superposition is its impact on how information is stored. In classical computation, a register of n bits can represent exactly one of 2^n possible binary strings at any given time. The memory configuration is always definite: each bit is either 0 or 1, and the register encodes a single classical value.

Quantum computation offers a fundamentally different form of information storage. Because qubits can exist in superposition, a quantum register of n qubits can encode *all* 2^n classical configurations simultaneously, within a single quantum state. These configurations are not stored as separate classical values, but as probability amplitudes associated with the basis states of the quantum register. This property can be interpreted as a form of enhanced memory capacity, made possible by quantum superposition.

To make this idea concrete, let us consider a quantum register composed of three qubits. Classically, three bits can represent one of the $2^3 = 8$ possible binary strings:

$$000,\ 001,\ 010,\ 011,\ 100,\ 101,\ 110,\ 111.$$

At any moment, a classical three-bit register stores exactly one of these strings.

In contrast, a three-qubit quantum register can be prepared in a superposition of all eight classical configurations. A general three-qubit quantum state can be written as

$$|\psi\rangle = \alpha_0 |000\rangle + \alpha_1 |001\rangle + \alpha_2 |010\rangle + \alpha_3 |011\rangle + \alpha_4 |100\rangle + \alpha_5 |101\rangle + \alpha_6 |110\rangle + \alpha_7 |111\rangle \ , \tag{2.3}$$

where the complex coefficients α_i satisfy the normalization condition

$$\sum_{i=0}^{7} |\alpha_i|^2 = 1.$$

Each basis state $|ijk\rangle$ corresponds to one classical three-bit string, but the quantum register encodes all eight possibilities at once through their associated probability amplitudes. In this sense, a system of three qubits provides access to an exponentially large configuration space using only three physical units of information.

An important point is that when qubits are in a superposition but not entangled, the state of each qubit is independent of the others. Each qubit can evolve on its own, changing its amplitudes and relative phase through single-qubit operations, without affecting the state of the other qubits. In other words, in this scenario each operation is local. For example, applying a rotation to the first qubit changes the amplitudes of all basis states that differ in the first bit, while the second qubit remains unaffected.

The exponential scaling of possible memory states with qubit number constitutes one key resource that quantum algorithms are designed to harness. Complementing this computational advantage is entanglement, a quintessential feature of quantum mechanics that fundamentally distinguishes quantum from classical information processing.

Entanglement

The full potential of quantum computing is realized through the introduction of entanglement. When a two-qubit operation is applied, the quantum states become entangled, rendering their subsequent evolution correlated rather than independent. As an illustrative example, we present one of the Bell states [18], [79]:

$$|\beta_0\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \tag{2.4}$$

This is a maximally *entangled state*: the two qubits are now correlated, and measuring one immediately determines the state of the other. If we measure one of the two qubits in $|0\rangle$ we are sure to measure the other in the same state. In the same way, if we measure one of the qubits in $|1\rangle$ the other qubit will surely be in $|1\rangle$. The creation of entanglement happens only when a two- or multi-qubit operation is applied on the system

Entanglement is a uniquely quantum resource that enables correlations and interference patterns across multiple qubits, which classical computation cannot reproduce. Together with superposition, it forms the foundation of quantum information processing and underlies the exponential growth of the quantum state space. At the same time, these same principles impose fundamental limitations on how quantum information can be manipulated.

No-cloning Theorem

A particularly important example is the *no-cloning theorem* [18]. In classical computation, information can be copied freely: a classical bit string can be duplicated without restriction, allowing data to be stored, transmitted, and reused arbitrarily. Quantum information behaves very differently. There exists no physical process that can perfectly copy an unknown quantum state.

To see why a universal quantum cloning machine is impossible, consider a qubit in an arbitrary state

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,.$$

Suppose there exists a unitary operation U that can clone an arbitrary state by acting on the qubit and an ancillary state $|0\rangle$:

$$U\,|\psi\rangle\,|0\rangle = |\psi\rangle\,|\psi\rangle\,.$$

Now, for U to work as a universal cloner, it must also correctly clone the basis states individually:

$$U\,|0\rangle\,|0\rangle = |0\rangle\,|0\rangle\,, \quad U\,|1\rangle\,|0\rangle = |1\rangle\,|1\rangle\,.$$

By linearity of quantum mechanics, the action of U on a superposition $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ is

$$U(\alpha\,|0\rangle + \beta\,|1\rangle)\,|0\rangle = \alpha U\,|0\rangle\,|0\rangle + \beta U\,|1\rangle\,|0\rangle = \alpha\,|0\rangle\,|0\rangle + \beta\,|1\rangle\,|1\rangle\,.$$

However, this is not equal to the perfectly cloned state

$$|\psi\rangle\,|\psi\rangle = (\alpha\,|0\rangle + \beta\,|1\rangle) \otimes (\alpha\,|0\rangle + \beta\,|1\rangle) = \alpha^2\,|0\rangle\,|0\rangle + \alpha\beta\,|0\rangle\,|1\rangle + \beta\alpha\,|1\rangle\,|0\rangle + \beta^2\,|1\rangle\,|1\rangle\,.$$

Since the two expressions are generally different for arbitrary α and β, there cannot exist a universal unitary U that clones all quantum states. This illustrates the fundamental impossibility of a universal quantum cloning machine.

The no-cloning theorem highlights a crucial aspect of quantum information: superposition states cannot be treated as classical data objects. They cannot be copied and distributed to create correlations. Instead, correlations between qubits must be generated dynamically through multi-qubit operations, giving rise to entanglement. In this sense, entanglement is not obtained by copying quantum information but by coherently coupling qubits through controlled interactions.

This limitation has deep consequences for quantum computation. This explains why quantum algorithms must rely on interference and entanglement rather than redundancy and copying, and why quantum operations must be carefully designed to manipulate the global quantum state as a whole. Superposition, entanglement, and the no-cloning theorem define both the power and the constraints of quantum information processing.

In summary, single qubits form the basic building blocks, each capable of independent evolution in superposition. Two-qubit (or more generally, multi-qubit) operations can create correlations and entanglement, producing complex quantum states. This sets the stage for discussing quantum operations, which, in analogy to classical logic gates, manipulate qubits and quantum registers to perform computation.

2.3 Quantum Operations and the Circuit Formalism

Following the analogy with classical computation and Turing machines, once the information has been prepared in an initial state, the next essential ingredient is the set of operations that act on it. In quantum computation, these operations evolve quantum states while respecting the principles of quantum mechanics.

In quantum mechanics, the state of a closed system is described by a normalized vector, and its evolution must be both linear and reversible [18]. By "closed system" we mean a system that does not interact with the environment; for example, no measurement or external disturbance occurs.

As long as the system remains closed, its evolution is described by *unitary transformations*. A unitary operator U has the property

$$U^\dagger U = I,$$

which guarantees that the total probability is preserved during the evolution. In simpler terms, unitary transformations are like rotations in a high-dimensional space: they change the direction of the quantum state vector but never its length, ensuring that the state remains properly normalized.

In quantum computation, unitary operations play the role of computational instructions: they coherently manipulate the quantum register, transforming it step by step from an initial state to a desired final state, without losing any information along the way.

Quantum operations can act on different parts of the register. *Single-qubit operations* modify the state of an individual qubit, changing amplitudes and relative phase without directly affecting other qubits. It is important to point out that single-qubit unitaries are local operators, acting as $U \otimes I$, and directly affect only the targeted qubit. However, when the qubit is entangled with others, a local unitary changes the joint state and can modify correlations, while leaving the reduced state of the other qubits and the entanglement spectrum unchanged. Non-unitary local operations, such as measurements, behave differently and can alter both the reduced states and the entanglement.

Multi-qubit operations act on two or more qubits simultaneously and are responsible for creating correlations and entanglement. Both types of operation are essential for exploiting the full power of quantum computation.

There exist different approaches to describe quantum computation. In this Thesis, we focus on the *digital approach*, also known as the *quantum circuit formalism*. In this framework, the evolution of the quantum system is represented as a sequence of operations called *quantum gates*. There is a direct analogy with classical circuits: the wires correspond to qubits, and the gates correspond to unitary operations applied to one or more qubits. Complex algorithms are represented by connecting gates in sequence, forming a *quantum circuit*.

It is important to stress that this is only a formalism to describe the evolution of the system. While a classical circuit is built on a physical chip using physical gates, a quantum gate does not correspond to a physical device like that in classical electronics; rather, it represents an operation that is performed on the quantum state. It is a pure formalism; the actual physical system is a quantum state, on which a series of operations are done in sequence in order to obtain the final state. Circuit representation is a convenient and intuitive way to describe these concepts, making it easier to design, analyze, and communicate quantum algorithms.

Quantum circuits therefore provide a clear and visual way to represent the preparation of quantum states, their evolution through unitary operations, and the final measurement. This formalism will serve as the primary framework for discussing quantum algorithms throughout this Thesis.

Quantum Registers

Before introducing specific quantum gates, it is useful to recall the circuit representation and the notion of a quantum register. In the circuit formalism, each qubit in the quantum register is represented by a horizontal line, or *wire*, which tracks the state of that qubit throughout the computation. Thus, a quantum register of n qubits is represented by n parallel wires, each corresponding to a single qubit.

Quantum gates correspond to unitary operations that act on one or more of these qubits. A single-qubit gate acts on a single wire, modifying the state of that qubit without directly affecting the others, while multi-qubit gates act on two or more wires, creating correlations and entanglement. The sequence of gates along the wires defines the evolution of the quantum register, from its initial prepared state to the final state that is measured at the end of the computation.

For computational purposes, it is convenient to represent the basis states $|0\rangle$ and $|1\rangle$

as column vectors:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{2.5}$$

These two states form what is called the *computational basis*. In this representation, single-qubit gates correspond to 2×2 unitary matrices acting on these vectors. This provides a concrete and consistent framework for describing quantum operations in the circuit formalism.

2.3.1 Single-Qubit Gates

Single-qubit gates are the simplest quantum operations as they act on one qubit at a time, changing its state. Despite their simplicity, they form the building blocks for more complex operations and algorithms.

A natural starting point for single-qubit gates are the *Pauli matrices*, which form a complete basis for 2×2 Hermitian operators. They are fundamental in quantum computation because any single-qubit unitary operation can be expressed in terms of these matrices (up to a global phase). The Pauli matrices are:

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

Each of these matrices corresponds to a basic type of qubit transformation:

Pauli-X gate.

The X gate is the quantum analog of the classical NOT gate. It flips the state of a qubit:

$$X|0\rangle = |1\rangle, \qquad X|1\rangle = |0\rangle.$$

Pauli-Y gate.

The Y gate combines bit-flip and phase-flip operations. Its action is:

$$Y|0\rangle = i|1\rangle, \qquad Y|1\rangle = -i|0\rangle.$$

Pauli-Z gate.

The Z gate leaves the computational basis states unchanged but flips the phase of the $|1\rangle$ component:

$$Z|0\rangle = |0\rangle, \qquad Z|1\rangle = -|1\rangle.$$

Hadamard gate.

The Hadamard gate is widely used to create superposition:

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

Its matrix form is:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

It is crucial for generating superposition, enabling quantum parallelism in algorithms.

Parametrized single-qubit rotations.

In addition to discrete gates such as X, Y, Z, and H, it is often useful to describe single-qubit operations that can be continuously varied. These are called *parametrized rotations* and allow for precise control over the state of a qubit.

A rotation around a given axis $\hat{n} \in \{X, Y, Z\}$ by an angle θ is defined as the unitary

$$R_{\hat{n}}(\theta) = e^{-i\frac{\theta}{2}\hat{n}}.$$

In matrix form, the standard rotations around the principal axes are as follows:

$$R_X(\theta) = \cos\frac{\theta}{2}\mathbb{I} - i\sin\frac{\theta}{2}X = \begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix},$$

$$R_Y(\theta) = \cos\frac{\theta}{2}\mathbb{I} - i\sin\frac{\theta}{2}Y = \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix},$$

$$R_Z(\theta) = \cos\frac{\theta}{2}\mathbb{I} - i\sin\frac{\theta}{2}Z = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix}.$$

These rotations generalize the discrete Pauli gates: for example, $R_X(\pi) = X$, $R_Y(\pi) = Y$, and $R_Z(\pi) = Z$. By tuning the rotation angle θ, the amplitudes and phases of the qubit can be continuously changed.

Moreover, it is possible to express any single-qubit unitary as a combination of three rotations around two axes (commonly Z and Y):

$$U = e^{i\alpha}R_Z(\phi)R_Y(\theta)R_Z(\lambda),$$

with real parameters $\alpha, \phi, \theta, \lambda$. This decomposition shows that parametrized rotations provide a complete and universal description of single-qubit operations, making them a convenient tool for both theoretical analysis and hardware implementation.

2.3.2 Two-Qubit and Multi-Qubit Gates

While single-qubit gates control individual qubits, multi-qubit gates act on two or more qubits simultaneously. These gates are essential for creating correlations between qubits, generating entanglement, and implementing complex quantum algorithms.

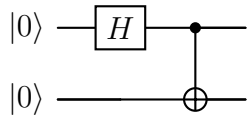


Figure 2.1: Quantum circuit state used for the production of the Bell's state in Eq. (2.4). It reads from left to right, first an Hadamard gate is applied to the top qubit, what follows is a CNOT gate controlled by the top qubit and acting on the bottom one.

The CNOT gate

The Controlled-NOT (CNOT) or Controlled-X (CX) gate is the most common two-qubit gate. It flips the state of a target qubit if and only if the control qubit is in the state $|1\rangle$. In the computational basis $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$, its matrix representation is

$$\mathrm{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

An example of the CZ in a quantum circuit is shown in Fig. 2.2.

To illustrate its action, consider a two-qubit system as in Fig. 2.1. After the Hadamard gate the system is in the state

$$|\psi\rangle = |+\rangle \otimes |0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle . \tag{2.6}$$

Expanding the tensor product gives the following.

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle \otimes |0\rangle + |1\rangle \otimes |0\rangle) = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle).$$

Now we apply the CNOT gate with the first qubit as the control and the second qubit as the target:

$$\mathrm{CNOT}\,|00\rangle = |00\rangle , \quad \mathrm{CNOT}\,|10\rangle = |11\rangle .$$

Therefore, the resulting state is

$$\mathrm{CNOT}(|\psi\rangle) = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle),$$

which is a maximally entangled Bell state, as in Eq. (2.4). This calculation illustrates two key features of two-qubit gates: their ability to generate entanglement from initially separable states, and their capacity to produce correlations that cannot be expressed as tensor products of individual qubit states.

The Controlled-Z (CZ) gate.

The CZ gate applies a phase flip to the target qubit if the control qubit is in state $|1\rangle$:

$$\mathrm{CZ}\,|00\rangle = |00\rangle , \quad \mathrm{CZ}\,|01\rangle = |01\rangle , \quad \mathrm{CZ}\,|10\rangle = |10\rangle , \quad \mathrm{CZ}\,|11\rangle = -\,|11\rangle .$$

It can be implemented using a CNOT gate and two Hadamard gates on the target qubit:

$$\mathrm{CZ}_{c,t} = (\mathbb{I} \otimes H)\,\mathrm{CNOT}_{c,t}\,(\mathbb{I} \otimes H),$$

where c is the control and t is the target. This decomposition shows the flexibility of quantum circuits in constructing different two-qubit gates from simpler operations. An example of the CZ in a quantum circuit is shown in Fig. 2.2.

Multi-Controlled Gates

Beyond two-qubit gates, quantum circuits often require operations controlled by multiple qubits. These are called *multi-controlled gates*. A general multi-controlled gate applies a unitary operation U to a target qubit if and only if all control qubits are in specified states.

For example, the Toffoli gate (CCNOT) is a three-qubit gate with two control qubits and one target qubit. It flips the target qubit only if both control qubits are in state $|1\rangle$:

$$\text{Toffoli}\ |a, b, c\rangle = |a, b, c \oplus (a \cdot b)\rangle\,,$$

where $a, b, c \in \{0, 1\}$ and $\oplus$ denotes addition modulo 2.

In general, each control qubit can be set to trigger the operation when in state $|1\rangle$ or $|0\rangle$. The notation is typically:

- $\bullet$ in a control qubit denotes the control by 1 (the gate is activated if the qubit $= |1\rangle$).
- $\circ$ in a control qubit denotes the control by 0 (the gate activates if the qubit $= |0\rangle$).

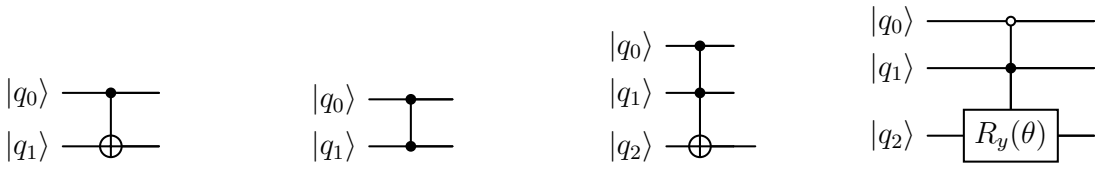
Examples the Toffoli and Multi-Controlled-Gate (MCG) in a quantum circuit are shown in Fig. 2.2.

For example, a controlled gate with two qubits where the first is control-by-1 and the second is control-by-0 applies the operation U to the target only if the first qubit is $|1\rangle$ and the second qubit is $|0\rangle$.

Multi-controlled gates can have an arbitrary number of controls. Any n-qubit multi-controlled unitary can be decomposed into single-qubit and two-qubit gates (like CNOTs), possibly using ancillary qubits to reduce the number of operations. This decomposition is crucial in practice, because most quantum hardware natively supports only single- and two-qubit gates.

2.3.3 Circuit Cost, Gate Decomposition, and Ancilla Qubits

After introducing single- and multi-qubit gates, it is important to discuss how these gates are implemented in practice, how complex operations can be decomposed, and how this affects the cost and feasibility of quantum circuits.



(a) CNOT gate (b) CZ gate (c) Toffoli (CCNOT) (d) Multi-controlled R_y.

Figure 2.2: Examples of two- and three-qubit gates in the circuit model. (a) Controlled-NOT (CNOT) gate. (b) Controlled-Z (CZ) gate. (c) Toffoli gate (CCNOT). (d) Multi-controlled R_y.The first control is on $|0\rangle$, the second on $|1\rangle$

Gate decomposition and universality. Multi-qubit operations, such as the Toffoli Controlled-Controlled-NOT(CCNOT) gate or more general multi-controlled unitaries, are typically not available as elementary operations on quantum hardware. Instead, they are implemented by decomposing them into sequences of single- and two-qubit gates [80]. In practice, a quantum processor supports only a limited collection of elementary instructions that act directly on one or two qubits at a time; however, these are sufficient to construct arbitrary quantum operations.

As an example, in superconducting quantum processors, only a restricted set of control pulses can be applied to the physical system. These pulses define the *native gate set*, which usually consists of single-qubit rotations and a two-qubit entangling gate such as the controlled-Z (CZ), acting on the computational basis states $|0\rangle$ and $|1\rangle$. More complex operations, including Toffoli or multi-controlled gates, must therefore be expressed as sequences of these native gates.

This decomposition plays a central role in assessing the cost of a quantum circuit. By rewriting an algorithm in terms of the native gate set, one can estimate the total number of gates, the circuit depth, and the possible need for ancilla qubits. From a software viewpoint, this process is analogous to a classical compiler: the high-level algorithm must be *transpiled* into operations supported by the hardware. In addition, the qubit topology must be taken into account. Many architectures do not allow direct two-qubit gates between arbitrary pairs of qubits, so additional gates or routing operations are often required to move quantum information, which can further increase the circuit depth and overall resource requirements.

Ancilla qubits. Ancilla qubits are auxiliary qubits used to simplify the decomposition of complex operations or to temporarily store intermediate results. They do not carry algorithmic input or output but can make circuits more efficient in terms of gate count or depth, at the price of additional qubits. They can be considered as extra qubit used in order to decompose complex multi-qubits gates in a more efficient manner.

Circuit cost. The complexity of a quantum circuit can be quantified in terms of several complementary metrics. One commonly used measure is the *gate count*, defined as the total number of single- and multi-qubit gates applied. Another important quantity is the *circuit depth*, corresponding to the length of the longest sequence of mutually dependent gates; this is often interpreted as a measure of time complexity, as it is directly related to the execution time of the circuit. Finally, the *circuit width*, or ancilla usage, quantifies the

number of auxiliary qubits required and can be regarded as a measure of space complexity, reflecting the amount of quantum resources needed.

Circuit cost is closely related to the physical limitations of current devices, as longer circuits or circuits with many multi-qubit operations are more susceptible to errors. Careful analysis of circuit resources is therefore essential for evaluating the feasibility of a quantum algorithm on a given device.

Example: Toffoli decomposition. Consider a Toffoli gate with control qubits c_1, c_2 and target t. One standard decomposition uses single-qubit rotations on the target and six CNOT gates between the control and target qubits. If an ancilla qubit a is available, it can be used to reduce either the total number of CNOTs or the circuit depth, depending on the chosen decomposition [18], [80]. This example illustrates that multi-qubit gates can always be expressed in terms of single- and two-qubit operations, which allows the estimation of gate count, depth, and resource requirements for larger circuits.

Physical relevance. The discussion of gate decomposition and circuit cost highlights an important connection between the abstract description of quantum algorithms and their physical realization. While single-qubit operations act locally on the computational basis states $|0\rangle$ and $|1\rangle$ and can usually be implemented with relatively high precision, multi-qubit gates require coordinated control over several qubits and their interactions. As a result, these operations tend to be slower and more sensitive to imperfections.

As quantum circuits grow in depth and complexity, errors introduced during gate execution and the unavoidable interaction of the system with its environment accumulate. This leads to decoherence, progressively degrading the quantum state, and limiting the length of computations that can be reliably performed. Ancilla qubits can be employed to optimize gate decompositions and reduce circuit depth, but at the cost of increasing the total number of physical qubits involved.

For these reasons, the cost of a quantum circuit, measured in terms of gate count, depth, and ancillary resources, is not just a theoretical notion, but a key factor in determining the feasibility of quantum algorithms. This naturally motivates a more detailed discussion of noise, errors, and decoherence, which play a central role in both near-term quantum devices and the development of fault-tolerant quantum computation.

2.4 Noise and Decoherence

In an ideal quantum computer, the evolution of a quantum register is fully described by unitary operations acting on well-isolated qubits. In practice, however, a quantum system can never be perfectly isolated from its environment. Uncontrolled interactions with external degrees of freedom, as well as imperfections in the implementation of quantum gates, introduce deviations from the intended evolution. These effects are collectively referred to as *noise*.

Noise manifests itself in two closely related ways: through *errors* in the applied operations and through *decoherence*, the gradual loss of quantum coherence in the system [18], [81]. Decoherence can be understood as the process by which a quantum state loses its ability to exhibit superposition and interference, effectively acting more like a classi-

cal probabilistic mixture. Since quantum computation relies crucially on superposition and entanglement, decoherence represents one of the main obstacles to scalable quantum computation.

A useful way to characterize decoherence is in terms of characteristic time scales. The *relaxation time* (T_1) describes the tendency of a qubit to decay from the excited state $|1\rangle$ to the ground state $|0\rangle$, while the *dephasing time* (T_2) quantifies the loss of relative phase coherence between $|0\rangle$ and $|1\rangle$. Both processes limit the time window during which reliable quantum operations can be performed.

Typical values of T_1 and T_2 depend on the qubit technology. For superconducting qubits, T_1 and T_2 are often in the range of microseconds [82]; while trapped-ion qubits can reach much longer coherence times, in the order of seconds [83].

Noise affects quantum circuits cumulatively. Each gate introduces a small error, and as the circuit depth increases, these errors add up, eventually overwhelming the useful quantum signal. For this reason, the cost of a circuit—measured in terms of depth and number of multi-qubit gates—is directly linked to its vulnerability to noise. In particular, entangling operations are typically more sensitive to noise than single-qubit gates, making them a critical bottleneck in many quantum algorithms.

2.4.1 NISQ and Fault-Tolerant Quantum Computing

The unavoidable presence of noise and decoherence fundamentally shapes how quantum computers can be used in practice. Depending on how errors are treated, it is common to distinguish between two computational regimes: *Noisy Intermediate-Scale Quantum* (NISQ) [20] computing and *fault-tolerant* quantum computing [81].

NISQ devices operate with a moderate number of qubits and without full quantum error correction. In this regime, quantum circuits must be executed within the limited coherence time of the hardware, since errors accumulate as the circuit depth increases. As a consequence, only relatively shallow circuits can be reliably run, and the cost of multi-qubit gates becomes a critical constraint. Algorithms designed for NISQ hardware therefore emphasize low-depth constructions, careful gate optimization, and often hybrid quantum–classical approaches that reduce the required quantum resources.

Fault-tolerant quantum computing represents a different paradigm in which errors are actively detected and corrected throughout the computation. Logical qubits are encoded into many physical qubits using quantum error-correcting codes, allowing reliable operations even in the presence of noise, provided that physical error rates remain below a certain threshold. Although this approach enables arbitrarily long and accurate computations in principle, it introduces substantial overhead in terms of qubit count, gate complexity, and circuit depth.

Understanding the distinction between theoretical and NISQ-focused quantum algorithms is crucial for assessing their relevance and practical requirements. Some algorithms are primarily of theoretical interest, assuming access to fault-tolerant resources, while others are designed specifically to work within the constraints of NISQ devices. Recognizing where an algorithm lies on this spectrum helps framing the discussion of the algorithms presented in this Thesis.

Specifically, we begin with a NISQ-oriented quantum algorithm in Ch. 4, tailored for current intermediate-scale devices, to highlight the practical challenges such algorithms

face. Later, we turn to algorithms that assume more fault-tolerant resources, including one based on the linearization of the SP equations in Ch. 5 and the algorithm for neighbor search in Ch. 6.

2.5 Measurements in Quantum Computing

In the classical Turing machine framework, computation ends when the machine produces an output on its tape: the result is read directly as a string of bits. Similarly, in quantum computing, after the quantum register has been initialized and evolved through a sequence of quantum gates, the final step is to extract classical information by performing *measurements*.

Measurement in quantum mechanics is fundamentally different from reading a classical bit. Although classical bits have definite values at all times, qubits exist in superpositions of $|0\rangle$ and $|1\rangle$. When a qubit is measured on the computational basis, its state $|\psi\rangle = \alpha|0\rangle+\beta|1\rangle$ collapses probabilistically to $|0\rangle$ with probability $|\alpha|^2$ and to $|1\rangle$ with probability $|\beta|^2$. Importantly, this collapse is destructive: once measured, the qubit can no longer retain the original superposition. Consequently, recovering information about the full quantum state requires multiple repetitions of the same circuit.

Consider, for example, the maximally entangled two-qubit Bell state:

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle).$$

If we measure the first qubit on the computational basis, we obtain $|0\rangle$ with probability $1/2$ and $|1\rangle$ with probability $1/2$. After the first measurement, the overall state collapses accordingly: if we observe $|0\rangle$, the second qubit is immediately projected onto $|0\rangle$; if we observe $|1\rangle$, the second qubit collapses to $|1\rangle$. To estimate the probabilities and correlations in the Bell state, this procedure must be repeated many times, each time preparing the same initial state and executing the same sequence of gates. The distribution of outcomes over these repetitions allows us to reconstruct the statistical properties of the state.

This probabilistic approach introduces the concept of *variance* in measurement outcomes. For N independent repetitions (or shots) of the circuit, the standard deviation of the measured probabilities scales as $1/\sqrt{N}$. Achieving the desired precision therefore requires a sufficient number of repetitions, which must be carefully estimated when designing a quantum algorithm to ensure that the potential quantum advantage is preserved. An explicit example and a more detailed discussion of this aspect is provided in Chapter 5.

In practice, measurements are typically performed on the computational basis, but more general measurements can be realized by applying additional gates to rotate the basis before readout. The results are then recorded as classical bit strings that can be further processed by classical algorithms to extract meaningful information. Measurement, together with initialization and quantum gates, completes the full computational cycle, providing a direct bridge between the abstract quantum evolution and the observable results, and forming the quantum analog of the input-output process in a classical Turing machine.

2.6 Summary

This chapter provided a comprehensive introduction to the foundations of quantum computation, establishing both the conceptual and practical framework necessary to understand quantum algorithms.

We began by recalling classical computation, emphasizing Turing machines and Boolean circuits, where information is encoded in bits, processed through logical operations, and read out as definite outputs. Motivated by the physical limitations of classical devices and the potential for new computational paradigms, we introduced quantum computation, where the fundamental unit of information is the qubit.

Qubits can exist in superpositions of classical states, allowing a quantum register to represent exponentially many configurations simultaneously. When qubits are entangled, correlations arise that have no classical analog, enabling phenomena such as quantum parallelism and interference, which underpin the power of quantum algorithms.

Quantum computation proceeds through sequences of unitary operations, or quantum gates. Single-qubit gates manipulate individual qubits, while multi-qubit gates, such as CNOT and Toffoli, create entanglement and correlations. Complex operations are decomposed into sequences of gates from the device's native set, which provides a practical way to estimate circuit cost, including gate count, depth, and ancillary qubits. These quantities are closely tied to susceptibility to noise and decoherence, which limit the coherence time and accuracy of computations.

Noise and decoherence arise from environmental interactions and imperfections in gate implementation, motivating the distinction between NISQ devices, which operate with shallow circuits and partial error mitigation, and fault-tolerant quantum computers, which employ error correction to perform arbitrarily long computations at the cost of additional qubits and operations.

Finally, measurement collapses qubits into classical outcomes, completing the quantum computation cycle. Due to the probabilistic nature of quantum measurement, multiple repetitions are necessary to extract reliable information, with statistical variance decreasing as the number of runs increases. Measurement thus bridges the quantum evolution of the system to the classical information that algorithms ultimately seek to compute.

Together, these concepts — qubits, superposition, entanglement, gates, circuit decomposition, noise, and measurement — provide a unified picture of how information is represented, processed, and extracted in a quantum computer. This framework sets the stage for studying quantum algorithms, analyzing their resource requirements, and assessing their feasibility on both current NISQ devices and future fault-tolerant architectures. Currently, only NISQ devices exist, with limited qubits and susceptibility to noise, while fully fault-tolerant quantum computers remain a long-term goal. This distinction helps determine which algorithms can be implemented today and which are relevant for future, error-corrected systems.

For example, in Ch. 4 we introduce a variational algorithm to solve SP, where both the quantum states and the circuits used in the evolution are constructed from the elementary gates discussed earlier. Similarly, in Ch. 6 we present a quantum algorithm for neighbor search, including a detailed analysis of how multi-controlled gates can be decomposed into elementary gates. This allows us to evaluate the algorithm's resource requirements and feasibility on both NISQ devices and future fault-tolerant architectures, linking directly

to the distinctions and considerations introduced in the previous sections.

Chapter 3

Comparison of Vlasov Solvers

In this chapter, we compare three numerical approaches for solving the VP system, with particular emphasis on the SP method, which forms the basis for the development of a quantum algorithm. The main objective is to assess under which conditions SP is able to recover the VP dynamics. In addition, this framework allows us to examine how the N-body solution compares with the direct integration of VP. To this end, we implement two Eulerian VP solvers, a PM N-body solver, and a spectral solver for the SP system.

The chapter is organized into three sections. Section 3.1 presents a comprehensive description of the numerical implementations, highlighting the specific design choices and innovations introduced for this work. We detail the spectral and semi-Lagrangian VP solvers. While a general overview of the PM method was provided earlier, here we describe our implementation in depth, including practical strategies and numerical techniques. Notably, we employ a Cloud-in-Cell interpolation scheme combined with leapfrog integration. The pseudo-spectral SP method is also discussed, with the introduction of a quantity analogous to the Reynolds number in VP, emerging naturally from the Schrödinger framework.

Section 3.2 presents a series of numerical tests for the VP solvers, highlighting the challenges posed by cold dark matter initial conditions. We trace the origin of these challenges and demonstrate that CDM configurations generate shocks and stiffness in the velocity field. Through careful analysis, we identify the source of the problem and propose a viable solution, which is supported by an extensive set of numerical tests.

Section 3.3 provides a comparative analysis of the solvers using two representative scenarios: a one-dimensional sinusoidal perturbation and a two-dimensional Gaussian collapse. These examples show that, after appropriate coarse graining, the SP method successfully reproduces the key features of VP evolution. This outcome not only validates the Schrödinger approach in the classical context but also establishes a clear pathway for its application in quantum computing frameworks aimed at simulating cold dark matter.

By the conclusion of this chapter it will be clear if SP can be used to simulate DM distributions instead of VP, or as a viable alternative.

3.1 Numerical solvers

In this section, we describe the numerical solvers used in this Thesis to study the Vlasov-Poisson system. We explain how the algorithms are implemented and discuss the practical

considerations needed to achieve accurate and efficient simulations. The goal is to provide a clear picture of the methods that will be applied in the analysis that follows.

3.1.1 Eulerian Vlasov-Poisson Solver

An Eulerian VP solvers evolve the phase-space distribution function on a smooth grid, avoiding the sampling noise inherent in particle-based methods and allowing accurate resolution of the fine filamentary structures that arise from collisionless stretching and mixing. Unlike N-body approaches, they are not affected by spurious collisions or artificial relaxation. The main limitation of this approach is the computational cost, which grows rapidly due to the need to evolve the system in a six-dimensional phase space.

To integrate Eqs. (1.7) and (1.8), we adopt the splitting scheme of Cheng & Knorr [58], which separates the problem into two advections: one in velocity space and one in physical space, described by Eqs. (1.21) and (1.20). Each of these steps can be further reduced to a sequence of one-dimensional advections along the respective directions. This decomposition provides a natural framework for numerical implementation.

For temporal integration, we employ a second-order leapfrog scheme, which is widely used in the literature for VP solvers. The integration is initialized with the velocity advection step, a choice that is purely conventional and does not affect the overall accuracy. Following the procedure outlined in [61], the three-dimensional integration can be written as

$$\begin{aligned} f(\mathbf{x}, \mathbf{v}, t + \Delta t) = &\left[\mathcal{D}_{v_x}(\Delta t/2)\, \mathcal{D}_{v_y}(\Delta t/2)\, \mathcal{D}_{v_z}(\Delta t/2)\right] \\ &\times \left[\mathcal{D}_x(\Delta t)\, \mathcal{D}_y(\Delta t)\, \mathcal{D}_z(\Delta t)\right] \\ &\times \left[\mathcal{D}_{v_x}(\Delta t/2)\, \mathcal{D}_{v_y}(\Delta t/2)\, \mathcal{D}_{v_z}(\Delta t/2)\right], \end{aligned} \tag{3.1}$$

where $\mathcal{D}_i(t)$ denotes the advection operator along the dimension i over a time t. This approach resembles a kick-drift-kick (KDK) time-integration scheme and reduces the problem to a series of one-dimensional advections, which greatly simplifies implementation and clarifies the structure of the numerical method.

For clarity and to streamline the discussion of the implementation, we focus on the one-dimensional case in the following sections. In this case, the VP equations reduce to

$$\frac{\partial f(x,v,t)}{\partial t} + v\,\frac{\partial f(x,v,t)}{\partial x} - \frac{\partial \phi}{\partial x}\,\frac{\partial f(x,v,t)}{\partial v} = 0, \tag{3.2}$$

$$\nabla^2 \phi(x,t) = \alpha\,(\rho^* - 1), \tag{3.3}$$

with $\alpha = 4\pi G\bar{\rho}$ and $\rho^* = \rho(x,t)/\bar{\rho}$, where $\bar{\rho}$ is the reference density of the universe. In this one-dimensional setting, the advection steps correspond to simple one-dimensional equations :

$$\frac{\partial f(x,v,t)}{\partial t} + v\,\frac{\partial f(x,v,t)}{\partial x} = 0\,, \tag{3.4}$$

$$\frac{\partial f(x,v,t)}{\partial t} - \frac{\partial \phi}{\partial x}\,\frac{\partial f(x,v,t)}{\partial v} = 0\,. \tag{3.5}$$

Conservation laws

Before examining the details of the implementation of VP solvers, we must first establish criteria for evaluating their performance. A good numerical integrator should accurately reproduce the evolution of the distribution function, which is not known a priori. Since we lack an analytical benchmark, we need alternative metrics to assess the quality of the solution. Is it proven, however, that the VP system verifies a certain set of conservation properties. We will study the behavior of such proprieties to verify which numerical integration method is best suited in our scenario. The properties we want to check are the following:

- Maximum principle This property follows from the fact that the distribution function f is constant along the phase-space characteristics of VP. Since no sources or sinks are present, f cannot create new extrema during its evolution, and its values remain bounded by those of the initial condition.

$$0 \leq f(\mathbf{x}, \mathbf{v}, t) \leq \max_{(\mathbf{x},\mathbf{v})} \{f_0(\mathbf{x}, \mathbf{v})\}$$

- Conservation of L^p norm, $p \in [1, \infty)$ The flow induced by VP is measure-preserving in phase space. As a consequence, integrals of any power of the distribution function over phase space are invariant in time, which implies the conservation of all L^p norms for $1 \leq p < \infty$.

$$||f_t||_{L^p} = \int d\mathbf{x}\, d\mathbf{v}\, [f(\mathbf{x}, \mathbf{v}, t)]^p \ . \tag{3.6}$$

- Total mass conservation. For any volume V of phase space

$$M = \int_V d\mathbf{x}\, d\mathbf{v} f(\mathbf{x}, \mathbf{v}, t)\,. \tag{3.7}$$

- Momentum conservation

$$P = \int_V d\mathbf{x}\, d\mathbf{v} f(\mathbf{x}, \mathbf{v}, t)\mathbf{v}\,. \tag{3.8}$$

- Energy conservation

$$E = \frac{1}{2} \int_V d\mathbf{x}\, d\mathbf{v} f(\mathbf{x}, \mathbf{v}, t) \left(\mathbf{v}^2 + \phi(\boldsymbol{x}, t)\right) \ , \tag{3.9}$$

  with $\phi(\mathbf{x}, t)$ from Eq. (1.8).

Numerical Scheme

Following a grid-based approach, the spatial domain of length L is divided into N grid points with periodic boundary conditions, while the velocity domain is restricted to the dynamic range $[-V_{\max}, V_{\max}]$ and discretized into M grid points. From a theoretical

standpoint, all velocities should be allowed, which would lead to an infinite domain—something that is clearly unfeasible in practice. Fortunately, in realistic applications, the velocity domain is naturally bounded due to the attractive nature of the problem. This allows the use of periodic boundary conditions in velocity, provided that $|V_{\max}|$ is chosen to be larger than the maximum velocity reached by the particles in the physical system.

The discrete representation of $f(x, v, t)$ now takes the form of a matrix $M \times N$. Consequently, Eq. (3.5) gives us M advection equations for position, with each vector consisting of N components

$$\frac{\partial f}{\partial t}(x_i, v_j, t) = -v_j \frac{\partial f}{\partial x}(x_i, v_j, t)\,, \tag{3.10}$$

and Eq. (3.5), N equations of M components for the velocity

$$\frac{\partial f}{\partial t}(x_i, v_j, t) = \frac{\partial \phi}{\partial x}(x_i) \frac{\partial f}{\partial v}(x_i, v_j, t)\,, \tag{3.11}$$

To go from f^n to f^{n+1} we employ a leapfrog scheme, bringing the accuracy to $\mathcal{O}(\Delta t^2)$; by solving Eq. (3.11) for the half time step

$$f^*(x, v) = f^n(x, v - F\Delta t/2) \tag{3.12}$$

then solve Eq. (3.10) for a full time step

$$f^{**}(x, v) = f^*(x - v\Delta t, v)\,. \tag{3.13}$$

With f^{**} we solve the Poisson equation (1.8) and find a new $F = -\partial_x \phi[f^{**}]$. As a final step, we solve Eq. (3.11) for a second half time step.

$$f^{n+1}(x, v) = f^{**}(x, v - F\Delta t/2) \tag{3.14}$$

The two different solvers we will present here will follow this numerical scheme. The only discriminating part between the two will be the approach chosen to solve the advection equations.

Positive Flux and Conservative (FPC)

Among the various grid methods proposed for the solution of VP, the Positive Flux and Conservative (FPC) approach [60] has proven to behave considerably well in different scenarios, preserving positivity of the distribution function and the maximum principle. We have chosen to display only the results obtained with the FPC method based on the results presented in [84] and the fact that it has proven to be the best choice within the pool of semi-Lagrangian method we tested from [85]. We will briefly summarize the FPC approach for a single advection equation.

The FPC approach is part of a bigger family of semi-Lagrangian solvers based on the same flux-balance principle. We start by writing the value $f_i = f(x_i, v, t)$ as the mean value in the cell

$$f_i^n \Delta x = \int_{x_i - \Delta x/2}^{x_i + \Delta x/2} f(x, v, t^n)\, dx\,. \tag{3.15}$$

The flux entering or exiting the cell is consequently defined as

$$\Phi_\pm = \int_{X(t^n, t^{n+1}, x \pm \Delta x/2)}^{x \pm \Delta x/2} f(x, v, t^n)\, dx\,, \tag{3.16}$$

with $X(t^n,\, t^{n+1},\, x \pm \Delta x/2)$ we refer to the value of the variable x if we trace the solution back to the point $(x_i \pm \Delta x/2, t^{n+1})$. In the simple case of Eq. (3.4), using the method of the characteristics we get the following.

$$X(t^n,\, t^{n+1},\, x_i + \Delta x/2) = x_i + \frac{\Delta x}{2} - v_j \Delta t \tag{3.17}$$

Finally, using Eqs. (3.15), (3.16) we can find the expression for the distribution function in the next timestep

$$f_i^{n+1} = f_i + \frac{1}{\Delta x}\left(\Phi_- - \Phi_+\right) . \tag{3.18}$$

The main discriminant between all the flux-based solvers resides in the flux, the interpolation process, and in the regularization techniques employed to ensure positivity.
In the FPC scheme, the interpolation of Φ is based on a essentially non-oscillatory (ENO) [86] reconstruction of the primitive functions of f. This approach eliminates spurious oscillation, but might not behave well under shock effects. Let $C_j = [x_{j-1/2}, x_{j+1/2}]$ be the cell that contains the point x_j. For positive advection velocity, an entering flux Φ_+ passes through $x_{j+1/2}$ following the characteristic of Eq. (3.17). In case of positive advection, i.e., the advection coefficient is > 0 we define the quantity $\alpha_i = x_{j+1/2} - X(t^n, t^{n+1}, x_{i+1/2}) \in [0, \Delta x]$ and the FPC closed-form expression of the flux reads

$$\Phi_{i+1/2} = \frac{1}{\Delta x}\sum_{k=j+1}^{i} f_k \;+\; \alpha_i \Bigg[f_j \;+\; \frac{\varepsilon_j^+}{6}\Big(1 - \frac{\alpha_i}{\Delta x}\Big)\Big(2 - \frac{\alpha_i}{\Delta x}\Big)(f_{j+1} - f_j) \tag{3.19}$$

$$+\; \frac{\varepsilon_j^-}{6}\Big(1 - \frac{\alpha_i}{\Delta x}\Big)\Big(1 + \frac{\alpha_i}{\Delta x}\Big)(f_j - f_{j-1})\Bigg], \tag{3.20}$$

where f_k denotes cell averages, as in Eq. (3.15). In case the advection coefficient v_j in Eq. (3.4) or $-\partial_x \phi$ in Eq. (3.5) is negative, we have $\alpha_i = x_{j-1/2} - X(t^n, t^{n+1}, x_{i+1/2}) \in [-\Delta x, 0]$ and

$$\Phi_{i+1/2} = \frac{1}{\Delta x}\sum_{k=j+1}^{i} f_k \;+\; \alpha_i \Bigg[f_j \;-\; \frac{\varepsilon_j^+}{6}\Big(1 - \frac{\alpha_i}{\Delta x}\Big)\Big(1 + \frac{\alpha_i}{\Delta x}\Big)(f_{j+1} - f_j) \tag{3.21}$$

$$-\; \frac{\varepsilon_j^-}{6}\Big(2 + \frac{\alpha_i}{\Delta x}\Big)\Big(1 + \frac{\alpha_i}{\Delta x}\Big)(f_j - f_{j-1})\Bigg], \tag{3.22}$$

The coefficients ε_j^+ and ε_j^- are slope limiters that preserve positivity and ensure the reconstructed polynomial satisfies $0 \le f_h \le f^\infty$, where $f^\infty = \max_i f_i$. They are defined by

$$\varepsilon_i^+ = \begin{cases} \min\left(1,\; \dfrac{2 f_i}{f_{i+1} - f_i}\right), & f_{i+1} - f_i > 0, \\ \min\left(1,\; \dfrac{-2(\, f^\infty - f_i\,)}{f_{i+1} - f_i}\right), & f_{i+1} - f_i < 0, \end{cases} \tag{3.23}$$

$$\varepsilon_i^- = \begin{cases} \min\left(1,\; \dfrac{2(\, f^\infty - f_i\,)}{f_i - f_{i-1}}\right), & f_i - f_{i-1} > 0, \\ \min\left(1,\; \dfrac{-2 f_i}{f_i - f_{i-1}}\right), & f_i - f_{i-1} < 0. \end{cases} \tag{3.24}$$

These regulators rescale local slopes so that the high-order reconstruction remains consistent, conservative, and strictly positive, thereby preventing the creation of non-physical negative values during phase-space advection.

For the implementation of the VP solver, we combined FPC with the structure given in 3.1.1. In doing so, it is important to remember that in order to advance in time, one must solve the N or M advection equations, as shown in Eqs. (3.10) and (3.11).

Spectral method

The spectral method proposed by [87] solves the one-dimensional Vlasov–Poisson and Vlasov–Maxwell systems using a spectral approach. The distribution function $f_N(t, x, v)$ is approximated by a partial sum of a Fourier series:

$$f_N(t, x, v) = \sum_{k=-N}^{N} \hat{f}_k(t, v) \exp\left(-i\frac{2\pi k x}{L}\right), \tag{3.25}$$

where L is the domain length in the x-direction, and $\hat{f}_k(t, v)$ are the Fourier coefficients. These coefficients are computed from the discrete values of f sampled at the grid points x_j as:

$$\hat{f}_k(t, v) = \sum_j f(t, x_j, v) \exp\left(i\frac{2\pi k x_j}{L}\right), \quad k \in \{-N, \ldots, N\}. \tag{3.26}$$

By means of Eqs. (3.25) and (3.26) the evaluation of the advection term in Eq. (3.12) can be treated a phase shift applied in Fourier space. For each mode k, this shift is expressed as:

$$\hat{f}_k^*(v) = \hat{f}_k(t_n, v) \exp\left(-i\frac{2\pi k}{2L} v \Delta t\right), \tag{3.27}$$

where Δt is the time step. Thus, the algorithm proceeds via a split-step scheme. First, a forward Fast Fourier Transform (FFT) is performed to compute the Fourier coefficients $\hat{f}_k$ from the spatial distribution $f(x, v)$. Next, the phase shift is applied to evolve the solution in Fourier space. Finally, an inverse FFT is used to reconstruct the updated distribution function in the physical space.
The solver proceeds by combining the structure described in Section 3.1.1 with the spectral method. A single time step is performed as follows:

1. Apply a Fourier transform along the v axis. Using the appropriate phase shift, solve Eq. (3.11).

2. Perform a Fourier transform along the x axis. Using a phase shift, solve Eq. (3.10).

3. Return to the phase space (x, v) using a 2D inverse FFT to evaluate the potential. Solve the Poisson equation, Eq. (3.3), in this space.

4. Transform again along the v axis to return to the frequency domain. Apply the phase shift to solve Eq. (3.11).

It is convenient to remain in the frequency domain for most of the computation, returning to (x, v) only to evaluate the potential and save data.

One final remark regards the validity of the approach. In principle, the underlying hypoThesis for the FFT is under periodic boundary conditions. This is not the case in momentum space, where the velocity values are naturally bounded. However, if the chosen maximum velocity $V_{\max}$ is much larger than the velocities actually present in the system, i.e. $|V_{\max}| \gg \max|v|$, the use of periodic boundary conditions provides a good approximation. Under this condition, the probability of particles reaching the boundaries of the velocity domain is negligible, and the artificial periodicity does not significantly affect the dynamics. In effect, the periodic boundary conditions emulate an unbounded domain while keeping the computational problem tractable. This is analogous to the numerical artifact of the zero padding at the border in the context of FFTs.

It should be noted, however, that if $|V_{max}|$ is not chosen appropriately and the velocities of the system approach the boundaries, artificial wrap-around effects can occur, leading to aliasing in the velocity space. Such effects can distort the computed distribution function and introduce spurious interactions, so it is crucial to choose $V_{\max}$ sufficiently large to minimize these errors.

3.1.2 Spectral method for Schrödinger-Poisson

In a one-dimensional case the Schrödinger-Poisson equations reads as

$$i\frac{\partial}{\partial t}\Psi(x,t) = \left(-\frac{\lambda}{2}\partial_x^2 + \frac{1}{\lambda}U(x,t)\right)\Psi(x,t)\,; \tag{3.28}$$

$$\nabla^2 U(x,t) = \alpha(|\Psi(x,t)|^2 - 1)\,, \tag{3.29}$$

where $\lambda = \hbar/m$ is the quantum diffusion length scale and $\alpha = 4\pi G\bar{\rho}$ an amplitude coefficient for the potential. As explained in Ch. 1, a direct correspondence can be made with VP when $\bar{\rho}$ is the mean density of the universe, and $\lambda \to 0$. Moreover, we identify with $|\Psi(x,t)|^2$ the normalized density field $\rho(x,t)/\bar{\rho}$. Remembering that this time we are working with a complex wave function $\Psi(x)$, we will make use of the pseudo spectral method, as in [88] is shown to be the most reliable approach. From Eq. (3.28) we notice how the definition of the Hamiltonian of the system naturally emerges as

$$\frac{\hat{H}(x,t)}{\hbar} = \left(-\frac{\lambda}{2}\hat{\partial}_x^2 + \frac{1}{\lambda}\hat{U}(x,t)\right)\,. \tag{3.30}$$

In this way, we can rewrite Eq. (3.28) in the form $i\partial_t\Psi = (\hat{H}/\hbar)\psi$. This system evolves according to quantum mechanical laws with the time-evolution operator

$$\hat{O} = \exp\left\{-i\frac{\hat{H}}{\hbar}dt\right\}\,. \tag{3.31}$$

Then, with a leapfrog scheme $\hat{O}$ is split into the kinetic and potential part. The kinetic energy operator $\hat{T} = -(\lambda/2)\hat{\partial}_x^2$ in momentum space becomes $\hat{T} = (\lambda/2)\hat{k}^2$, with k the Fourier frequencies. The potential part we remember to be $\hat{V} = (1/\lambda)\hat{U}(x,t)$ solves the Poisson equation. Identifying with $\Psi_i^t = \Psi(x_i,t)$ the number associated with the value of the wavefunction on the point of the grid x_i at time t; the evolution scheme becomes the following:

- An FFT maps Ψ_i^t in the frequencies domain $\tilde{\Psi}_k^t = \mathrm{FFT}(\Psi_i^t)$
- The first kinetic kick of $\Delta t/2$ is:

$$\tilde{\Psi}_k^* = \exp\left\{-i\frac{\lambda k^2}{4}\Delta t\right\}\tilde{\Psi}_k^t \tag{3.32}$$

- An inverse FFT returns to position domain $\Psi_i^* = \mathrm{FFT}^{-1}\tilde{\Psi}_k^*$
- The potential U is then found using a Poisson solver to find the solution of Eq. (3.29) where the source term is $|\Psi_i^*|^2$
- The wavefunction is updated using U

$$\tilde{\Psi}_i^{(1)} = \exp\left\{-i\frac{U}{\lambda}\Delta t\right\}\Psi_i^* \tag{3.33}$$

- Going back in momentum space to give the final kinetick kick advances to the next timestep

$$\Psi_i^{t+1} = \mathrm{FFT}^-1\left(\exp\left\{-i\frac{\lambda k^2}{4}\Delta t\right\}\mathrm{FFT}\left(\Psi_i^{(1)}\right)\right) \tag{3.34}$$

Note that when data storage is not required in the final step, the last kick can be extended to length Δt. This avoids two unnecessary FFTs, reducing computational cost.

3.1.3 Particle mesh

To solve the Vlasov–Poisson system with the N-body approach, we implement a particle-mesh code based on the Cloud-in-Cell (CIC) interpolation scheme. This is a straightforward, yet effective method that offers a good balance between implementation effort and accuracy. We have described the general framework in Sec. 1.2.2; here we discuss the specific numerical choices and implementation details.

The first step in an N-body method is to sample the positions and velocities of the particles from the initial distribution $f_0 = f(x, v, 0)$. This sampling process introduces Poisson noise that scales as $1/\sqrt{N_p}$, where N_p is the number of particles.
To reduce this shot noise, we employ a Sobol quasi-random generator instead of standard pseudo-random numbers. Sobol sequences are designed to fill a space more uniformly than purely random points, which helps to reduce the effective Poisson noise in the initial particle distribution.
Each Sobol sequence produces points $\mathbf{x}_n = (x_n^{(1)}, x_n^{(2)}, \ldots, x_n^{(d)})$ in $[0,1]^d$ deterministically, so that each new point fills the remaining largest gaps. For example, in one dimension, the first point might be 0.5, the next two points 0.25 and 0.75, and the following points 0.125, 0.375, 0.625, 0.875, etc. This ensures that the points quickly cover the interval evenly, avoiding large gaps or clustering typical of purely random sampling.
Mathematically, the n-th draw is generated using the binary expansion of the index n,

$$n = b_0 2^0 + b_1 2^1 + b_2 2^2 + \ldots, \qquad b_k \in \{0, 1\},$$

together with a set of precomputed *direction numbers* $h_k^{(j)}$, which determine how each bit contributes to the final point in dimension j. The j-th coordinate of the n-th point is then

$$x_n^{(j)} = b_0 h_1^{(j)} \oplus b_1 h_2^{(j)} \oplus b_2 h_3^{(j)} \oplus \ldots,$$

where $\oplus$ denotes the bitwise XOR operation, and the direction numbers $h_k^{(j)} = m_k^{(j)}/2^k$ are derived from a primitive polynomial on $\mathbb{F}_2$.
A primitive polynomial over $\mathbb{F}_2$ is a polynomial with coefficients 0 or 1, chosen such that the generated sequence of binary numbers cycles through all possible patterns before repeating. In practice, this ensures that the Sobol points are well-balanced and avoid unwanted regularities. For example, for a 3-bit sequence in one dimension, a primitive polynomial might generate the order of points as $0.5, 0.25, 0.75, 0.125, 0.375, 0.625, 0.875$, exactly filling in the gaps of the previous points.
By combining the binary representation of the index, the direction numbers, and the primitive polynomials, the Sobol generator produces a sequence that is both deterministic and low-discrepancy. Each dimension $x_n^{(j)}$ is generated independently in this way, and the resulting set of points $\{\mathbf{x}_n\}$ uniformly covers the phase space. This approach significantly reduces sampling noise compared to purely random draws, improving the accuracy of the initial condition representation while preserving the statistical properties of the distribution.

Time integration is performed using a drift-kick-drift (D-K-D) leapfrog scheme. A single time step proceeds as follows:

- **First drift:** advance the positions by half a time step using the current velocities:

$$q_i^{t+\frac{1}{2}} = q_i^t + \frac{1}{2} v_i^t \Delta t.$$

- **Compute forces:** evaluate the density $\rho(q^{t+\frac{1}{2}})$ using a CIC 1.2.2 interpolation of the intermediate positions. Solve the Poisson equation, Eq. 3.3, to obtain the potential. Compute the accelerations a_i from the potential using finite differences and deposit them on the particles using the same CIC scheme.

- **Kick:** update the velocities using the computed accelerations:

$$v_i^{t+1} = v_i^t + a_i \Delta t.$$

- **Second drift:** advance the positions by another half time step using the updated velocities:

$$q_i^{t+1} = q_i^{t+\frac{1}{2}} + \frac{1}{2} v_i^{t+1} \Delta t.$$

This scheme maintains second-order accuracy in time while keeping positions and velocities properly staggered, consistent with the standard leapfrog approach.

3.1.4 Poisson Solver

The Poisson equation that we solve in the 1D scenario is presented in Eq. (3.29)

$$\Delta\phi(x) = \alpha\left(\frac{\rho}{\rho^*} - 1\right), \tag{3.35}$$

where $\alpha = 4\pi G\rho^*$. To numerically solve such an equation we employ the convolution approach using the Fourier transform from [52].

Under periodic boundary conditions, the discrete Fourier transform (DFT) of the density, $\hat{\rho}(k)$, is calculated using the fast Fourier transform (FFT). The corresponding gravitational potential in Fourier space is expressed as:

$$\hat{\phi}(k) = \hat{G}(k)\hat{\rho}(k), \tag{3.36}$$

where $\hat{G}(k)$ is the Green's function in Fourier space

$$\hat{G}(k) = -\frac{\alpha\Delta x^2}{4\sin^2(k\Delta x/2)}. \tag{3.37}$$

3.2 Numerical Test of the Solvers

Before going into the real use-case simulation, we tested our solvers on known problems. In this section, we perform numerical test for the VP solvers for different initial condition, to verify the behavior of the integration scheme. All simulations are performed with $\rho^* = G = 1$. We have taken into account the evolution up to a single dynamical time $T = \sqrt{1/G\rho^*} = 1$. As maximum velocity, we use $V_{max} = 2L/T$.

3.2.1 Details on numerical implementation

The test cases that we will consider for this work have a relatively small size and were only conceived for the purpose of comparing the accuracy of three approaches, rather than their computational performances. The bigger test-case presented will be a 2D VP simulation with $64^2 \times 128^2$ grid points; nonetheless, we tried to optimize the codes as best as possible, while keeping them in python for an easy readability and access. We have run all simulations on an a laptop with the $M2$ processor, exploiting Python vectorization and, where this was not enough, we used `numba` version 0.61.0 [89] for parallelization and just-in-time (JIT) compilation to speed up the code. Numba translates selected Python functions into optimized machine code at runtime, allowing them to run much faster and, when used in parallel mode, to distribute computations across multiple CPU cores efficiently. Both spectral methods employ in-place operation and pre-allocated arrays in oderd to speed up execution time.

The code is written in Python, using `numpy` and `scipy` to implement the FPC scheme. A significant portion of the code was vectorized using `numpy`, which improved runtime performance. For the spectral method, we employed the `PyFFTW` which works exactly like $FFTW$ for C. We worked with pre-allocated arrays in order to optimize the performance.

3.2.2 Timestep

As already anticipated, all the evolutions in this section are done up to a dynamical time $T = (G\,\bar{\rho})^{-1/2} = 1$. Neither the FPC scheme nor the spectral method strictly necessitates the Courant-Friedrichs-Lewy CFL condition, however, for both schemes we have chosen to allow the maximum displacement for a single timestep to be a fraction $C = 1/2$ of the cell to ensure a more stable solution [84].
We choose the starting timestep as in [61]:

$$\Delta t = C \cdot \min \left\{ \Delta t_p = \frac{\Delta x}{V};\ \Delta t_v = \frac{\Delta v}{2 \max_i |\partial_x \phi_i|}\,, \right\} \tag{3.38}$$

where the factor $1/2$ comes from the fact that we do half timestep in the velocity integration. The position timestep is constant throughout the whole integration process and, while Δt_v might change, because of the dependence on the force, does not happen in our simulations.

3.2.3 Maxwellian initial condition

The first scenario in which we aim to test the FPC and the spectral solver is represented by a smooth initial distribution. We aim to verify the conservation laws mentioned in Sec. 3.1.

A commonly used test for the accuracy of a chosen integration scheme for VP is the evolution of the Maxwellian distribution [61], [84], [85]

$$f(x, v, t) = \frac{\rho^*}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{v^2}{2\sigma^2}\right) \left[1 + A \sin\left(\frac{2\pi}{L}x - \frac{\pi}{2}\right)\right] \tag{3.39}$$

with $A = 1/2$ and

$$\sigma^2 = \frac{4\pi G \rho^*}{2\pi/L} \mathcal{R}^2\,. \tag{3.40}$$

Adjusting the ratio $\mathcal{R} = k/k_j$ between the wave number $k = 2\pi/L$ and the Jeans wave number k_j we can move from linear to non-linear dynamics, specifically, by setting $\mathcal{R} = 1/2$, we study the case of non-linear Landau damping.

The results are presented in Fig. 3.1, where we tested both integration schemes on a grid composed of $N = 512$ by $M = 1024$ points, presenting the evolution of physical observables that should be conserved by a VP system. As it is shown, all quantities are preserved up to 10^{-10}, with the only side notes concerning the fact that the spectral method conserves energy up to 10^{-4} and the FPC the L_2 norm up to 10^{-5}. We can thus state that all the results hint towards the fact that both integration schemes work correctly.

3.2.4 CDM Initial condition

Let us consider now the CDM case of a sinusoidal perturbation on the density:

$$\rho(x, v, t = 0) = 1 + A \sin\left(\frac{2\pi}{L}x - \frac{\pi}{2}\right)\,, \tag{3.41}$$

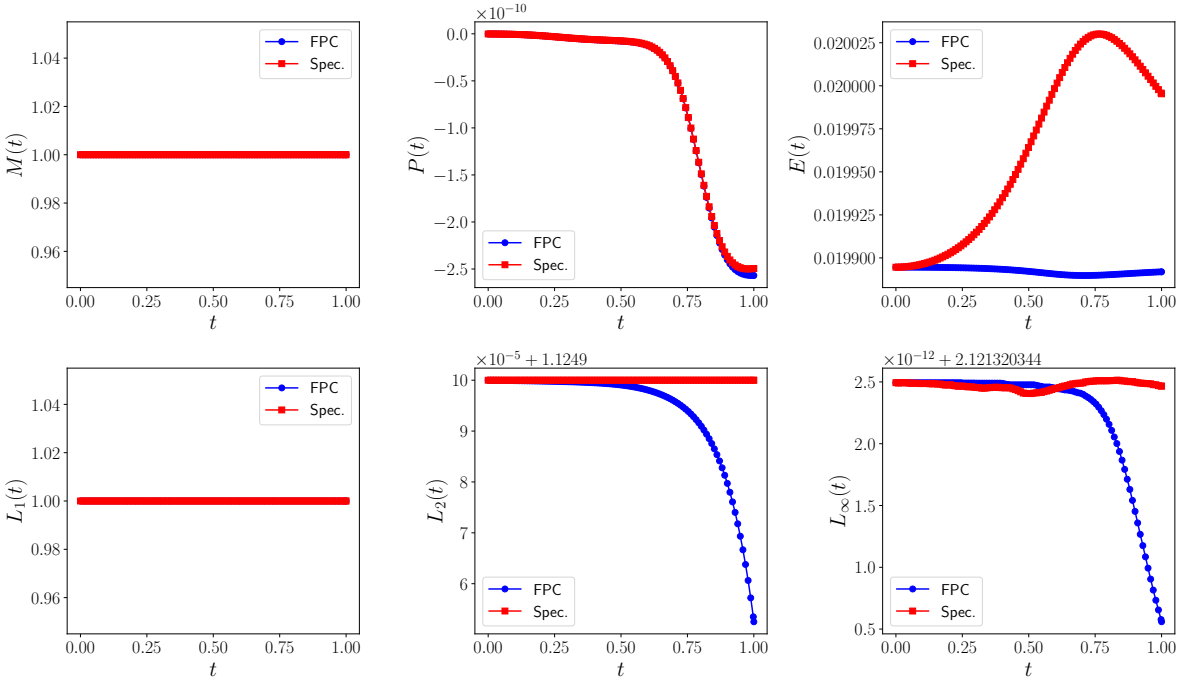


Figure 3.1: Smooth initial condition : Evolution of physical observables (top row) and norm (bottom row) as a function of time in a simulation of a VP system with $N = 2^9$, $M = 2^{10} + 1$. Starting from the top-left corner we find, Mass $M(t)$, momentum $P(t)$ and energy $E(t)$; from bottom-left corner we find the L_1, L_2 and $L - \infty$ norm of the distribution function $f(x, v, t)$. The red line is obtained using a spectral method, the blue one using the FPC.

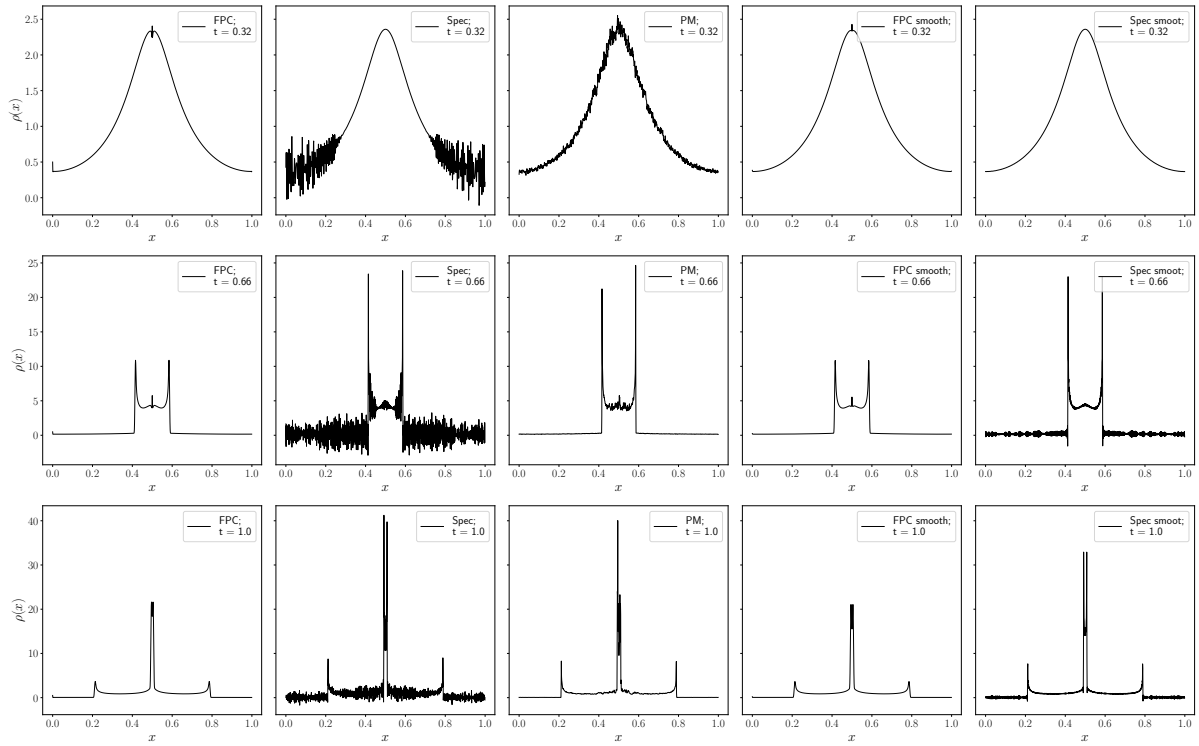


Figure 3.2: Comparison of densities $\rho(x)$ between VP results and a PM simulation (center column). The VP simulation are run on a $N \times M = 1024 \times 2049$ grid, while the PM simulation runs on 1024 grid points and 10^6 particles. The two column on left are CDM simulation done with the FPC and Spectral method respectively and discontinuous initial condition (Subsec. 3.2.4), while the two on the right evolves the smoothed initial condition with $\mathcal{R} = 0.04$ (Subsec. 3.2.5). The figure can be read row-wise or column-wise as on each row we compare the density $\rho(x)$ at the same time t for different integration scheme, while on each column we can follow the evolution of the density for a given integration scheme.

where everything is still, in other words, the initial velocity distribution is given by a Dirac's delta $\delta(v - 0)$. Thus, our initial distribution function can be factorized in a velocity part and a spatial density.

$$f(x, v, t = 0) = \delta(v, t = 0)\, \rho(x, v, t = 0)\,. \tag{3.42}$$

Repeating the analysis performed for the Maxwellian initial condition, we simulate the evolution of a Vlasov–Poisson (VP) system on a 1024×2049 grid to ensure that both the spectral method and the FPC scheme conserve the physical quantities outlined in Sec. 3.1. The results are shown in Fig. 3.3, where we observe that the FPC scheme conserves all quantities except for the norm L_2 and the energy, with deviations limited to 10^{-4}. As discussed in [84], the inclusion of slope correctors in the FPC scheme accelerates the decay of the discrete norm L_2. However, when oscillations caused by nonlinear effects are damped or smoothed through grid projection, the L_2 norm stabilizes.

This behavior is not observed in the spectral method. Without any limiter, spurious oscillations can cause the distribution function to become negative, resulting in violations of L_1 norm, momentum, and L_∞ norm conservation.

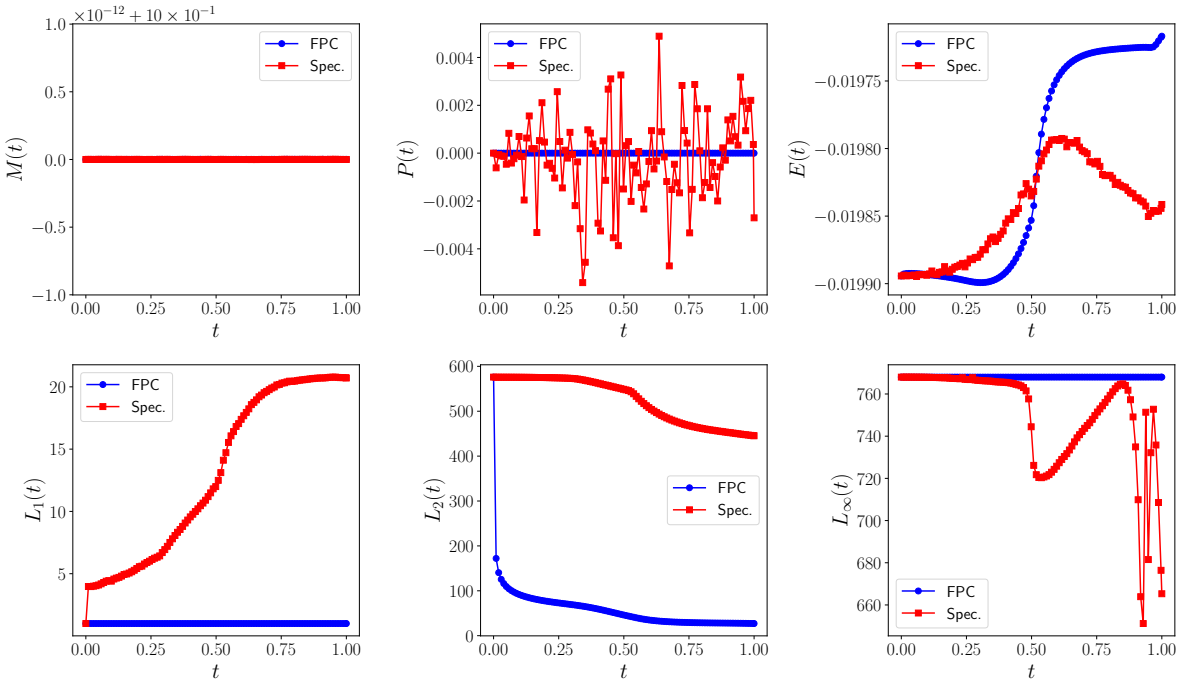


Figure 3.3: CDM stiff initial condition : Evolution of physical observables (top row) and norm (bottom row) as a function of time in a simulation of a VP system with $N = 2^{10}$, $M = 2^{11} + 1$. Starting from the top-left corner we find, Mass $M(t)$, momentum $P(t)$ and energy $E(t)$; from bottom-left corner we find the L_1, L_2 and $L - \infty$ norm of the distribution function $f(x, v, t)$. The red line is obtained using a spectral method, the blue one using the FPC.

If the analysis is restricted to the conservation of the aforementioned quantities, one might conclude that the FPC scheme accurately reproduces the VP dynamics, while the spectral method does not. However, this conclusion is misleading. Fig. 3.2 compares the densities obtained using the FPC scheme and the spectral method with those from a particle-mesh (PM) simulation, which serves as a reference. The first three columns reveal that the FPC scheme (first column) produces anomalous density peaks at $x = 0$ and $x = 1/2$. Furthermore, the density, ρ, appears smoother compared to both the spectral method (second column) and the PM simulation (center column). This smoothing arises from a combination of the stiff, discontinuous initial condition and the limiters applied in the FPC scheme.

The discontinuity is most prominent along the velocity axis, where the function resembles $\delta(v)$. Analytical evolution would shift this function in accordance with Eq. (3.5), resulting in $\delta(v - F\Delta t)$. To compute these values numerically, interpolation is required. Fig. 3.4 illustrates how a delta function evolves under the FPC and spectral methods. Instead of propagating a delta function, $\delta(v - F\Delta t)$, as expected, the FPC scheme produces a smoothed, truncated version, which accounts for the smoothing observed in ρ.

This issue does not arise in the spectral method, as the negative values retain information about the shifted delta in Fourier space. However, the spectral method introduces noise along with the desired information, and spurious oscillations cannot be suppressed due to the absence of limiters.

In addition to smoothing the distribution function, the above phenomenon causes two stationary density peaks to appear in the FPC results at $v = 0$ and $F = 0$. These two remain in fact stationary points during the whole evolution because the smoothed density prevents the potential (and consequently the force, F) from growing large enough to destabilize these points.

3.2.5 Smoothed initial condition

If the previous considerations are true, the main obstacle is Dirac's delta in the initial condition. We thus propose as an alternative to the stiff discontinuous CDM initial condition a smoother version, which however does not change the core of the dynamics. The Dirac's delta can be seen as an extreme case of a Gaussian distribution as $\sigma \to 0$. Thus we choose to recover the distribution function in Eq. (3.42) with small $\mathcal{R} = 0.004$ to obtain a softer version of the Dirac's delta, where instead of having one point different from zero we have three. In fact, from Eq. (3.40) we notice how choosing a small $\mathcal{R}$ corresponds to almost zero variance. This small numerical artifact provides a better starting point for both integration schemes at a small price: a very slight diffusion.

We report the results obtained on a 1024×2049 simulation in Fig. 3.5 and Fig. 3.2.

Compared with the results obtained in the previous section (Fig. 3.3), we immediately observe significant improvements in the spectral method. The L_1 norm smoothly transitions from 1 to a maximum of 5, in contrast to the abrupt discontinuity observed in Fig. 3.3. This behavior arises because smoothing the initial conditions eliminates the initial presence of negative values in the distribution function. For the same reason, Fig. 3.5 shows that momentum is conserved to a precision of 10^{-4}, compared to 10^{-2} in the case of discontinuous initial conditions. Additionally, the L_∞ norm is effectively conserved by the spectral method when the initial conditions are smoothed, as the relative variation is

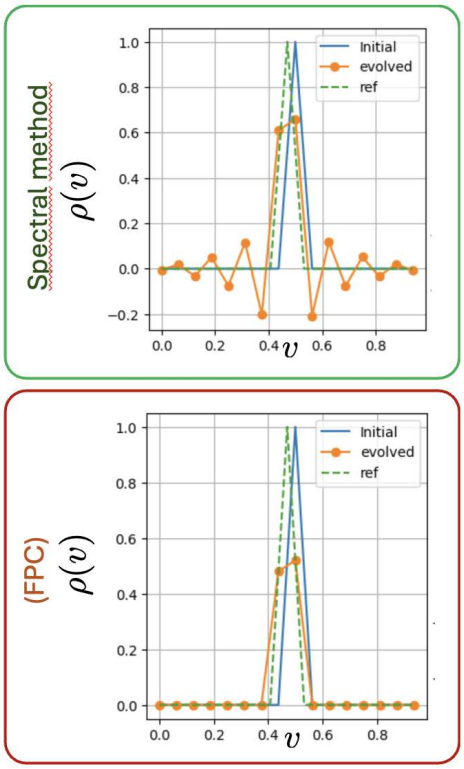


Figure 3.4: Toy model of the first iteration of FPC and spectral method on a Dirac's delta. The blue line represents the initial condition, the green dashed line the analytical expression and the yellow points represents the evolution that is done by the integration scheme.

$\mathcal{O}(10^{-4}/10^{2}) = \mathcal{O}(10^{-6})$. Finally, the L_2 norm exhibits an approximate variation of order unity, corresponding to a relative variation of $\mathcal{O}(10^{-2})$. This is significantly smaller than the relative variation of $\mathcal{O}(1)$ observed for the discontinuous case (Fig. 3.3).

This improvement arises because the spectral method is highly sensitive to steep discontinuities and lacks mechanisms to regulate spurious oscillations. However, despite the improvements introduced by the spectral method once the stiffness issue is resolved, no clear-cut winner between the spectral and the FPC methods emerges from examining the physical quantities shown in Fig. 3.5. Therefore, we turn our attention to the behavior of the density function.

In Fig. 3.2, we can clearly see how smoothing has benefited both the FPC and the spectral schemes. For the spectral method (rightmost column), oscillations are significantly damped, and while negative densities persist, they are comparable to minor numerical noise. The peak heights align almost perfectly with the PM results. A closer examination reveals a slight diffusion effect due to the smoothed initial condition, as the function lacks the sharp edges observed in its counterpart with discontinuous initial conditions (second column).

Regarding the FPC scheme (second column from the right), we observe only minor changes. The density is slightly less smooth, but the most notable improvement is that the peak at $x = 0$ has almost disappeared. This supports our hypoThesis that the primary issue lies in the integration of a steep, discontinuous distribution function. Smoothing the initial condition has effectively eliminated one of the peaked structures.

Even though the spectral method exhibits negative values, which are non-physical, the overall physical observables are accurately reproduced when the smoothed initial condition is evolved using the spectral method. The next step in our analysis will consider how the regularization of the initial conditions impacts the negative values in the evolution. ——————————-

3.2.6 The right smoothing choice

In this section we address two closely related questions: what is the minimum amount of smoothing required to eliminate negative values in the density, and what is the cost of this operation in terms of physical fidelity. We also assess how the resulting density compares with the reference case.

We recall that replacing the Dirac delta in Eq. (3.42) with a narrow Gaussian regularizes the problem without altering its essential physical properties. Nevertheless, a more systematic analysis of how this Gaussian should be chosen is warranted. A convenient approach is to adopt the standard form of a Gaussian distribution and to express its standard deviation in terms of the velocity resolution Δv:

$$\rho(v) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{v^2}{2\sigma^2}\right), \qquad \sigma = \frac{C\,\Delta v}{6}. \tag{3.43}$$

The parameter C encodes the width of the Gaussian in units of velocity cells Δv. For example, $C = 5$ corresponds to a Gaussian spanning approximately five cells, while $C = 13$ and $C = 20$ describe progressively broader kernels.

We compare the resulting density and distribution function f for different values of C against those obtained from a Particle–Mesh (PM) reference solution. This comparison

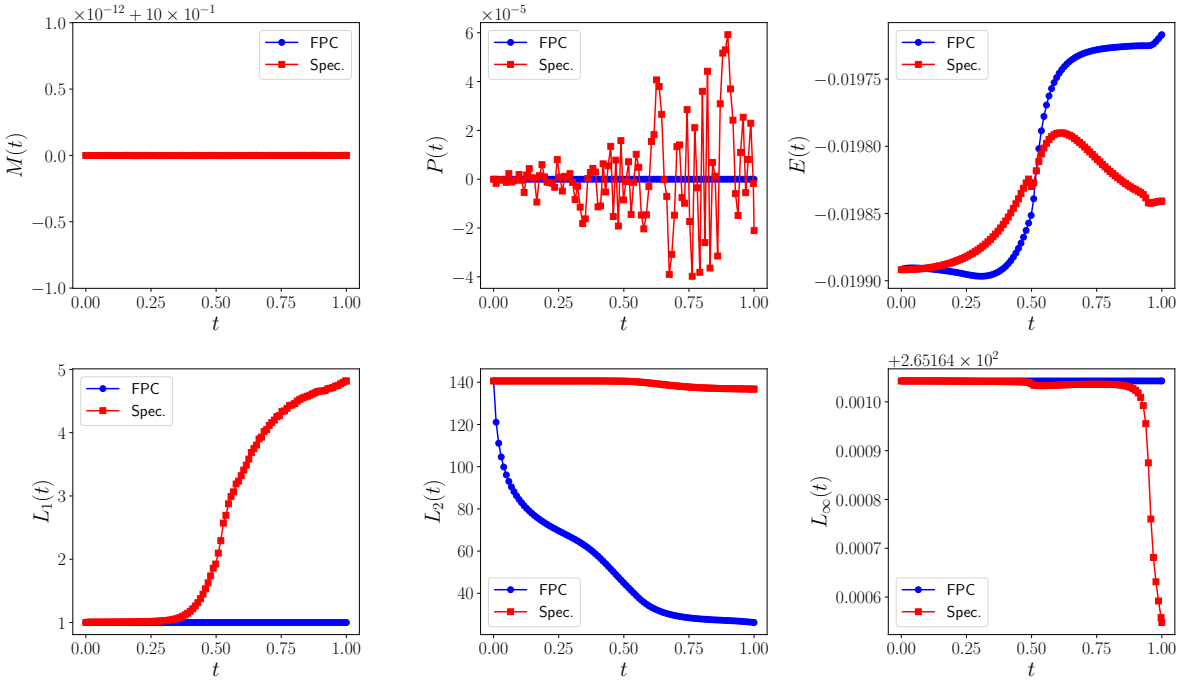


Figure 3.5: CDM smoothed initial condition with $\mathcal{R} = 0.004$: Evolution of physical observables (top row) and norm (bottom row) as a function of time in a simulation of a VP system with $N = 2^{10}$, $M = 2^{11} + 1$. Starting from the top-left corner we find, Mass $M(t)$, momentum $P(t)$ and energy $E(t)$; from bottom-left corner we find the L_1, L_2 and $L - \infty$ norm of the distribution function $f(x, v, t)$. The red line is obtained using a spectral method, the blue one using the FPC. The y-axis have different range than in the previous plots.

is carried out at two different resolutions in order to verify that the choice of C is not related to the specific discretization, as it is defined directly in terms of Δv. The two simulations employ grids of 1024×2049 and 512×1025 points, respectively.

The results are shown in Fig. 3.6, where we first examine the simulations with fixed resolution. Increasing the smoothing parameter C progressively regularizes the density: for $C = 18$, no negative values or oscillations are observed. However, this comes at the cost of not capturing the right height of the density peaks.

Importantly, the optimal choice of C is largely independent of resolution, since the Gaussian width is defined relative to the velocity-space discretization Δv.

These findings provide practical guidance for CDM simulations using the VP solver. If avoiding negative densities is the main concern, $C = 18$ is appropriate; if accurate reproduction of peak heights of PM is prioritized, $C = 13$ offers the best compromise. In general, $C = 13$ captures the PM peaks faithfully while introducing only minimal oscillations.

3.2.7 Summing up

In summary, we showed how VP solvers fail in correctly representing the evolution in case of CDM-like initial conditions. The problems arise from the steep discontinuity in the initial condition, which standard schemes cannot handle reliably. Introducing a minimal smoothing removes this numerical stiffness without affecting the physical dynamics. With this adjustment, the spectral method accurately reproduces the density and distribution function while maintaining good conservation properties, outperforming the FPC scheme. We therefore use, in our comparison simulations, a gaussian of total width of approximately $C = 13$ gridpoints, together with a spectral integrator for discontinuous VP problems. The velocity domain will contain, in order to avoid aliasing, double the number of gridpoints than the spatial domain; i.e., if N gridpoints are chosen to represent one of the spatial dimensions, the number of grid points for the velocity domain will be $2N - 1$, in order to account for the presence of $v = 0$.

3.3 Numerical Comparison

In this section, we compare the results obtained with the three VP solvers: the direct spectral method, PM, and SP. We assess convergence in two test cases: a one-dimensional sinusoidal perturbation and a two-dimensional collapse. The primary goal is to identify the scenarios in which the SP method accurately reproduces the correct VP results. Additionally, we examine whether even in these simple settings there are dynamical features that the PM method fails to capture.

3.3.1 One-dimensional sinusoidal perturbation

We start with the simple case of a one-dimensional sinusoidal perturbation, as defined in Eq. (3.41), assuming an initially static configuration. A crucial assumption is that the velocity field is single-valued, which allows the initial distribution function to be represented by a Dirac delta, as in Eq. (3.42).

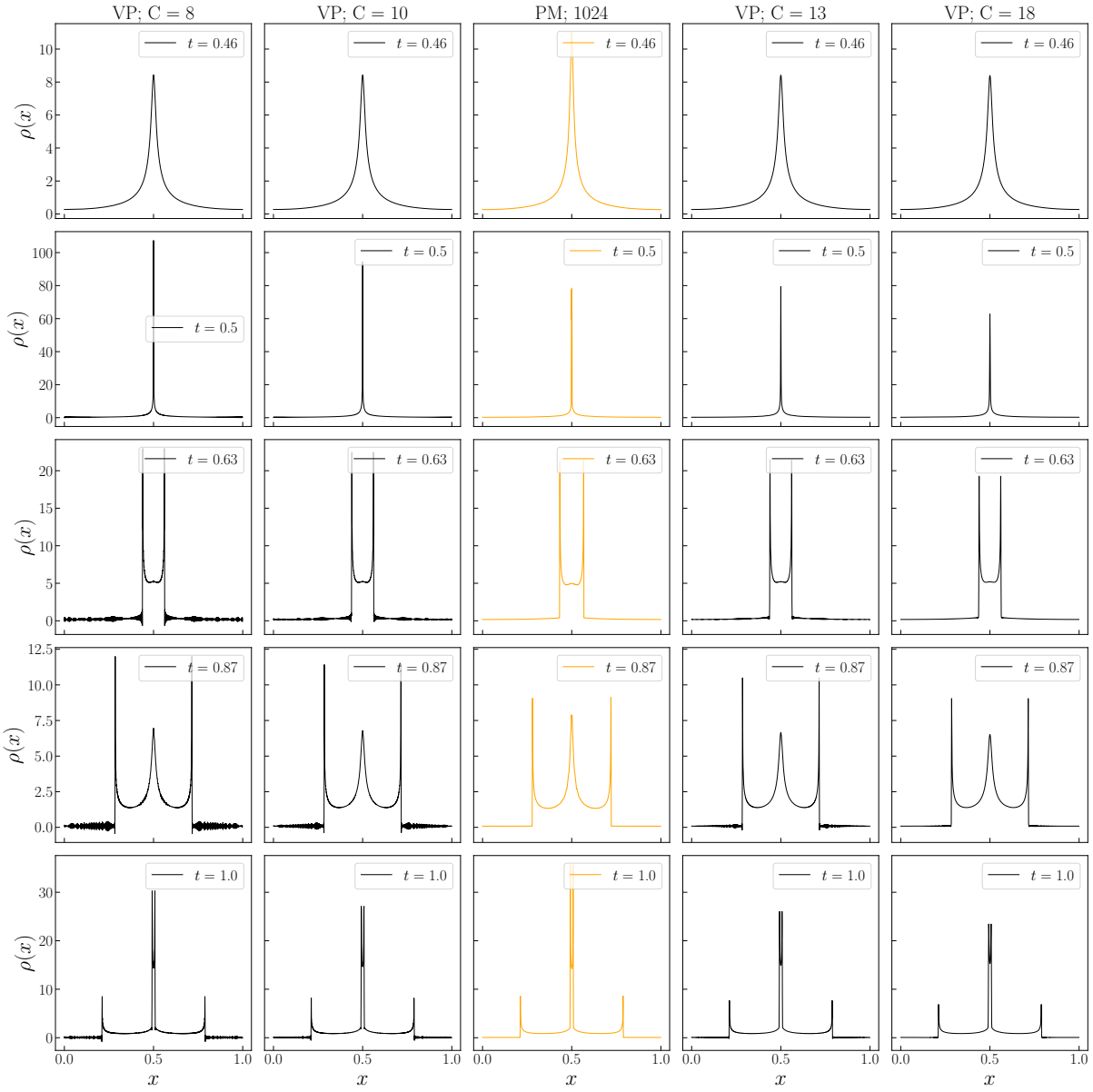


Figure 3.6: Density $\rho(x)$ as a function of the position for different smoothing of the initial condition $\rho(v)$ obtained with the spectral VP solver (black) compared to the reference PM simulation (orange). Read the figure in columns. The simulations are done on a 1024×2049 grid. The value C represent the the with of the gaussian on term of grid spacing Δv.

The numerical setup is as follows. The spatial domain is discretized with $N = 1024$ points over a length $L = 1$, while the velocity space is sampled with $2N-1$ points spanning $[-V_{\max}, V_{\max}]$, explicitly including $v = 0$. We set $\bar{\rho} = G = 1$ and evolve the system up to a final time $T = 1$. Based on previous analysis, we adopt a smoothing coefficient $C = 13$ for the VP solver, effectively replacing Eq. (3.42) with a Gaussian of width $13\,\Delta v$.

We start our analysis with the smallest value of λ that can be reliably resolved at this resolution, $\lambda = 5 \times 10^{-4}$. For smaller values, the dynamics are no longer accurately captured. Figure 3.7 displays the density profiles at selected representative times. The first row corresponds to the linear regime prior to shell crossing, where all three solutions are in excellent agreement. This is followed by the collapse phase, still preceding shell crossing. In this stage, the SP density peak is slightly lower, although the qualitative agreement remains good.

The last three rows show snapshots taken after shell crossing. At this point, the two density peaks associated with the perturbation cross each other, and the velocity field ceases to be single-valued. This evolution is clearly visible in Fig. 3.8 and represents the most critical and physically relevant phase of the dynamics. Again, in this section we adopt the PM solution as the reference. In the case of simple dynamics with no free-streaming, such as the one considered here, its accuracy is well established. As observed in previous sections, the VP solver reproduces the PM results well, apart from numerical oscillations associated with a shock-like phenomenon. The SP solver, in contrast, exhibits small-scale quantum-like fluctuations. The amplitude of these fluctuations depends on λ; for fixed N, choosing too small a value of λ leads to pronounced sub-grid effects that hinder accurate resolution. Despite these fluctuations, the overall density profile remains well captured.

In Figure 3.8 we compared the distribution functions obtained with the three solvers, at the same time-frames as in Fig. 3.7. For the SP solver, the phase-space distribution is constructed following the procedure described in Sec. 1.2.3, using a Gaussian smoothing parameter $\mu = 4\,\Delta x$ to regularize the initial velocity distribution (i.e., using a Gaussian instead of a Dirac's delta). The resulting distribution is slightly broader, indicating a small velocity dispersion introduced by the convolution. In contrast, the VP solution shows barely any smoothing, with effects arising solely from the regularization of the initial condition.

Based on previous observations, our next goal is to determine the appropriate value of λ for a fixed N. We then choose $N_{SP} > N$ to ensure that the dynamics is accurately resolved. The resulting density is subsequently coarse-grained to map it back to a grid of N points. For consistency, the reference VP and PM simulations were kept at $N = 1024$, while λ and N_{SP} were varied until a faithful representation of the dynamics was achieved.

Figure 3.9 shows the density profiles in the same time frames as in Fig. 3.7. In this case, the SP solver is initially run on a grid of $N_{SP} = 2^{15}$ points with $\lambda = 2.5 \times 10^{-5}$. The results are then recursively smoothed to obtain a representation on a grid of 1024 points.

This procedure reproduces the PM density remarkably well, with only negligible residual oscillations visible. The accuracy of SP in capturing the PM density highlights its potential to effectively emulate VP dynamics in practice. These small oscillations are a natural consequence of the quantum nature of the SP method and can be further reduced by increasing the simulation resolution and applying additional smoothing. In this context, the smoothing procedure plays a role analogous to softening or the choice of grid

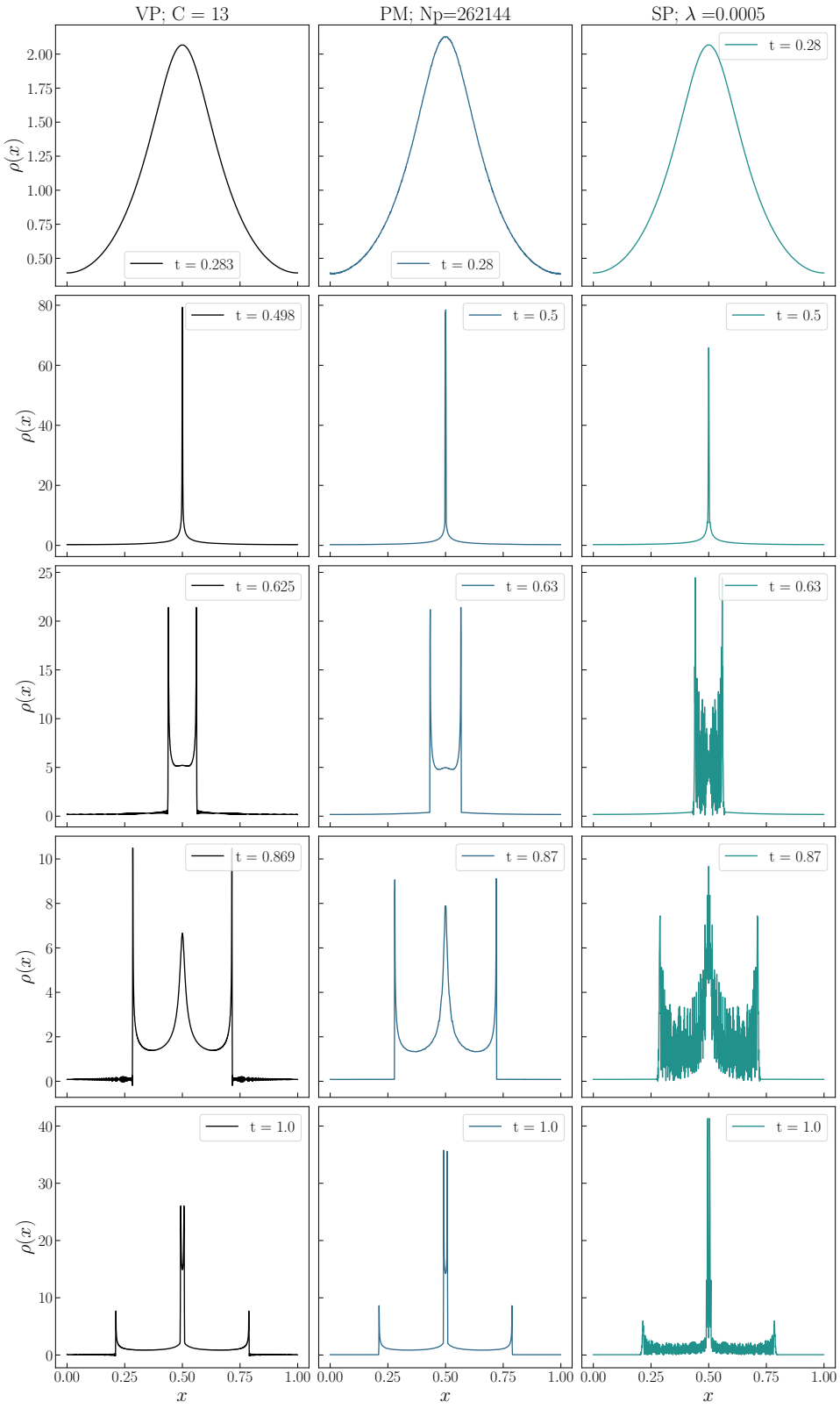


Figure 3.7: Evolution of density for a sinusoidal perturbation with three different solver on 1024 grid points. The figure reads in columns. We compare a direct VP spectral solver (left), a PM with 2^{18} particles (center) and a solution of the SP equation with $\lambda = 5 \times 10^{-4}$. We smoothed the initial velocity distribution for the VP solution using a gaussian of width $C = 13\Delta v$, in place of a Dirac's delta.

size in classical N-body codes, helping to control numerical artifacts while preserving the overall structure of the density field. This result demonstrates that SP can serve as a practical and accurate tool to emulate classical dynamics.

It is also worth noting that the time frames do not perfectly coincide between SP and PM. This discrepancy arises because the SP solver is run at higher resolution with a different time step compared to the VP and PM simulations.

To examine the correspondence between the different methods in more detail, we analyze how fluctuations affect the agreement. For this purpose, we take the logarithm of densities and study the probability distribution of the logarithm. The results are shown in Fig. 3.10, where the logarithms of the distributions are compared across different time frames.

We focus on times around and after shell crossing, when oscillations are more pronounced. In this regime, the PM and smoothed SP results are nearly identical, with the minimal SP oscillations comparable to the sampling noise in the PM solution. In contrast, VP exhibits significant oscillations in regions where the density is small.

We also tested two different initial smoothings, $C = 13$ and $C = 18$. Interestingly, when analyzing the fluctuations, the choice $C = 18$ appears to provide the best agreement with the reference solutions.

In summary, for the case of an initially static sinusoidal perturbation, the SP solver reproduces the solution accurately. The only requirement is to run at higher resolution and then compute the coarse-grained density on a lower-resolution grid. The need for higher resolution arises from the requirement to recover the correct small-scale structures.

The VP solver, on the other hand, requires regularization of the initial condition with a Gaussian of width approximately $C = 13$ if small oscillations are acceptable and $C = 18$ if these oscillations are to be fully suppressed, at the cost of slightly lower density peaks.

In this analysis, we focus on the density rather than the full phase-space distribution, as the velocity dynamics is relatively simple. There are no exotic phenomena in this setup where SP fails and VP becomes strictly necessary. Such scenarios could arise in warm dark matter simulations, where the initial velocity distribution is non-trivial [90].

On a side-note, the integration of WDM models is in general motivated by the need to address small-scale structure issues, as discussed in Chapter 1. Analogously, alternative dark matter models formulated directly in the SP framework, such as FDM, tackle similar small-scale problems through the wave-like properties of ultralight axions rather than through thermal velocity dispersion.

We performed a series of numerical tests, reporting the execution times in Tab. 3.1. To obtain a VP result on a $2^{10} = 1024$-point grid, SP must be run on a 2^{15}-point grid and subsequently smoothed. In terms of execution times, SP is more efficient than the other approaches, whereas direct VP integration is the most demanding because of the high dimensionality. Note that SP results require coarse-graining and higher-resolution runs to recover the classical smooth solution.

Taking these factors into account, the PM method is the preferred classical approach for this scenario, particularly because the chosen initial conditions favor PM. In CDM systems, where the velocity distribution is sharply peaked, PM performs efficiently and accurately. The SP approach can approximate the VP evolution, but requires higher spatial resolution to capture the wavefunction dynamics faithfully. As a result, on classical hardware, SP does not provide a clear advantage over PM for CDM.

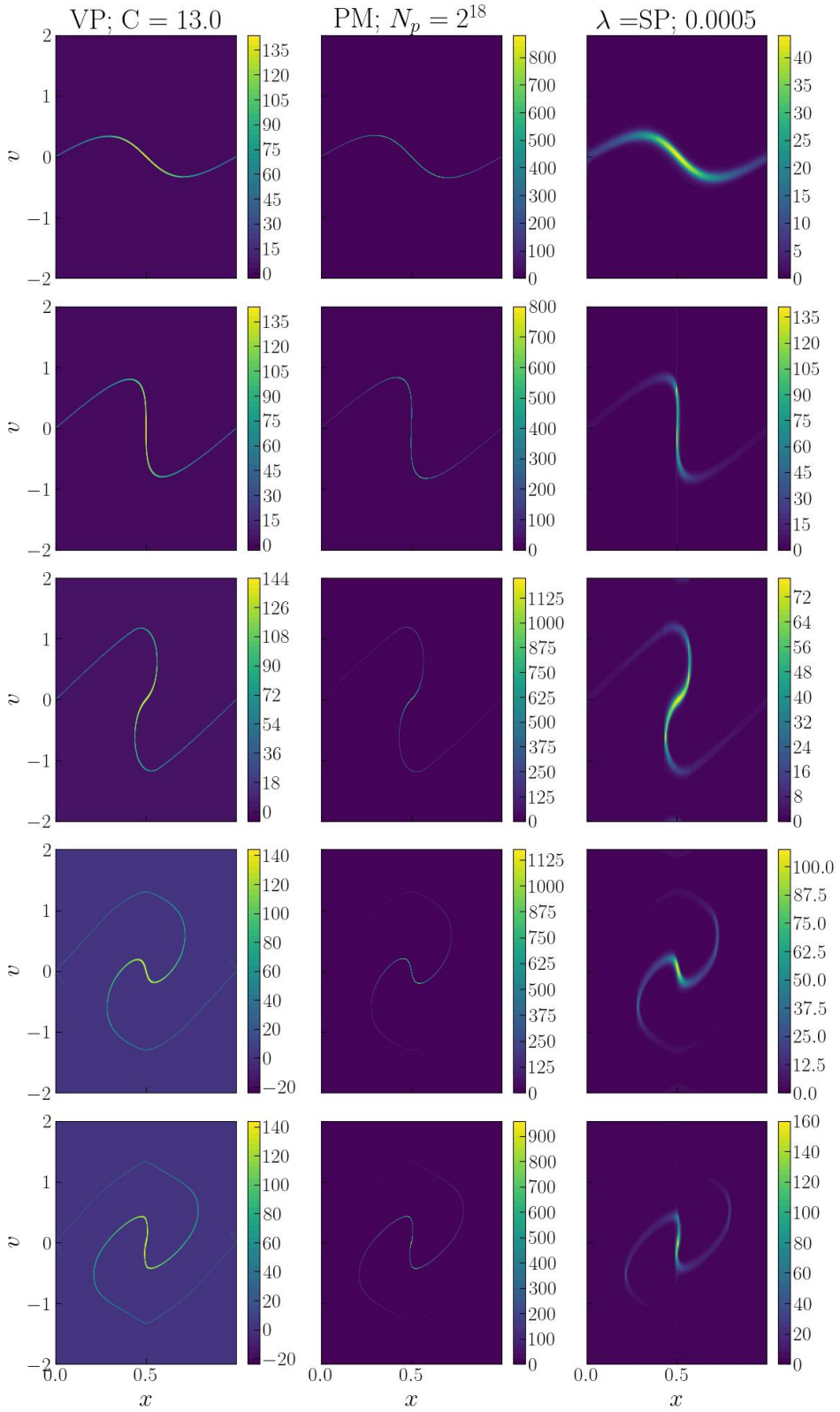


Figure 3.8: Evolution of the distribution function f for a sinusoidal perturbation with three different solver on 1024 grid points. Each column represents the evolution of the distribution function with a different solver, while each rows shares the same time. The plots correspond to times $\{0.28, 0.5, 0.63, 0.87, 1.0\}$. We compare a direct VP spectral solver (left), a PM with 2^{18} particles (center) and a solution of the SP equation with $\lambda = 5 \times 10^{-4}$(left). We smoothed the initial delta distribution for the VP solution using a gaussian of width $C = 13\Delta v$.

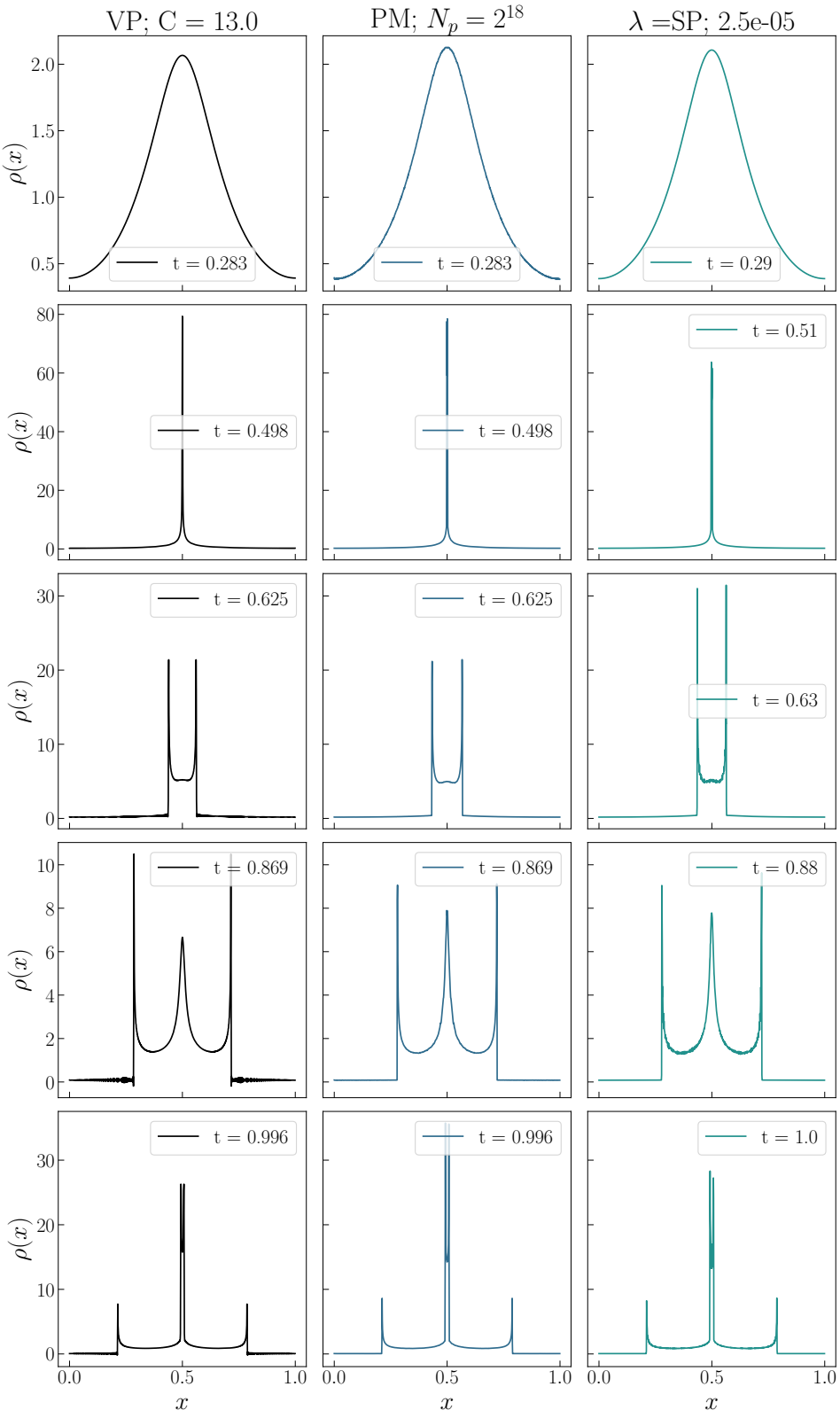


Figure 3.9: Evolution of density for a sinusoidal perturbation with three different solver on 1024 grid points. The figure reads in columns. We compare a direct VP spectral solver (left), a PM with 2^{18} particles (center) and a solution of the SP equation with $\lambda = 2.5 \times 10^{-5}$ (right). We smoothed the initial delta distribution for the VP solution using a gaussian of width $C = 13\Delta v$. The results of the SP simulations are obtained with $N_{SP} = 2^{15}$. The density is then coarse grained onto $N = 2^{10}$ grid. The time frames for SP are slightly different from VP and PM.

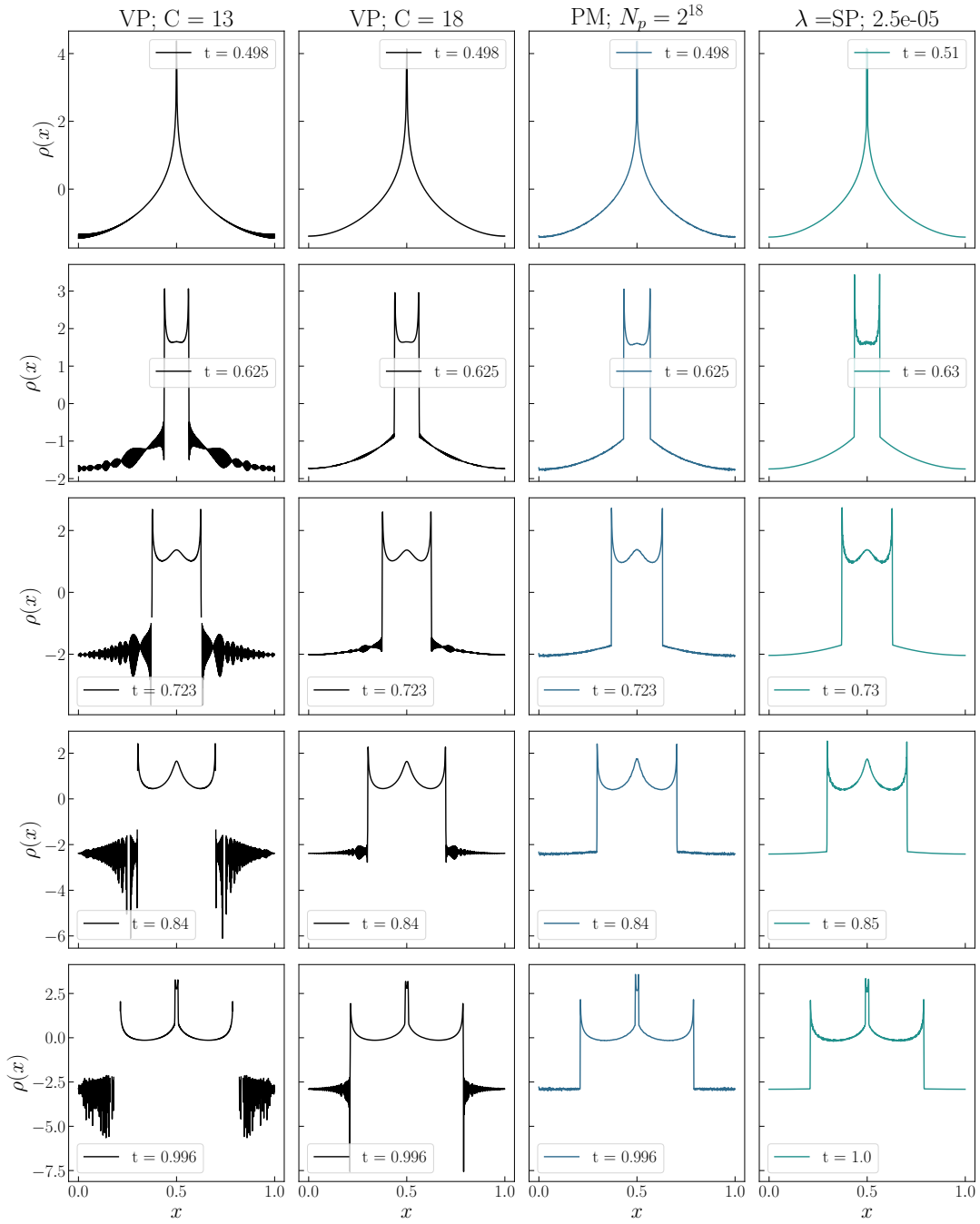


Figure 3.10: Evolution of density for a sinusoidal perturbation with three different solver on 1024 grid points. The figure reads in columns. We compare a direct VP spectral solver (first two column on the left, in black), a PM with 2^{18} particles (third column from the left, in blue) and a solution of the SP equation with $\lambda = 2.5 \times 10^{-5}$ (right-most column, in green). We smoothed the initial delta distribution for the VP solution using a gaussian of width $C = 13\Delta v$ (first left-column) and $C = 18$ (second left-column). The results of the SP simulations are obtained with $N_{SP} = 2^{15}$. The density is then coarse grained onto $N = 2^{10}$ grid. The time frames for SP are slightly different from VP and PM.

	N	N_p	λ	$t\,(minutes)$
VP	2^{10}			2.7
VP	2^{11}			31.6
PM	2^{10}	2^{18}		0.1
PM	2^{12}	2^{18}		0.5
PM	2^{13}	2^{20}		2.8
PM	2^{15}	2^{20}		14.4
PM	2^{15}	2^{21}		29.7
PM	2^{16}	2^{21}		61.1
SP	2^{15}		10^{-4}	2.1
SP	2^{16}		1.25^{-5}	7.5

Table 3.1: Execution time of the different simulations of the one-dimensional sinusoidal perturbation in 3.3.1. Simulations are run on a $M2$ laptop processor.

Nevertheless, for larger systems or higher-resolution studies where PM becomes memory-limited, the VP–SP mapping offers a promising testing ground for quantum algorithms. Quantum computers can exploit qubit superposition to efficiently represent the full Hilbert space, enabling SP-based simulations to handle larger system sizes and finer resolution than is feasible classically. In this sense, while PM remains the most practical choice for classical CDM simulations, SP on a quantum computer provides a pathway toward genuine computational advantage in regimes constrained by classical resources.

Finally, we aim to extend this analysis to the two-dimensional case.

3.3.2 Two dimensional collapse

We now turn to a two-dimensional density perturbation. As in the one-dimensional case, the initial distribution can be factorized into a spatial density component and a velocity component,

$$f(x, y, v_x, v_y, 0) = \rho(x, y)\, \delta_D(v_x, v_y)\,, \tag{3.44}$$

where δ_D denotes the Dirac delta. As a representative example, we consider the collapse of a spherically symmetric perturbation described by a Gaussian density profile,

$$\rho(x, y, 0) = \frac{1}{2\pi\sigma^2} \exp\left[-\frac{(x - L/2)^2 + (y - L/2)^2}{2\sigma^2}\right], \tag{3.45}$$

with $\sigma = L/10$.

The system is evolved in a square domain of size $L \times L$, discretized with $N = 64$ grid points along each spatial direction. As in the previous sections, we set $L = G = \bar{\rho} = 1$ and evolve the system up to a final time $T = 1$. The velocity space is defined by pairs (v_x, v_y) with each component bounded by $|v_i| \leq V_{\max}$, where $V_{\max} = 2$. This space is discretized on

a square grid with $2N-1$ points per direction, leading to a four-dimensional representation of the distribution function with dimensions $64 \times 64 \times 127 \times 127$. To handle data sets of this size efficiently, the arrays are stored on disk using the `zarr` library rather than fully loaded into memory.

As in one dimension, the SP solver must be run at higher resolution and subsequently coarse-grained to allow for a direct comparison with PM and VP. We pushed the SP simulations up to resolutions as high as 2048×2048. However, the computational cost grows rapidly. For this reason, we focus here on results obtained with $N = 1024$ and $\lambda = 5 \times 10^{-4}$, which already provide a meaningful comparison. As shown in Tab. 3.2, a PM simulation on a 64×64 grid requires only 0.03 minutes, whereas SP on a 1024×1024 grid takes approximately two minutes. Increasing the SP resolution further would raise the runtime to roughly twenty minutes, making a direct comparison with the very short PM execution times impractical. Comparing these runs directly is misleading, since the resolutions differ by a factor of 16, and obviously on a classical computer the run-times weight in favor of the PM, with lower resolution. Actually, if PM and SP are compared at the same resolution (e.g, 1024 gridpoints), SP can actually outperform PM. However, the key point is that the SP approach, which requires higher resolutions, is to be considered in the context of a possible advantage on quantum computers, where memory-efficient handling via Hilbert-space methods can make high-resolution simulations feasible.

Figure 3.11 summarizes the results. In the first column, we show the radial density profiles obtained with the three solvers; in the same rows, we display the corresponding two-dimensional density maps $\rho(x, y)$. In the first row, the perturbation is still collapsing, and the three methods are in excellent agreement. The second row corresponds to the collapse phase, during which the agreement remains good.

After collapse, a shell crossing occurs and the VP solution begins to develop slightly negative density values. In subsequent snapshots, the SP and PM results continue to agree well in terms of radial density profiles. This agreement is particularly important because the PM solution serves as a reference in this relatively simple, spherically symmetric setup, allowing us to validate the accuracy of the SP approach.
However, differences start to appear in the two-dimensional density maps. In both SP and VP simulations, numerical effects lead to a mild break of spherical symmetry: small deviations from perfect symmetry emerge due to the different approaches used and boundary effects. In particular, for periodic boundary conditions, the density can be artificially influenced by mass concentrations that appear on the opposite side of the domain, effectively creating spurious interactions as if nearby matter were present beyond the boundaries. These effects are likely amplified by the fact that the initial conditions were chosen too close to the domain edges, so that the finite domain and the imposed boundary conditions interact with the evolving density field, creating small but noticeable fluctuations.

3.4 Summary

In this study, we analyzed the performance of VP, PM, and SP solvers for cold dark matter (CDM) simulations, focusing on initially static perturbations in one- and two-dimensional dimensions. The VP approach retains full phase-space information, but is technically demanding, requiring careful treatment to handle shock-like numerical issues. Although VP can, in principle, provide highly accurate solutions, it is not strictly necessary for

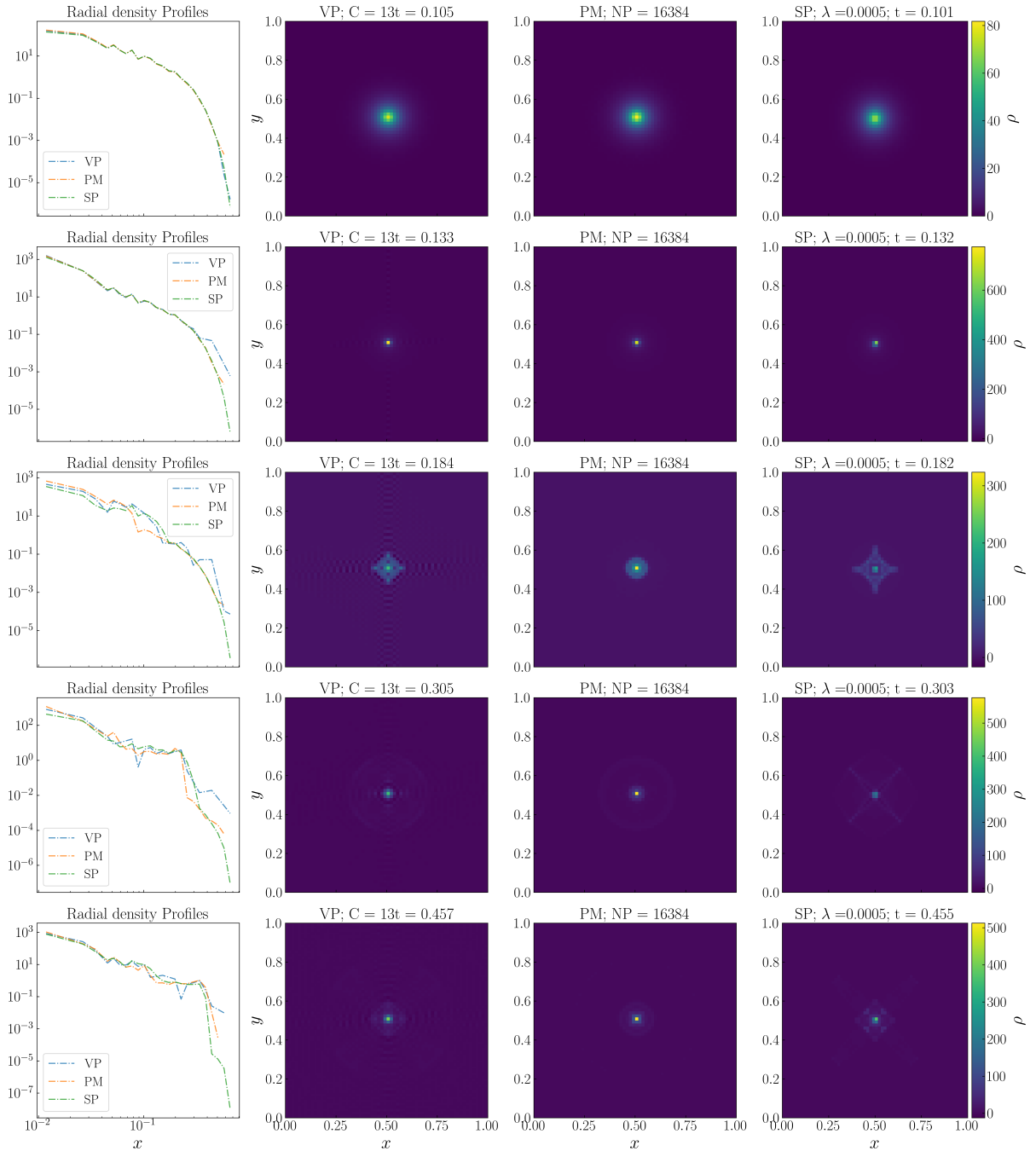


Figure 3.11: (first column) Comparison of the radial density profile of a 2D gaussian perturbation as in 3.3.2 in case of direct VP solver, PM and SP. The results with SP are obtained at higher resolution $N_{SP} = 1024$ and then coarse grained into $N = 64$. PM and SP are obtained with $N = 64$. The other columns represents respectively the density profile for VP (second column), PM(third column) and SP(right-most column).

	N	N_p	λ	$t\,(minutes)$
VP	64			13.2
PM	2^6	2^{13}		0.03
PM	2^{10}	2^{18}		2.8
SP	1024		5×10^{-4}	2.4
SP	2048		10^{-4}	23.4

Table 3.2: Execution time of the different simulations of the two-dimensional Gaussian perturbation in 3.3.2. Simulations are run on a $M2$ laptop processor.

CDM, as PM already reproduces the dynamics with sufficient accuracy and minimal computational cost, at least in these simple scenarios.

A central finding is that the SP solver can reliably simulate the evolution of CDM perturbations. This requires performing the simulations at sufficiently high resolution and subsequently smoothing the results onto the target grid. By doing so, small-scale structures, determined by the parameter λ, are accurately captured, and the solution matches the expected dynamics.
Even though the resolution required by this approach is computationally demanding in the classical scenario, quantum computing could in principle mitigate this requirement. Thanks to the exponential memory advantages afforded by qubit superposition, quantum algorithms may handle large Hilbert spaces more efficiently, potentially mitigating the cost of a higher-resolution run for SP, instead of a PM.

For the scenarios studied here, SP and PM both perform well, while VP can develop small numerical artifacts such as negative densities in post-collapse regions. Our analysis confirms that SP faithfully represents the CDM dynamics when appropriate resolution and smoothing are applied.

We have shown that it is possible to use SP to simulate dark matter as a viable alternative to the PM and VP approaches. Specifically, SP provides a viable alternative to the PM with the potential for further improvements through quantum computing. VP, although powerful, is best reserved for cases requiring the full phase-space description or when small-scale effects are critical.

In the next chapter, building on the key finding that SP can accurately solve the CDM problem, we will explore a possible quantum algorithm to simulate its evolution.

Chapter 4

A Quantum Algorithm for Schrödinger-Poisson

In this chapter, we explore the challenges arising in the implementation of cosmological simulations on quantum devices. The dynamics is governed by the SP equation, where a self-gravitating potential introduces nonlinearities in the problem. The mapping of the nonlinear problem onto a quantum device is solved using a classical-hybrid variational algorithm similar to the one proposed in Lubasch et al. [23]. The evolution of the wavefunction is carried out using a variational time evolution (VTE) approach, tailored for nonlinear self-consistent problems defined on a grid, which allows for an exponential saving in computational memory resources through the encoding of N grid points in $\log_2(N)$ qubits. Building on [24], we adapt the VTE algorithm to the case where the potential is given by a variational ansatz, proposing quantum circuits for the evaluation of the required matrix elements whose depth scaling is polynomial with the number of qubits and the number of samples required for a desired accuracy scales polynomially with the number of grid points N.

We investigate the behavior of spatial resolution as the SP dynamics converges towards the classical limit ($\hbar/m \to 0$). Our investigation will unveil an empirical logarithmic correlation between the required number of qubits and the scale of the SP equation ($\hbar/m$).

This chapter is structured as follows. In Section 4.1 we describe the mapping of the cosmological SP equation on a quantum computer, including a discussion of the strategies that must be adopted in the latter for the description of non-linear problems.

Section 4.2 is devoted to the description of the VTE algorithm for self-consistent nonlinear problems, including a discussion on the quantum circuit implementation. Numerical simulations for a one dimensional 5-qubit (i.e., 32 grid points) system will be given in Section 4.3. The results include an analysis of the time-evolution obtained with different choices of physical parameters interpolating between the pure quantum regime and a classical, $\hbar/m \to 0$, limit. A study on the resolution convergence in this classical regime is also presented. Finally, we discuss the computational costs of our quantum algorithm and the conditions for potential quantum advantage. We draw our main conclusions in Section 4.4

4.1 The nonlinear SP equation on quantum computers

The correspondence between VP and SP, as treated in Sec. 1.2.3, opens up the possibility of using quantum algorithms (QA) for the investigation of dark matter dynamics, as it was already demonstrating that QA can reduce the scaling complexity for the solution of quantum mechanical problems in many-body physics and quantum chemistry [91]–[93]. We propose a scalable quantum algorithm for the simulation of the time propagation of non-linear Schrödinger-like equations of the form

$$i\frac{\partial}{\partial t}\Psi = H[\Psi]\,\Psi \tag{4.1}$$

where $H[\Psi]$ indicates the functional dependence of the Hamiltonian from the system wavefunction. We consider a complex wavefunction $\Psi(\mathbf{x}, t)$ (with $\mathbf{x} \in \mathbb{R}^3$) defined in such a way that $|\Psi|^2 = \rho/\rho^*$. The following normalization emerges naturally from the definition of the volume-mean density ρ^*

$$\frac{1}{\mathcal{V}}\int d\mathcal{V}\,|\Psi|^2 = 1, \tag{4.2}$$

The SP equation of interest (see diagram in Fig. 1.1) assumes the general form

$$i\frac{\partial}{\partial t}\Psi(\mathbf{x}, t) = \left(-\frac{\lambda}{2}\nabla^2 + \frac{1}{\lambda}V[\Psi(\mathbf{x}, t)]\right)\Psi(\mathbf{x}, t)\,. \tag{4.3}$$

with the self-interacting potential $V[\Psi]$ defined as

$$\nabla^2 V[\Psi] = \nabla^2 V(\mathbf{x}, t) = |\Psi(\mathbf{x}, t)|^2 - 1\,. \tag{4.4}$$

Here $\lambda = \hbar/m$ is the intrinsic scale of the problem and $V[\Psi]$ is a redefinition of the self interacting potential $U[\Psi]$ that renders the Poisson equation dimensionless. We use square brackets, e.g., $V[\Psi]$, to denote functional dependence. Details on how to recover Eqs. (4.3), (4.4) from Eq. (1.26) are given in Appendix A.1. This set of equations can be seen as a time-dependant Schrödinger-like equation (TDSE), where the self-interacting nature of the potential in Eq. (4.4) causes the dynamics of the system to be strongly nonlinear. It features two main processes, whose intensity are regulated by the magnitude of λ. We observe that if $\lambda \to \infty$ the potential term vanishes, leaving only the free Schrödinger equation which leads to diffusion [94] (however, due to the imaginary coefficient $i\lambda/2$, the Schrödinger equation cannot be strictly classified as a diffusion equation). In this case we expect to see a spatial smoothing of the density distribution. In the opposite limit, when $\lambda \to 0$, the potential term dominates: this should cause the collapse of the distribution followed by a series of peaks and caustics. As such, this can be seen as the onset of the classical regime of gravitational instability [53].

While quantum computation proved to be efficient in solving linear partial differential equations (PDEs) [95]–[97] problems arise when dealing with *nonlinear* equations due to the intrinsic *linearity* of quantum mechanics [18], [80], [91]. Two main challenges are associated with the non-linearity of Eq. (4.3). The first one is related to the fact that quantum states are usually prepared and evolved through unitary operations. This

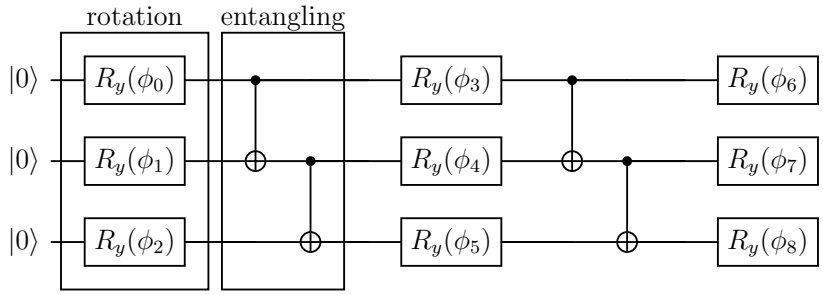


Figure 4.1: Example of a 3-qubit R_y-CNOT ansatz circuit used for the wavefunction $|\Phi_{\tilde{V}}\rangle$ used to evaluate the potential according to Eq. (4.13). This circuit has 3 rotational blocks U_{rot} and 2 entangling blocks U_{ent} with *linear* entanglement. The output function is parameterized through the nine real parameters $\boldsymbol{\theta}$ such that $U(\boldsymbol{\theta})\,|\mathbf{0}\rangle = |\Phi_{\tilde{V}}(\boldsymbol{\theta})\rangle$; in this case the number of parameters exceeds the Hilbert space dimension $2^3 = 8$.

preserves the well-known probability-like normalization of the quantum register: $\langle\psi\,|\psi\rangle = 1$. Thus, the *physical* wavefunction $|\Psi\rangle$, that solves Eq. (4.3), and the generic *quantum* state on the quantum register $|\psi\rangle$ live in two different Hilbert spaces. We will give more details on this subject in Section 4.1.1. The second complication arises from the self-consistency of the problem, which motivates the search for alternative time evolution algorithms beyond standard Trotter-based expansions [98], [99].

For context, the Trotter decomposition (or Trotter-Suzuki expansion) is a widely used method to approximate the time evolution of a quantum system. Suppose the Hamiltonian can be split into two non-commuting parts, $H = H_A + H_B$. The exact evolution operator over a small time step Δt is

$$U(\Delta t) = e^{-iH\Delta t}.$$

Since H_A and H_B do not commute, one cannot simply write $e^{-iH\Delta t} = e^{-iH_A\Delta t}e^{-iH_B\Delta t}$. The Trotter decomposition approximates the evolution as

$$e^{-iH\Delta t} \approx e^{-iH_A\Delta t}e^{-iH_B\Delta t} + \mathcal{O}(\Delta t^2),$$

becoming more accurate as Δt decreases. Higher-order Trotter-Suzuki formulas can further reduce the error, at the cost of deeper circuits [100].

While effective in many quantum simulations, Trotter methods face challenges when the Hamiltonian depends on the evolving state itself, as in self-consistent problems. In such cases, small approximation errors at each step can accumulate and feed back into the Hamiltonian, potentially leading to inaccurate or unstable evolution over long times. This motivates the development of alternative time evolution algorithms that can handle state-dependent Hamiltonians more robustly, ensuring stability and accuracy even in strongly self-consistent systems.

To address both issues, in this work we propose a variational time evolution algorithm specifically adapted to the nonlinearity of the problem, which relies on the development and the application of novel quantum circuits described in Sec. 4.2.

4.1.1 The quantum computing approach to the SP equation

A first attempt to solve the nonlinear SP equation was given by Mocz and Szasz [32]. Such a solution is fully variational and makes use of a finite difference optimization of the

potential and of the system wavefunction evaluated at two subsequent time steps. The variational nature of this approach also allows one to bypass the costly solution of the Poisson equation in Fourier space in favor of a variational optimization of the potential as implemented in a separate qubit register.

In this work, we propose a novel set of quantum circuits that enable the implementation of a different strategy based on an adapted variational time-dependent quantum algorithm for the propagation of the variational parameters defining the system wavefunction (See Section 4.1.1). This enables a more rigorous implementation of the wavefunction dynamics, avoiding the instabilities implicit in most VQE optimization procedures (e.g., slow convergence due to trapping in local minima and barren plateaus [101], [102]).

In particular, *barren plateaus* are regions of the variational parameter space where the gradients of the cost function with respect to the parameters ∇_θ vanish or become exponentially small as the system size increases. They typically arise in deep or highly entangled variational circuits, where the parameters are effectively randomized, causing the cost function landscape to become almost flat. In these regions, classical optimizers struggle to identify a meaningful update direction, leading to extremely slow convergence or even stalling entirely. The presence of barren plateaus is a fundamental limitation for many variational quantum algorithms, especially when scaling to larger systems, and motivates the search for alternative methods, such as the Variational Time Evolution (VTE) approach, that avoid reliance on gradient-based optimization over highly rugged or flat landscapes. On the other hand, the VTE algorithm comes at the cost of evaluating additional matrix elements for the solution of the equation of motion for the wavefunction parameters.

Grid-based representation of the system wavefunction

A typical space discretization associated to problems in first quantization [23], [24], [32], [98], [99] approximates the continuous space with a grid. In 1D, a line of length L is divided in arbitrary N equidistant points. For each grid point x_j we have $\Psi_j \simeq \Psi(x_j)$, with $j \in \{0, 1, ..., N-1\}$ and periodic boundary conditions $\Psi_N = \Psi_0$.
With a n-qubit quantum register, one can generate a quantum state $|\psi\rangle$ belonging to a N-dimensional Hilbert space, where $N = 2^n$. Making use of the *amplitude encoding*, only $n = \log_2 N$ qubits are needed to describe a N-point grid. A generic state $|\psi\rangle$ can hence be represented on a quantum register as a superposition of computational basis states,

$$|\psi\rangle = \sum_{j=0}^{N-1} \psi_j |\text{bin}(j)\rangle, \tag{4.5}$$

where $\text{bin}(j)$ is the binary representation of the grid position j and $\psi_j \in \mathbb{C}$ is the associated amplitude or weight, such that the probability distribution of measuring the different basis states (i.e., different positions on the grid) is normalized as $\langle\psi|\psi\rangle = \sum_{j=0}^{N-1} |\psi_j|^2 = 1$. By combining this relation with the discretization of Eq. (4.2), we can establish a correspondence between the approximated physical wavefunction on the grid point x_j and the corresponding coefficient of the j-th basis $|\text{bin}(j)\rangle$ in Eq. (4.5), such that $\Psi_j = \sqrt{N}\psi_j$.

The dynamics of the system wavefunction is described by means of a time-dependent variational approach [103]. To this end, we define a quantum trial state $|\psi(\boldsymbol{\theta}(t))\rangle$, parameterized by a set of (time-dependent) variables $\boldsymbol{\theta}(t) = \{\theta_1(t), ..., \theta_{M_p}(t)\}$, which evolve

Algorithm 1 alg: VTE for Self-consistent potential.

$i = 0$ ▷ Initialization
$\psi_{t_0} \leftarrow \boldsymbol{\theta}_{t_0}$
while $\{\boldsymbol{\phi}_{t_0}\}$ not converged **do**
  $V_{t_0} \leftarrow \boldsymbol{\phi}_{t_0}$
  cost $= ||V_{t_0} - |\Psi_{t_0}|^2 + 1||^2$
end while
for $i = 0, ..., N_t - 1$ **do**:
  evaluate $\mathrm{M}_{k,l}(\boldsymbol{\theta}_i), \mathrm{B}_k(\boldsymbol{\theta}_i, \boldsymbol{\phi}_{t_i})$ ▷ V.T.E
  $\boldsymbol{\theta}_{t_{i+1}} \leftarrow \mathrm{M} \cdot \dot{\boldsymbol{\theta}}_{t_i} = \mathrm{B}$
  $\psi_{t_{i+1}} \leftarrow \boldsymbol{\theta}_{t_{i+1}}$

  while $\{\boldsymbol{\phi}_{t_{i+1}}\}$ not converged **do** ▷ Pot. Opt.
    $V_{t_{i+1}} \leftarrow \boldsymbol{\phi}_{t_{i+1}}$
    cost $= ||V_{t_{i+1}} - |\Psi_{t_{i+1}}|^2 + 1||^2$
  end while
end for

according to well-defined equations of motion [103]. The initial state is prepared through a suitable choice of a parameterized unitary (quantum circuit) $U(\boldsymbol{\theta}(0))$. An explicit circuit example is shown in Fig. 4.2. Using the previous relation between Ψ_j and ψ_j, we can describe the time evolution of the physical state

$$|\Psi(\boldsymbol{\theta}(t))\rangle = \sqrt{N}\,|\psi(\boldsymbol{\theta}(t))\rangle\,. \tag{4.6}$$

using the updated parameters $\boldsymbol{\theta}(t)$.

Variational time propagation with nonlinearities

The trial wavefunction $|\psi(\boldsymbol{\theta}(t))\rangle$ is evolved adapting the VTE algorithm proposed in Ref. [24] to the case where the potential is self-consistent with the wavefunction and needs to be re-evaluated at each timestep. In VTE, the dynamics is tracked on the manifold spanned by the time-dependent parameters $\boldsymbol{\theta}(t)$ used to describe the trial wavefunction.

For a system evolving under the action of a Hamiltonian $\mathcal{H}$, we derive, from the McLachlan variational principle [24], [103], a set of equations of motion (EOM) of the form

$$\mathrm{M}\,\dot{\boldsymbol{\theta}} = \mathrm{B}, \tag{4.7}$$

where

$$\mathrm{M}_{kl} = \mathrm{Re}\left\{\langle \partial_{\theta_k}\Psi \mid \partial_{\theta_l}\Psi\rangle - \langle \partial_{\theta_k}\Psi \mid \Psi\rangle\langle \Psi \mid \partial_{\theta_l}\Psi\rangle\right\} \tag{4.8}$$

$$\mathrm{B}_k = \mathrm{Im}\left\{\langle \partial_{\theta_k}\Psi|\mathcal{H}|\Psi\rangle - \langle \partial_{\theta_k}\Psi \mid \Psi\rangle\langle \Psi|\mathcal{H}|\Psi\rangle\right\} \tag{4.9}$$

with

$$\mathcal{H}[\Psi] = \left(-\frac{\lambda}{2}\nabla^2 + \frac{1}{\lambda}V[\Psi]\right) \tag{4.10}$$

as defined in Eq. (4.3). The dependence of Ψ on the parameters $\boldsymbol{\theta}(t)$ is implicit. Note that to capture the exact evolution comprehensive of nonlinear effects, the terms in Eqs. (4.8)

and (4.9) are rescaled according to Eq. (4.6). The main obstacle to the application of such method is the evaluation of the term $\mathrm{Im}\,\langle\partial_{\theta_k}\Psi|\mathcal{H}|\Psi\rangle$ in Eq. (4.9). The difficulty stands in the fact that the self-consistent potential does not have a standard form, but it depends on the system wavefunction. The evaluation of this term is made possible by application of the novel quantum circuit scheme discussed in Sec. 4.2.

Optimization of the potential

As anticipated in Sec. 4.1, the functional dependence of the potential on the system wavefunction, Ψ, brings a further level of complexity into the dynamics of the system. While classically, the solution of the Poisson equation (4.4) for a generic wavefunction Ψ can easily be found using a spectral method in Fourier space [32], such strategy is not practical on near-term quantum computers, as it would require rather deep circuits [104]. We instead resort to a variational approach. Hence, we introduce a second set of parameters $\boldsymbol{\phi}(t) = \{\phi_V(t), \tilde{\boldsymbol{\phi}}(t)\} = \{\phi_V(t), \tilde{\phi}_1(t), ..., \tilde{\phi}_{M_p}(t)\}$, with M_p being the number of variational parameters. The parameters with the tilde are used in the preparation of the trial wavefunction and are in fact angles of the rotations, as shown in Fig. 4.1. The parameter ϕ_v is used to encode the norm of the potential, that due to the nonlinear nature of the problem is expected to change. This set of variational parameters allows for the description of the quantum state

$$|\Phi_V(\boldsymbol{\phi})\rangle = \phi_V \left|\Phi_{\tilde{V}}(\tilde{\boldsymbol{\phi}})\right\rangle . \tag{4.11}$$

The potential can be obtained as

$$|\Phi_V(\boldsymbol{\phi})\rangle = \sum_{j=0}^{N-1} V_j(\phi(t))\,|j\rangle = \phi_V \sum_{j=0}^{N-1} \tilde{V}_j(\phi(t))\,|j\rangle \ , \tag{4.12}$$

where the index j in $V_j(\phi(t))$ labels the grid position $\mathbf{x}_j$ associated to the bit string $\mathrm{bin}(j)$. In Eq. (4.12) the parameter ϕ_V [32] ensures the normalization of the potential,

$$\left\langle \Phi_{\tilde{V}}(\tilde{\phi})|\Phi_{\tilde{V}}(\tilde{\phi})\right|\left.\Phi_{\tilde{V}}(\tilde{\phi})|\Phi_{\tilde{V}}(\tilde{\phi})\right\rangle = \sum_{j=0}^{N-1} |\tilde{V}_j(\phi(t))|^2 = 1\,,\ \forall t\,. \tag{4.13}$$

The potential can therefore be interpreted as a functional of the circuit parameters, $V_j(\boldsymbol{\phi}(t))$. The parameters are iteratively updated to minimize the distance between the parameterized potential and the one arising from Eq. (4.4):

$$min_{\boldsymbol{\phi}} \left(\sum_{j=0}^{N-1} \left(\nabla^2 V_j(\boldsymbol{\phi}) - |\Psi_j(\boldsymbol{\theta})|^2 + 1 \right)^2 \right) . \tag{4.14}$$

Details about the terms appearing in Eq. (4.14) are given in Appendix A. When the optimization converges, the function $V_j(\boldsymbol{\phi}(t))$ approximates the exact potential $V(\mathbf{x}, t)$. As the criterion expressed in Eq. (4.14) is equivalent to finding the set of parameters that minimizes the distance between the trial state and the potential obtained using the right hand-side of Eq. (4.4) $||V(\boldsymbol{\phi}) - |\Psi(\boldsymbol{\theta})|^2 - \mathbf{1}||$. In practical implementations, the convergence criterion is defined by the optimization strategy of choice. In this work we employed gradient based methods, that, in general, stop when the gradient of the cost function is less than a given threshold.

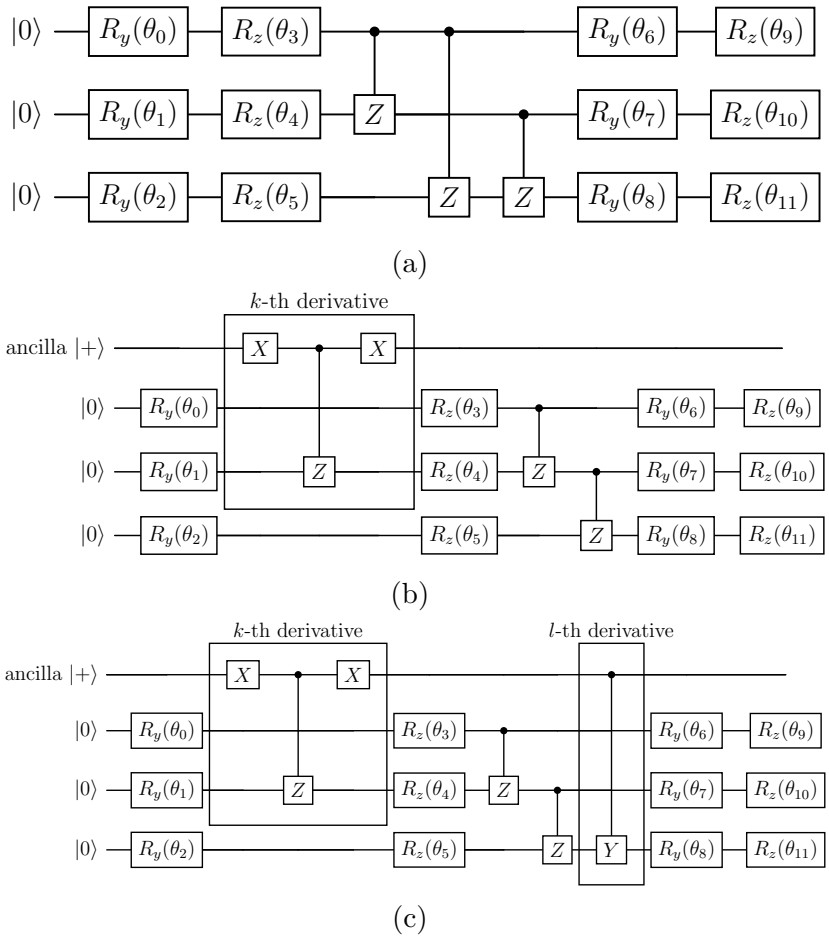


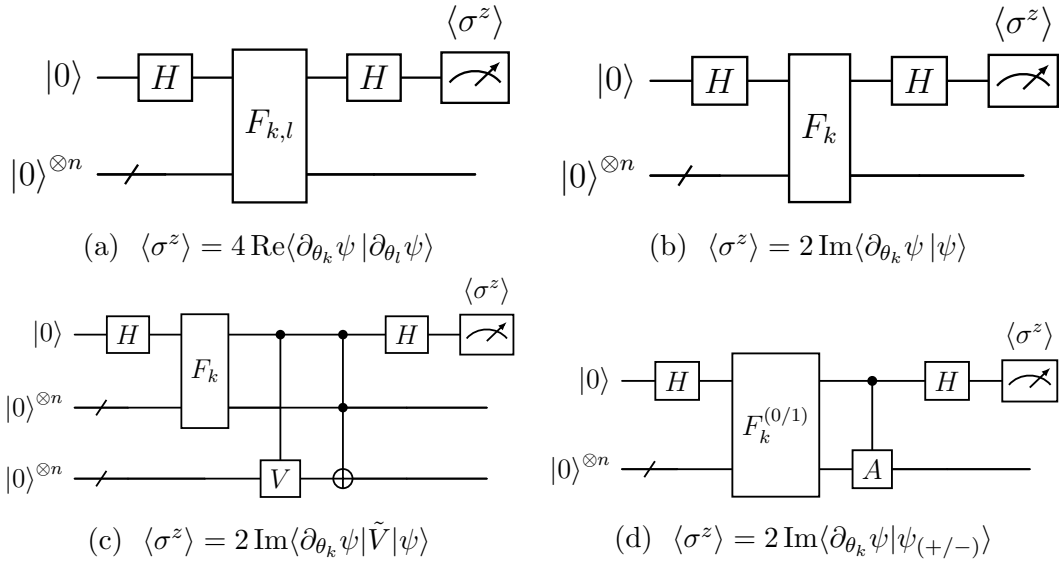
Figure 4.2: Quantum circuits used to prepare: (a) the trial wavefunction; (b) the unitary matrix F_k that generates states like the one in Eq (4.19); (c) the unitary matrix $F_{k,l}$ that generates states like the one in Eq. (4.20).

4.2 The Algorithm

The problem of self-consistency is solved, as anticipated in Section 4.1, by alternating the solution of the TDSE (VTE) and the optimization of the potential ($Pot.Opt$). The intrinsic nonlinear nature of the SP equation is reconciled with the requirements of a quantum circuit implementation imposing the correct normalization of the *physical* wavefunction and potential, as given by Eq. (4.6) and Eq. (4.12)), respectively. A scheme of this algorithm is reported in Alg. 1, where $\{\boldsymbol{\theta}_{t_i}\}$ and $\{\boldsymbol{\phi}_{t_i}\}$ refer to the parameters' set at time t_i; $i \in \{0, 1, ..., N_t - 1\}$. For conciseness, in Alg. 1 we use the following notation $\Psi_i \equiv \Psi(\boldsymbol{\theta}_{t_i})$.

4.2.1 Circuit implementation

The trial quantum states for both the wavefunction and the potential are implemented using a heuristic local ansatz [19], [23], [24], [32], [99], [105] that alternates single qubit



(a) $\langle\sigma^z\rangle = 4\,\mathrm{Re}\langle\partial_{\theta_k}\psi\,|\partial_{\theta_l}\psi\rangle$

(b) $\langle\sigma^z\rangle = 2\,\mathrm{Im}\langle\partial_{\theta_k}\psi\,|\psi\rangle$

(c) $\langle\sigma^z\rangle = 2\,\mathrm{Im}\langle\partial_{\theta_k}\psi|\tilde{V}|\psi\rangle$

(d) $\langle\sigma^z\rangle = 2\,\mathrm{Im}\langle\partial_{\theta_k}\psi|\psi_{(+/-)}\rangle$

Figure 4.3: Quantum circuits for the evaluation of the VTE matrix elements in Eq. (4.8), (4.9) through the measurement of the ancilla qubits $\langle\sigma_z\rangle$. The correspondence between expectation value and measurement is reported under the respective scheme. The unitaries $F_{k,l}$ and F_k are both reported in Fig. 4.2. They are used to produce the wavefunction and its derivatives. The Toffoli gate in panel (c) represents a Toffoli ladder: n Toffoli gates linking the wavefunction and the potential register qubit per qubit. (d) $F_k^{(1/0)}$ denotes F_k with different control states ($|0\rangle$ or $|1\rangle$) and A is the adder circuit [23] (See Appendix A.3 for more details on the adder circuit).

rotation layers $U^{rot}(\boldsymbol{\theta})$ and entangling layers U^{ent} (see example in Fig. 4.1)

$$U(\boldsymbol{\theta}) = U_0^{rot}(\boldsymbol{\theta}_0) \cdot \prod_{\xi=1}^{D} U_\xi^{ent} \cdot U_\xi^{rot}(\boldsymbol{\theta}_\xi) \tag{4.15}$$

where D is the number of entangling layers and $\boldsymbol{\theta}_\xi$ a subgroup of parameters. In Fig 4.2, we show the typical circuits used to encode the wavefunction $|\psi(\boldsymbol{\theta})\rangle$, while Fig. 4.1 reports the one used for the potential $|\Phi_{\tilde{V}}(\boldsymbol{\phi})\rangle$. The latter consists of just $\mathrm{R}_Y(\theta)$ rotations and CX gates, since the target potential function is real-valued.

The quantum part of the evolution algorithm resides in the measurement of the expectation values in Eqs. (4.8) and (4.9). In the following, we propose an efficient implementation of the circuits for the evaluation of the terms with derivatives in Eqs. (4.8) and (4.9). In particular, we provide a detailed procedure for the calculation of those matrix elements that have a functional dependence on the non-linear potential, such as $\langle\partial_{\theta_k}\psi|\,\mathcal{H}(V(\boldsymbol{\phi}))\,|\psi\rangle$.

Given the structure of the ansatz in Eq. (4.15) and θ_k in the subset $\boldsymbol{\theta}_{\tilde{\xi}}$, the derivatives ∂_{θ_k} leaves the unitary unchanged, with the exception of the target rotational layer:

$$U_{\tilde{\xi}}^{rot}(\boldsymbol{\theta}_{\tilde{\xi}}) = \bigotimes_{j=0}^{n-1} \exp\{-\frac{i}{2}\alpha_j\theta_{\tilde{\xi},j}\} \tag{4.16}$$

where $\theta_{\tilde{\xi},j} \in \boldsymbol{\theta}_{\tilde{\xi}}$ and $\alpha_j \in \{X, Y, Z\}$ is a Pauli matrix, generator of single qubit rotations. Combining Eqs. (4.15), (4.16) and $|\psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})\,|\boldsymbol{\Xi}\rangle$, one gets for the partial derivative

∂_k

$$\partial_{\theta_k} U(\boldsymbol{\theta}) \left|\Xi\right\rangle = \left|\partial_{\theta_k}\psi(\boldsymbol{\theta})\right\rangle = -\frac{i}{2} W_k(\boldsymbol{\theta}) \left|\Xi\right\rangle \ , \tag{4.17}$$

for a generic quantum state $|\boldsymbol{\Xi}\rangle$. Here $W_k(\boldsymbol{\theta})$ is a modified version of $U(\boldsymbol{\theta})$ where the single qubit rotation $R_\alpha(\theta_k)$ is preceded by its own generator [24], [106].
In the search for an efficient quantum circuit able to reproduce the matrix and vector elements of the McLachlan equation of motion of Eq. (4.7), the main obstacle is to produce a quantum state with the following structure:

$$\left|\psi\right\rangle = \frac{1}{\sqrt{2}} \left(U_1(\boldsymbol{\theta}) \left|\boldsymbol{\Xi}\right\rangle \left|0\right\rangle + U_2(\boldsymbol{\theta}) \left|\boldsymbol{\Xi}\right\rangle \left|1\right\rangle\right) \ , \tag{4.18}$$

where U_1, U_2 are generic unitaries and the second quantum register (single qubit) is used to evaluate the value of the matrix element. In the specific case at study, these unitaries should be expressive enough to enable a suitable parameterization of the wavefunction and its derivatives (Eq. (4.17)). Given the structure of the circuit W_k, by controlling only the Pauli matrix that implements the derivative, it is possible to prepare the quantum states

$$\begin{aligned} F_k(\boldsymbol{\theta}) \left|\boldsymbol{\Xi}\right\rangle \left|+\right\rangle &= \frac{1}{\sqrt{2}} \left(W_k(\boldsymbol{\theta}) \left|\boldsymbol{\Xi}\right\rangle \left|0\right\rangle + U(\boldsymbol{\theta}) \left|\boldsymbol{\Xi}\right\rangle \left|1\right\rangle\right) \\ &= \frac{1}{\sqrt{2}} \left(2i \left|\partial_{\theta_k}\psi(\boldsymbol{\theta})\right\rangle \left|0\right\rangle + \left|\psi(\boldsymbol{\theta})\right\rangle \left|1\right\rangle\right), \end{aligned} \tag{4.19}$$

$$\begin{aligned} F_{k,l}(\boldsymbol{\theta}) \left|\boldsymbol{\Xi}\right\rangle \left|+\right\rangle &= \frac{1}{\sqrt{2}} \left(W_k(\boldsymbol{\theta}) \left|\boldsymbol{\Xi}\right\rangle \left|0\right\rangle + W_l(\boldsymbol{\theta}) \left|\boldsymbol{\Xi}\right\rangle \left|1\right\rangle\right) \\ &= i\sqrt{2}\Big(\left|\partial_{\theta_k}\psi(\boldsymbol{\theta})\right\rangle \left|0\right\rangle + \left|\partial_{\theta_l}\psi(\boldsymbol{\theta})\right\rangle \left|1\right\rangle \Big), \end{aligned} \tag{4.20}$$

for a given reference state $|\boldsymbol{\Xi}\rangle$ where $F_{k,l}(\boldsymbol{\theta})$ and $F_k(\boldsymbol{\theta})$ refer to unitaries for the different derivatives (see Fig. 4.2).

Fig. 4.3 summarizes all quantum circuits relevant for the evaluation of the terms in Eq. (4.8), (4.9). A brief discussion on how to evaluate them on a QC will follow, starting with the overlaps $\mathrm{Im}\,\langle\partial_{\theta_k}\psi|\,\psi\rangle$ and $\mathrm{Re}\,\langle\partial_{\theta_k}\psi|\,\partial_{\theta_j}\psi\rangle$. One can notice from Eqs.(4.19), (4.20) that upon applying a H gate, measuring $\langle\langle\sigma_z\rangle\rangle$ on the ancillary qubit returns the desired quantities. Furthermore, there is no need to evaluate the real part to compute the product of the overlaps in Eq. (4.8) since the term $\langle\partial_{\theta_k}\psi|\,\psi\rangle$ is purely imaginary. The circuits used to do so are shown in Figs. 4.3a, 4.3b.

The potential part $\mathrm{Im}\left\langle\partial_{\theta_k}\psi|\tilde{V}(\tilde{\boldsymbol{\phi}})|\psi\right|\left.\partial_{\theta_k}\psi|\tilde{V}(\tilde{\boldsymbol{\phi}})|\psi\right\rangle$, is what actually connects the solution of the $TDSE$ and the Poisson equation. $\tilde{V}(\tilde{\boldsymbol{\phi}})$ is given in Eq. (4.12) and is prepared using the parameters $\tilde{\boldsymbol{\phi}}$ resulting from the minimization of Eq. (4.14). The circuit in Fig. 4.3c is the one used for the evaluation of this linking term, where the series of n Toffoli gates provides a pointwise multiplication between the wavefunction and the potential registers (i.e., $\sum_k \tilde{V}_k\psi_k$).

Concerning the term $\mathrm{Im}\left\{\langle\partial_{\theta_k}\psi|\,\nabla^2\,|\psi\rangle\right\}$, a few considerations are needed. For systems of cosmological relevance, we expect accurate simulations to require a fine enough spatial resolution to resolve all spatial features. Therefore, using a finite differences approach, as also proposed in Ref. [23], can be justified as the discretization error should be irrelevant

Table 4.1: State fidelity $\mathcal{F}$ (between the classical reference and the evolved state at $t = 3$) for different VTE simulations. Hyperparameters: D_ψ and D_V, number of rotation layers in the wavefunction and potential ansatz respectively; M_p total number of parameters in the wavefunction ansatz; N_t number of timestep used in the simulation; r_c cutoff for singular values, used determine the effective rank of the matrix M in Eq. (4.7) (more information available in `Scipy` documentation [107]); ϵ regularization factor added to the diagonal of the matrix M in Eq. (4.8).

	D_ψ	D_V	M_p	N_t	r_c	ϵ	$\mathcal{F}$
4-qubits	4	4	32	$6 \cdot 10^2$	10^{-7}	10^{-3}	0.976
5-qubits	5	6	50	$9 \cdot 10^3$	10^{-8}	10^{-4}	0.944
5-qubits	5	6	50	$2 \cdot 10^4$	10^{-8}	10^{-4}	0.960
5-qubits	6	6	60	$6 \cdot 10^3$	10^{-8}	10^{-4}	0.956

at higher resolutions. In this framework, an approximation of the Laplace operator is given by

$$\mathrm{Im}\left\{\langle\partial_{\theta_k}\psi|\,\nabla^2\,|\psi\rangle\right\} = \frac{1}{\Delta x^2}\,\mathrm{Im}\left\{\langle\partial_{\theta_k}\psi|\,\psi_+\rangle - 2\,\langle\partial_{\theta_k}\psi|\,\psi\rangle + \langle\partial_{\theta_k}\psi|\,\psi_-\rangle\right\}\,, \tag{4.21}$$

with the positive (and negative) shifted wavefunctions $|\psi_\pm\rangle = \sum_{j=0}^{N-1}\psi_{j\pm1}\,|\mathrm{bin}(j)\rangle$, obtained using the adder circuit A [23], whose action on the j-th base is $|\mathrm{bin}(j)\rangle \mapsto |\mathrm{bin}(j-1)\rangle$, in combination with the unitary $F_k(\boldsymbol{\theta})$ of Eq. (4.19) with different control state allows to evaluate the shifted overlaps in Eq. (4.21). A scheme of the circuits needed to perform these operations is presented in Fig. 4.3d.

See Appendix A.4 for more details about the functioning the circuits in Figs. 4.3d, 4.3c.

4.3 Results and Discussion

Before addressing the setup used in our simulation, some consideration about the characteristic scales appearing in the SP equation and the corresponding units are needed.

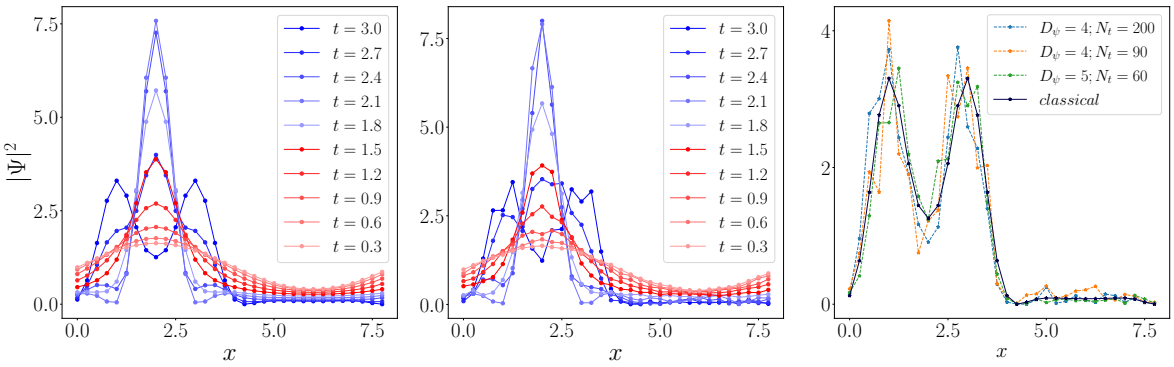
Given the invariance of the SP Eqs (4.3), (4.4) under the following scaling transformation:

$$\{x, t, \psi, \lambda\} \mapsto \{\alpha x, \beta t, \beta^{-1}\psi, \alpha^{-2}\beta\lambda\} \tag{4.22}$$

λ emerges as an intrinsic scale of the problem [53] as its scaling law combines changes in both the spatial and time domain (i,e system with different box dimension or evolution time will display different dynamics).

Concerning the dimension of the physical quantities appearing in the problem, we used arbitrary unit. See Appendix A.1 for details on the arbitrary values chosen for the density normalization ρ^* and the constant G in the transition from Eq. (1.26) to Eqs.(4.3), (4.4).

As a final remark, we would like to emphasize that in this preliminary study all simulations were performed in an idealized setting, without considering gate errors and sampling shot noise.



(a) (b) (c)

Figure 4.4: Comparison between probability distributions at different times for a 5 qubits system and $\lambda = 1$. The left panel (a) is the classical reference, obtained with a spectral method [32]. In the middle one (b) are presented the results obtained through a VTE simulation (using the algorithm in Fig. 1). In the left (c) we compare the classical probability distribution at $t = 3$ with the results obtained from the VTE simulations with different hyperparameters (more details in Tab. 4.1) . The one chosen for the simulation in the middle panel (b) are $N_t = 2 \cdot 10^4$, $\epsilon = 10^{-4}$, $r_c = 10^{-8}$, $D_\psi = 6, D_V = 6$.
.

4.3.1 Numerical simulations

We consider a one dimensional system of length $L = 8$ with periodic boundary conditions. As anticipated above, we use arbitrary units for both spatial coordinates and time variable. The choice of L and the total time of the simulation is done in such a way that, once we fix $\lambda = 1$, the self interacting potential Eq. (4.4) exactly balances the diffusion associated with the Schrödinger time-evolution. In order to compare our results with those from Ref. [32], we used as initial condition a sinusoidal distribution of the form

$$\Psi(x, 0) = \sqrt{1 + 0.6 \sin\left(\frac{\pi}{4} x\right)}, \tag{4.23}$$

evolved according to Eqs. (4.3) and (4.4). This specific initial condition is a well-known standard test case. It corresponds to one of the different Fourier components typically found in initial distributions for the VP equations, like Gaussian random fields. It is widely used as it makes it easy to observe the effects of shell-crossing.

For this proof-of-principle numerical implementation, the parameters $\boldsymbol{\theta_0}$ reproducing the initial quantum state are obtained by optimizing the state fidelity $\mathcal{F}(\psi(\boldsymbol{\theta}), \tilde{\psi})$ between the variational trial state $|\psi(\boldsymbol{\theta})\rangle$ and a target state $\left|\tilde{\psi}\right\rangle$. In this work we refer to $\mathcal{F}$ as the state fidelity between two *quantum* states [108] (i.e., state normalization is 1). In the situation where $|\psi_1\rangle$ and $|\psi_2\rangle$ are pure states, we have $\mathcal{F}(\psi_1, \psi_2) = |\langle\psi_1|\psi_2\rangle|^2$. This value will be also used to measure the convergence of the states obtained with the variational method to the ones obtained classically. We point out that this has noting to do with the convergence to the actual solution of the physical problem (i.e., does not take into consideration the grid discretization error). The classical optimization of the potential (*Pot. Opt.* in Algorithm 1) is performed using a combination of `COBYLA` (to start

the optimization) and `BFGS` (to find the best solution) algorithms as implemented in `SCIPY` v1.9.0. All simulations were performed in Qiskit [109] (version 1.3.0), within the `statevector` framework, i.e., using a matrix representation of the quantum circuit and a vector representation of the quantum state.

The equations of motion in Eq. (4.7) are integrated using an explicit Euler method with fixed timestep for a total of N_t steps. Here, it is important to mention that, in general, the inversion of the matrix M in Eq. (4.8) may become ill-defined. In fact, the equation can be rewritten in the form

$$\mathrm{M}_{ij} = \langle \partial_{\theta_i} \Psi | \left(\mathbb{I} - |\Psi\rangle \langle \Psi| \right) \left| \partial_{\theta_j} \Psi \right\rangle = \langle \partial_{\theta_i} \Psi | \, \Pi_{\Psi}^{\perp} \left| \partial_{\theta_j} \Psi \right\rangle \tag{4.24}$$

where $\Pi_{\Psi}^{\perp}$ is the projector onto the space orthogonal to that spanned by Ψ. The matrix M can be interpreted as the expectation value of a projector onto the subspace generated by the derivatives of Ψ with respect to its variational parameters. By definition, the projection of a vector onto a subspace yields zero components for directions orthogonal to that subspace. For instance, if $\langle \partial_{\theta_i} \Psi \, | \Psi \rangle = 1$, the derivative respect to θ_i is parallel to $|\Psi\rangle$ and the corresponding row of M will be entirely zero. Consequently, M is likely to contain zero rows or columns, leading to an ill-posed problem. In this context, we refer to the matrix as ill-defined.

To reduce the resulting instabilities of the dynamics, we used the `SCIPY` least squares solver [107] with a suitable choice of the corresponding hyperparameters: the cutoff r_c, used to determine the effective rank of the matrix in Eq. (4.7) such that the singular values smaller than $r_c \cdot \Lambda_{max}$ are set to zero (here Λ_{max} is the singular value of largest magnitude), and the regularization factor ϵ, applied to the diagonal of the matrix M in Eq. (4.8).

In order to determine the quality of the results, we should also consider the level of expressivity of the variational ansatz, which is used to encode both the system wavefunction and the potential. Here, expressivity refers to the ability of the ansatz to represent a wide range of quantum states, essentially measuring how well it can approximate the target state. A more expressive ansatz can, in principle, capture more complex features of the system, but may require more parameters and a more challenging optimization. To achieve accurate results, one would, in principle, need a number of circuit parameters $\boldsymbol{\theta}(t)$ for the wavefunction that approaches the dimension of the Hilbert space. On the other hand, the number of terms in the matrices and vectors used in the equations of motion, Eqs. (4.8) and (4.9), scale as M_p^2 and M_p, respectively, as shown in Tab. 4.2, where M_p is the number of parameters. Reducing the number of parameters significantly reduces the total number of circuit evaluations. This, however, translates in a lower accuracy of the dynamics, as the ansatz may not enable a thorough description of the sector of interest of the full Hilbert space. Similarly, a large number of parameters will enable a more accurate description of the self-consistent potential, at the price of a more cumbersome (classical) optimization process and an increased circuit depth.

To assess the quality of our implementation (including the adjustment of the hyperparameters), we performed two series of simulations. The first is a *classical spectral method* based on FFT as in [32]. Results obtained from this approach will be used as a reference. The actual implementation of our proposed quantum algorithm consists, instead, of repeated cycles of circuit optimization and VTE steps (Algorithm 1). When comparing its outcomes with the exact ones (Fig. 4.4 and Tab. 4.1) we observe that the quantum

Table 4.2: Number of different circuits used to evaluate the terms in Eq. (4.7) with the respective number of qubits needed for the implementation. Here M_p is the number of variational parameter in the wavefunction ansatz and $n = log_2 N$ the number of qubits used for the discretization.

Term	No. circuits	No. qubits
$\mathrm{Re}\,\langle\partial_{\theta_k}\psi\|\partial_{\theta_l}\psi\|\partial_{\theta_k}\psi\|\partial_{\theta_l}\psi\rangle$	$M_p(M_p-1)/2$	$n+1$
$\mathrm{Im}\,\langle\partial_{\theta_k}\psi\|\psi\|\partial_{\theta_k}\psi\|\psi\rangle$	M_p	$n+1$
$\mathrm{Im}\left\langle\partial_{\theta_k}\psi\|\tilde{V}\|\psi\right\rangle\left\langle\partial_{\theta_k}\psi\|\tilde{V}\|\psi\right\rangle$	M_p	$2n+1$
$\mathrm{Im}\,\langle\partial_{\theta_k}\psi\|\psi_\pm\|\partial_{\theta_k}\psi\|\psi_\pm\rangle$	$2M_p$	$2n-1$

approach rightfully captures the qualitative behavior of the wavefunction, although the probability distribution obtained from the VTE is not as smooth as the exact one. Most notably, the VTE simulation does not preserve the symmetry of the solution. This behavior is intrinsic to the variational approach: the time evolution is transferred from the actual wavefunction to a set of variational parameters. Since no physical constraints are imposed on these parameters' manifold to enforce the symmetry of the wavefunction, they evolve according to Eqs. (4.8)–(4.9), oblivious of such constraint. This ultimately leads to the loss of symmetry of the wavefunction. In order to keep the symmetry of the system with respect to the peak, this information should have been encoded in the variational ansatz of both, potential and wavefunction. However, we have chosen a generic variational ansatz, without imposing reflection symmetry, in order to avoid bias in the evolution toward a specific profile and to allow the wavefunction to develop according to the initial condition itself.

4.3.2 Interpretation of the SP results

Fig. 4.5 shows the time evolution of the initial sinusoidal distribution, as given in Eq. (4.23), over a time span of approximately 6 time units for two different choices of the parameter λ (left: $\lambda = 1$, right: $\lambda = 0.25$). The lower panels depict the same dynamics as a two-dimensional surface plot of the time dependent wavefunction. The higher the value of λ, the larger the quantum nature of the dynamics; in fact, in the limit of $\lambda \to 0$, the SP dynamics converges towards the classical VP dynamics [53]. Physically, the collapse and splitting of the probability distribution (left panels in Fig. 4.5) is an effect of the self-interacting potential. This is regulated by the scale of the problem λ. However, as stated in the preamble of Sect. 4.3, what really matters is not the absolute value of λ, but its value relative to the box size and time (e.g, if instead of $L = 8$ we had $L = 1$, we would need to change λ to $\lambda/64$, accordingly). In the classical limit $\hbar/m \to 0$, the quantum effects are suppressed, the potential cannot counter the diffusion anymore, and secondary peaks arise, as in the classical VP solution. In this scenario, the effects of shell-crossing are more pronounced. An example is shown in the right column of Fig. 4.5. Starting from the initial sinusoidal distribution, the gravitational attraction induces the concentration of matter density in a first peak (around time $t = 3$), which then collapses due to the

effect of gravity damping. This process repeats few more times, giving rise to a multitude of sub-peaks as a result of repeated episodes of shell-crossing.

4.3.3 Scaling of required resources

The largest cosmological simulations describe nowadays the evolution of boxes having a size of several Gigaparsecs, and using of the order of a trillion resolution elements (particles) [110]. Although simulations of this size are beyond the reach of what can be achieved on current quantum computers, the possibility of efficiently running large suites of simulations with $\sim 10^{10}$ particles each is still highly valuable for carrying out a number of useful calibrations of observational quantities and exploring the parameter space of cosmological models [111], [112]. We thus consider a situation of possible cosmological interest to be a 3D simulation with resolution in grid points per dimension of $2048 = 2^{11}$. Thanks to the logarithmic encoding, a total of 2^{33} grid points can be obtained with $n_{tot} = 33$ qubits. In Tab. 4.2 we report the number of qubits needed for every term of Eq. (4.7) and the relative number of different circuits used. In this exploratory work, we used a heuristic number of parameters M_p and timesteps N_t for our simulation. Thus we are not in a position to provide an accurate estimate of the number of parameters, or timesteps, required for a relevant cosmological simulation. What we can say is that such simulation would require a maximum of $2n + 1$ qubits, used in the evaluation of the potential term.

However, both PEC and PEA entail a substantial sampling overhead. In PEC, the number of measurements required scales as

$$N_{\text{meas}} \sim \frac{\gamma^2}{\epsilon^2}, \tag{4.25}$$

where γ is the noise amplification factor, typically growing exponentially with circuit depth, and ϵ is the desired statistical error. Similarly, PEA relies on repeated executions at amplified noise levels, leading to a polynomial or exponential increase in N_{meas} depending on the noise model.

From Table 4.1 we can retrieve some useful insights about the required timestep (to ensure numerical stability) and the scaling of the target error with the system size. It is worth mentioning that the following outtakes regard the scenario in which the EOM (4.7) is integrated by an explicit first order Euler method.
Firstly, we note that to precisely describe the full Hilbert space, the number of parameters M_p should increase by a factor of two with the addition of each qubit. Furthermore, increasing spatial resolution (number of qubits) necessitates a higher number of timesteps to maintain the desired level of accuracy in describing the dynamics. This phenomenon is analogous to what occurs in classical numerical integration problems, such as spectral methods or N-body simulations. On the other hand, when the fidelity $\mathcal{F}$ is kept constant, the expected number of timesteps N_t decreases as the number of variational parameters M_p increases. This trend can be attributed to the fact that the equation being integrated (Eq. (4.7)) operates within parameter space, whereas the original dynamics (i.e., the Hamiltonian in Eq. (4.10)) is only reflected in the vector term (as per Eq. (4.9)). Moreover, the variational approach enables the use of a parameter count smaller than the Hilbert space dimension. Consequently, capturing the same dynamics within a sub-

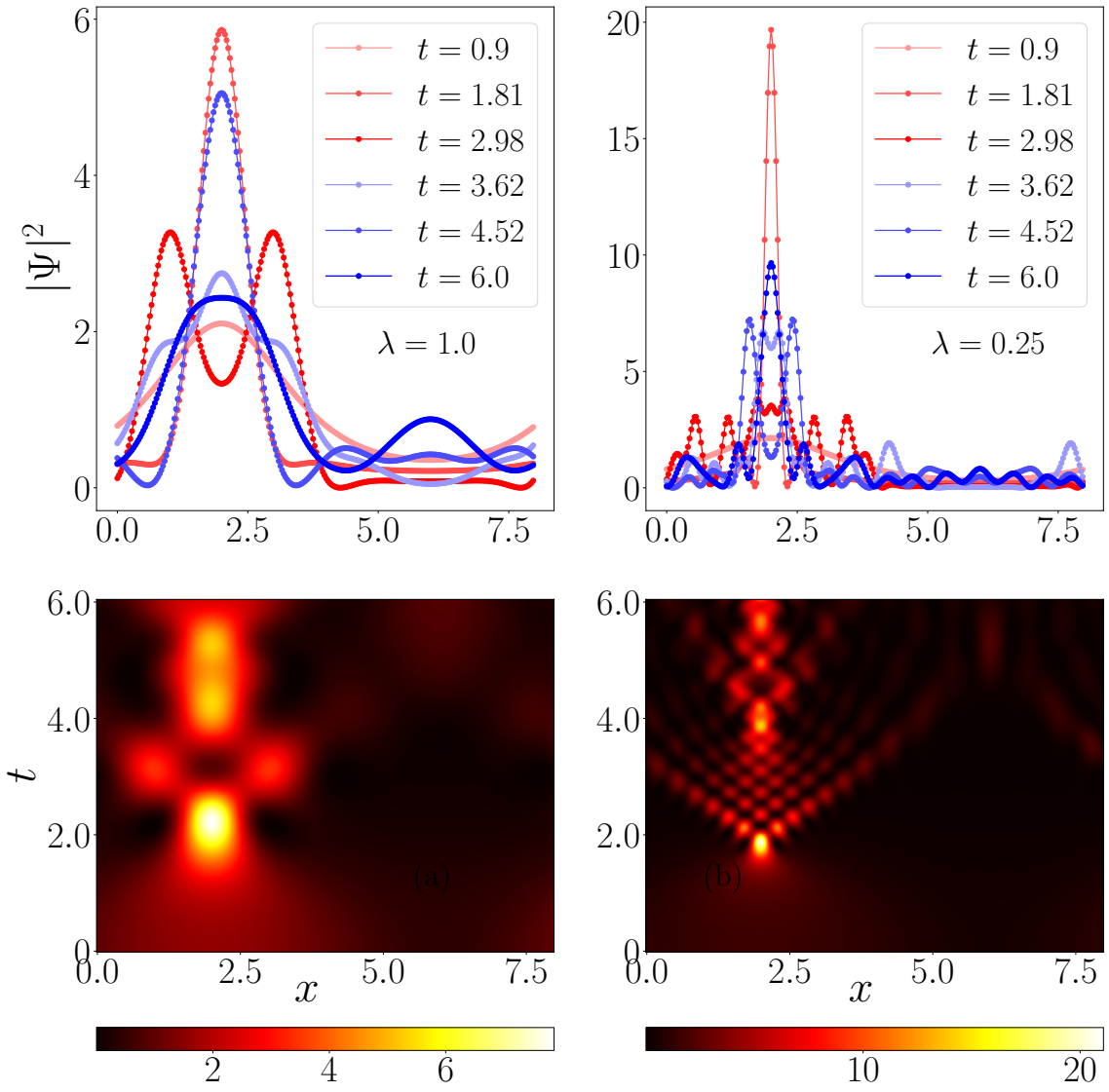


Figure 4.5: Classical evolution of the 1-D probability distribution under the effect of the gravitational potential, for different values of λ. Both simulations have been carried out with a spectral method [32]. In the top row, probability distributions are shown at fixed time frames. In the bottom row the same evolution is shown in 2-D perspective by a heatplot: the x axis represents the spatial coordinate, while the y axis is used for the time; the probability distribution magnitude is represented through a color gradient. The difference between these two simulations is given by the intensity of the quantum pressure term. In the left column (a) $\lambda = 1$ and the quantum effect balances the diffusion; In the right column (b), with $\lambda = 1/4$ the dynamics is similar to the classical one (VP).

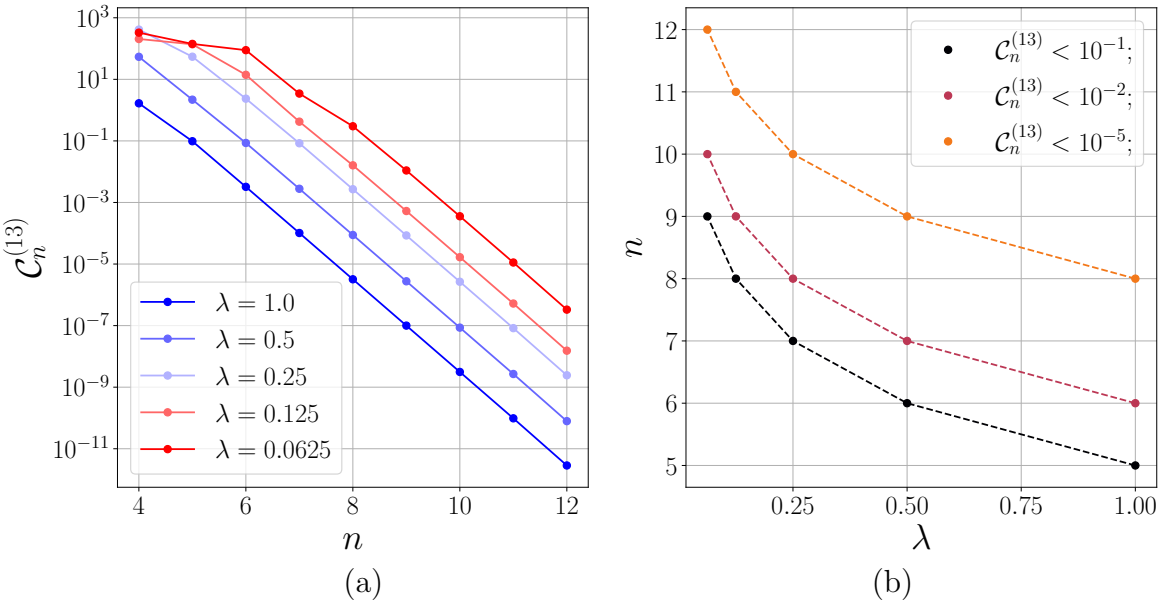


Figure 4.6: (a) Convergence $\mathcal{C}_n^{(13)}$ (see Eq. (4.26)) as a function of the resolution, given by the number of qubits n, for different values of λ (scale for quantum effects). To match the number of points between the two systems, extra points are taken onto the connecting line between two adjacent points in the n qubits discretization. (b) Minimum number of qubits required to obtain a fixed arbitrary fidelity $\tilde{\mathcal{C}}^{(13)}$ as a function of λ.

manifold, which offers less flexibility in terms of parameter evolution, requires a finer timestep.

In particular, to span the entire Hilbert space, we would need $M_p = 2N$ variational parameters. As M_p deviates from this value, our ability to capture dynamical fluctuations diminishes, necessitating more timesteps to accurately track the wavefunction evolution. This provides an explanation for the lower fidelity values observed in Table 4.1 when a larger number of qubits is employed. In such cases, either M_p or the number of timesteps does not increase in alignment with the scaling necessary to maintain fidelity at a stable level.

It is important to note further that this principle is applicable when $M_p > M_{min}$, where M_{min} represents the minimum number of variational parameters required to reproduce the target function with a specified accuracy. This minimum parameter count can change over the course of the wavefunction evolution on the basis of the complexity.

Space resolution and classical limit

In this preliminary study, we performed only numerical tests on relatively small-scale systems for which numerical simulations of our quantum algorithm were possible with the available computational resources. However, it is essential for us to confirm that the resolution we employed is sufficient to accurately capture the dynamics of the system. However, as we approach the classical limit, the space resolution needed to capture the right dynamical behavior increases. This is clear in the left panel of Fig. 4.6, where the convergence of the probability distribution is shown as a function of the spatial resolution

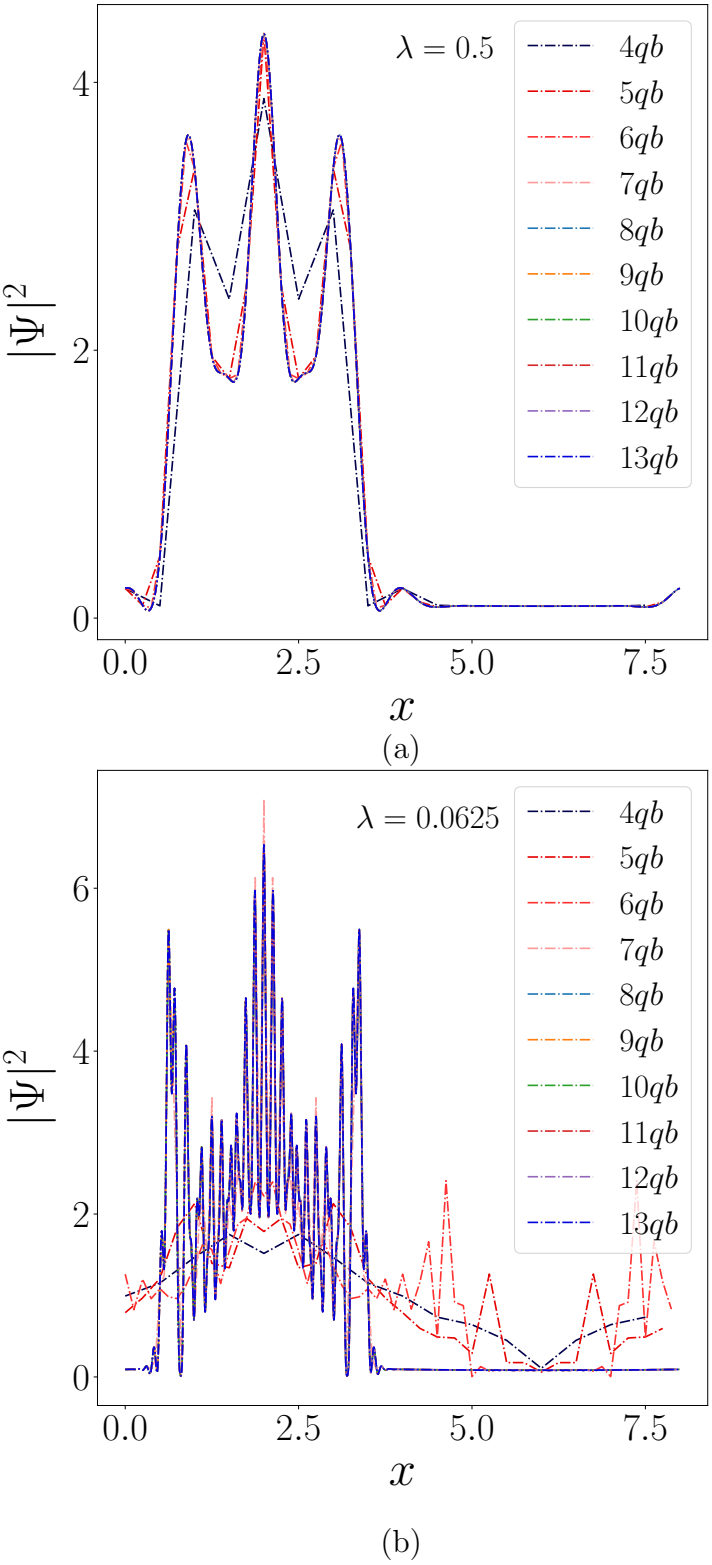


Figure 4.7: Density distribution at a fixed time frame for different resolutions (i.e., number of qubits n). On the top (a) and bottom panel (b) the scale λ is set respectively to 1/2 and 1/16. One can notice how higher resolution is needed to resolve a more classical system (lower λ).

for simulations with different scales λ. We observed that with decreasing λ accurate results require a finer representation of the space coordinate. This is mainly due to the appearance of peaked structures observed in the dynamics (see Fig. 4.5), which are harder to resolve than in the case of smaller λ values. It is worth mentioning that the increase in space resolution also requires a corresponding decrease of the simulation time step (Table 4.1).
In the right panel of Fig. 4.6 the resolution is shown as a function of the scale λ for different convergence values. From an empirical fit we showed that the number of qubits necessary to resolve the dynamics of a system scales as $\mathcal{O}(\log(\lambda))$. To quantify convergence, we used the L_2 norm between the n qubit probability distribution f_n – in a fixed time frame – mapped onto the 13 qubit grid and the 13 qubit probability distribution f_n

$$\mathcal{C}_n^{(13)} = ||f_{13} - f_n||_{L_2} \,. \tag{4.26}$$

In detail, the scaling law is fitted with a logarithmic function $n(\lambda, \tilde{\mathcal{C}}^{(13)}) = Klog(\lambda) + q(\tilde{\mathcal{C}}^{(13)})$, where $K = -1.44$ and $q(\tilde{\mathcal{C}}^{(13)})$ is the resolution needed to obtain the desired convergence factor $\tilde{\mathcal{C}}^{(13)}$ when $\lambda = 1$. Here $\tilde{\mathcal{C}}^{(13)}$ indicates a reference value of $\mathcal{C}_n^{(13)}$, chosen a priori, which therefore does not depend on n.

To be able to determine from a qualitative point of view what value of $\mathcal{C}_{13}$ is needed to obtain convergence in resolution, we plotted in Fig. 4.7 the probability distribution at a fixed time step, for different resolution and different λ. Comparing the images of this plot with the graphics in Fig. 4.6 tells us what *convergence level* is associated with a numerical value of $\mathcal{C}_{13}$. We observed that the right behavior can be captured as soon as the various density distributions start overlapping. More precisely, this happens for 6 qubits when $\lambda = 0.5$ and for 8 or 9 qubits when $\lambda = 0.625$. It is fair to assume that a L_2 distance of $\mathcal{O}(10^{-1})$ is enough to resolve the dynamic. We hence gather from both the fit and the previous remarks that a one dimensional resolution of 11 qubits can be enough to resolve a simulation approaching the classical limit with λ up to $\mathcal{O}(10^{-3})$. Moreover, it is possible to show that for the simulations reported in Figure 4.4 with $\lambda = 1$ a resolution of 32 grid points (equivalent to 5 qubits) is sufficient to converge the primary features of the dynamics.

Sampling and system size

In order to better assess the advantages of the algorithm over classical method, is important to study the number of measurements (N_s) needed to accurately evaluate the elements in Eqs. (4.8) and (4.9). Measurements introduce a statistical noise in the solution of the equation of motion for the propagation of the wavefunction parameters, which has an impact on the overall dynamics. Building on [23] we investigate the behavior mentioned above in the case of the newly introduced term $\langle \partial_{\theta_k} \psi | \, \mathcal{H} \, | \psi \rangle$. The potential part is directly proportional to the measurement of the the ancilla qubit $\langle \sigma_V^z \rangle$, thus the variance of the measurements can be estimated by the following

$$\mathcal{E}_V = \phi_V(n) L \sqrt{\frac{1 - \langle \sigma_V^z \rangle^2}{N_s}}, \tag{4.27}$$

where the value of σ_V^z is intended in the limit of $N_s \to \infty$ and the norm of the potential $\phi_V(n)$ scales with the number of qubits as $2^{n/2}$ (this can easily be seen applying the

spectral method proposed in Ref. [32] to obtain the potential, where the wavefunction is normalized as in Eq. (4.6)). The fact that the number of shots scales exponentially with the number of qubits is related to the nonlinear nature of the problem. Precisely, it is a consequence of the factorization of the physical wavefunction and the potential (recall Eqs. (4.5), (4.12)), where the normalization factor appears as an additional parameter that depend on the number of grid points.

The kinetic part is given by a linear combination of three different sets of measurements, see Eq. (4.21). The variance is estimated with a quadrature-sum as

$$\mathcal{E}_K = \frac{2^{2n}}{L}\sqrt{\frac{4 - \langle\sigma^z_{k+}\rangle^2 - \langle\sigma^z_{k-}\rangle^2 + 2\langle\sigma^z_k\rangle^2}{N_s}}. \tag{4.28}$$

Here, the factor 2^{2n} emerges from the term $1/\Delta x^2$ required from the finite difference method. We observe that in both situations the number of measurements required for an arbitrary accuracy increases with the number of qubits.

4.4 Summary

In this section, we tackled the problem of simulating a many-body problem of collisionless self-gravitating particles interacting only through a potential. In a cosmological context, this describes, e.g., the case of gravitational instability of a cold dark matter fluid in an expanding background. Our analysis builds on the possibility to recover the dynamics of the Vlasov-Poisson (VP) equations by mapping it to a framework more suited for quantum computing (QC), namely the Schrödinger-Poisson (SP) equations.

We proposed a variational time-evolution (VTE) algorithm for the solution of the corresponding non-linear time-dependant Schrödinger-like equation (TDSE) in which, at each time-step, the potential, which is a functional of the time-evolved system wavefunction, is obtained upon minimization of a suitable parameterized unitary in the quantum register.

The proposed quantum algorithm was developed with the aim of scaling up to system sizes, which are, in principle, much less favorable for classical computers than for quantum computers. To this end, we used a compact (i.e., logarithmic) encoding of the spatial grid (i.e., n qubits describing 2^n grid points), while enabling the representation of any self-consistent potential, which can be described by combining a parameterized unitary circuit and classical normalization factors. In particular, working with a circuit depth that scales polynomially with the number of qubits, we were able to reach a final state fidelity of approximately 0.96 in a 5 qubits simulation. The corresponding wavefunction is presented in Fig. 4.4. Concerning the scaling of the VTE circuit, the number of terms required to evolve the wavefunction in a single timestep scales quadratically with the number of variational parameters. However, the number of timesteps required to achieve a given fidelity increases as the ratio between the number of variational parameters and the Hilbert space dimension decreases, as shown in Tab. 4.1. This behavior might be related to the heuristic ansatz used in our implementation (e.g, Figs. 4.1, 4.15).

In addition, the number of measurements required to reach a desired accuracy shows a polynomial scaling with the number of grid points. We point out that this behavior is not specifically related to our proposed VTE algorithm, but to the approach chosen

to tackle the nonlinear nature of the problem, namely factorizing the potential and the wavefunction into unitary circuits followed by classical normalization.

Moreover, using classical simulations, we investigated how the required resolution changes as we approach the classical limit $\hbar/m \to 0$ in a $1D$ scenario. The proposed empirical log-scaling law opens up new interesting perspectives for the use of QC in the propagation of the SP equation in more general settings, including the $3D$ case.

In this chapter we posed the foundations for a quantum algorithm able to solve the dynamics of a self-gravitating collisionless fluid. While scaling up of the quantum approach to system sizes that may be relevant for cosmological prediction in 3D seems is unlikely before the advent of fault-tolerant quantum computing, there may be interesting studies (e.g., the study of static and dynamic phase transitions) which may occur already in low dimensions (1D) and that can become classically hard because of the complexity of the quantum description of the SP equation (e.g., because of the growing entanglement). A similar strategy was recently implemented in the domain lattice gauge theory (see [113]).

At the current state of development, our QC algorithm is clearly not competitive, in terms of accessible dynamic range, with respect to classical methods, both in cosmology and plasma physics, using near-term, noisy, QC with a number of qubits ~ 100 [114].

In the next chapter, we will focus on improving this algorithm in order to build a suitable candidate to run simulation of the SP equation.

Chapter 5

Towards a Schrödinger-Poisson algorithm for large scale simulations

The previous chapter introduced a promising variational algorithm and an only 5-qubit test case, though without examining its performance in higher-resolution scenarios.

The purpose of this Chapter is to assess the algorithm's behavior on such larger-scale systems. To this end, an in-depth analysis of the principal challenges associated with the scaling-up process is presented. Moreover, we propose an alternative approach to the SP equations , based on linearization.

The Chapter is structured as follows: in Section 5.1 we analyze the resources required for a mutli-dimensional implementation of the VTE algorithm for SP.

In Section 5.2, we detail the improvements implemented on the VTE algorithm. These include the use of a more refined variational ansatz, a rigorous selection of the time step, and an integration scheme based on Runge–Kutta methods, all aimed at enhancing stability and accuracy.

Section 5.3 addresses the challenges encountered when scaling up the algorithm. We identify the primary difficulty to be the construction of a suitable regularization scheme for the matrix defining the temporal evolution in Eq. (4.8). For the scheme to be effective, it must be system-independent while faithfully capturing the essential properties of the dynamics.

Finally, Section 5.4 presents an approach based on the linearization of the SP system using the Carleman method, which has been successfully applied in other contexts. This analysis reveals one of the limitations of the linearization strategy: the large number of terms required for a practical implementation. All results discussed in this chapter are supported by numerical simulations, providing a quantitative basis for the comparison between methods. Furthermore, they have not yet been published, as they are part of an ongoing study.

5.1 Multidimensional approach

The resources required to implement the algorithm presented in the previous chapter suggest promising potential for large-scale deployment. However, before proceeding with further implementations, it is crucial to carefully consider potential challenges and drawbacks that may arise when scaling up the problem.

Let us begin by considering the bidimensional case. In this scenario a continuous wavefunction evaluated in a specific point of the coordinates' space $\psi(x, y)$ maps to a coefficient ψ_{x_i,y_j} identified by the couple of integers (i, j). Now given that the box in which we model the spatial coordinate is isotropic, we can model it as an $L \times L$ box such that the grid corresponding to x is the same used for y; i.e with $L = 1, N = 4$ we have a box given by $[0, 0.25, 0.5, 0.75] \times [0, 0.25, 0.5, 0.75]$. In this way, we can write $\psi_{x_i,y_j} =: \psi_{i,j}$ with $i = 0; 1; ...; N - 1$. It is easy to spot that we can build a correspondence between the ordered couple (x, y) and an integer l such that

$$(x, y) \mapsto l = 2^n x + y\,. \tag{5.1}$$

This can be easily seen if we move the ordered couple (i, j) into binary domain. Then we impose a hierarchical (positioning) order. We now can combine the bit string obtained by the ordered binary couple and convert it to a decimal number l. We can generalize the previous expression for a $D-$dimensional case.
Let us note that we are dealing with a D dimensional box, with a single dimension grid composed by $N = 2^n$ points, with n the number of qubits. We can map the D dimensional ordered couple $(x_1, x_2, ...x_N)$ into the integer l such that

$$l = \sum_{i=1}^{N} x_i\, 2^{(D-i)n}\,. \tag{5.2}$$

The previous consideration shows us how is possible to implement a multidimensional amplitude encoding. The hierarchical structure given to the indices in Eq. (5.2) translates in the fact that the most significant qubits encodes one dimension while the least significant qubits another one. Let's take as an example a three dimensional box, where each block is divided into $2^5 = 32$ gridpoints. This can be implemented using a total of 15 qubits: the first 5 encodes the X-coordinate, the middle 5 the Y, and the most significant 5 qubits the Z-coordinate. This way, also the action of operation that acts selectively along one axis can be decomposed accordingly, i.e, the operator $\widehat{\boldsymbol{\nabla}} = \widehat{\nabla}_x \otimes \widehat{\nabla}_y \otimes \widehat{\nabla}_z$; this way, given that the depth of $\widehat{\nabla}_x$ scales as $\mathcal{O}(n)$, the required depth for $\widehat{\boldsymbol{\nabla}}$ remains $\mathcal{O}(n)$.
In practice, when we will implement the kinetic energy it will be enough to work separately on the three dimension X, Y, Z. On the other hand, given that the potential is a function on the whole D-dimensional space, it is not possible to decompose its action on each axis. The scaling will behave accordingly.

5.2 Improvements

For readability we write again Eqs. (4.8) and (4.9) describing the time evolution of the variational algorithm we aim to improve. The parameters that describe the evolution of the wavefunction Ψ are obtained by solving the following problem

$$\mathrm{M}\,\dot{\boldsymbol{\theta}} = \mathrm{B}, \tag{5.3}$$

where

$$\mathrm{M}_{kl} = \mathrm{Re}\left\{\langle \partial_{\theta_k}\Psi \mid \partial_{\theta_l}\Psi\rangle - \langle \partial_{\theta_k}\Psi \mid \Psi\rangle \langle \Psi \mid \partial_{\theta_l}\Psi\rangle\right\} \tag{5.4}$$

$$\mathrm{B}_k = \mathrm{Im}\left\{\langle \partial_{\theta_k}\Psi|\mathcal{H}|\Psi\rangle - \langle \partial_{\theta_k}\Psi \mid \Psi\rangle \langle \Psi|\mathcal{H}|\Psi\rangle\right\} \tag{5.5}$$

This is a variational method based on the evaluation of the terms appearing in both Eqs. (4.8) (4.9). The time evolution is then obtained by the solution of Eq. (4.7).

5.2.1 Reference Timestep

Another important factor to consider before scaling-up is how the number of timesteps increases with the dimension of the problem. In order to do so, we might need to make a few considerations first.

The fact that we used a variational approach does not allow us for a physical estimation of the timestep (e.g., CLF condition). We can, however, make a couple of considerations to set the boundaries for the timestep. Our method is based on a linearization of the problem at time $\tilde{t}$, during the evolution, to evaluate the parameter's update factor. This linearization occurs N_t times during evolution. However, the problem in question is physical in nature and has certain dynamical variables, such as the time scale of evolution $T \approx 1/\sqrt{G\rho^*}$. We can thus consider the maximum cell movement of a fluid element as the ratio $V\Delta t$, with $V = L/T$. If we introduce a constant C to mitigate this shift we get that, using a numerical integration scheme, we would have that a particle moves at maximum $C = V\Delta t/\Delta x$ cells during a single timestep. From this assumption, we can extract a reference timestep as

$$\Delta t = C\frac{\Delta x}{V} = \frac{C}{\sqrt{G\rho^*}}\frac{1}{N}\,. \tag{5.6}$$

We can now compare the number of timestep a simulation takes in order to converge and compare it with a reference timestep given in terms of cell movements, i.e, if a simulation converges with a $C = 10^{-4}$ we instantly know that is an overkill compared to the classical simulation and the dynamic of the problem.

In the previous Chapter 4, the algorithm used as time integration method the first-order Euler method. Although fairly straight-forward and simplistic, this scheme is not optimal for more complex time integrations, as the local error scales with $\mathcal{O}(\Delta t^2)$. Employing a more refined integration scheme, such as a fourth-order Runge-Kutta, could reduce the overall number of timesteps required, aiding scalability, and improving the algorithm's precision. With the ansatz choice made in previous chapter 4 it has not been possible to implement a Runge-Kutta scheme with adaptive timestep, because of noise building up to unreasonable results. This makes sense because the evolution is carried on the system's parameters, and the initial layout of the circuit determines how they interact with each other and how much they influence the output function. In other words, a good initial ansatz determines in some way how robust the algorithm is to noise and how well the wavefunction can be expressed during the evolution. In this sense a good ansatz choice is related to the time integration, as a good time integration scheme must be noise resilient.

5.2.2 Ansatz

For the reasons listed in Sec. 5.2.1, implementing an adaptive timestep method requires identifying the best ansatz for our problem, one that is both noise-robust and sufficiently flexible to capture the system's evolution. While testing a range of ansatz to select the optimal one is viable, it would require prohibitive time investment.

Instead, we take the ansatz in Fig. 4.2 and explore variations in circuit connectivity and rotational gates. We find that the best compromise between expressivity and noise resilience is achieved with full entanglement, where each qubit connects to all others. This aligns with our expectations: the problem is non-local, and evolving the entire system enables all-to-all interactions.

With this ansatz choice, we can now use a 4^{th}-order Runge-Kutta integration scheme with a self adapting timestep, where the error is evaluated up to the 5^{th} order.

5.3 Scaling Challenges

We have shown that the qubit requirements for implementing the algorithm are manageable and that the number of variational parameters can be reduced to make the method practical for realistic applications. Furthermore, the time-evolution procedure does not exhibit barren plateaus. Despite these advances, several key challenges still prevent the algorithm from scaling to larger systems. In this section, we examine these difficulties and outline our attempts to address them.

5.3.1 Regularization problem

The main obstacle is the numerical instability of the time-evolution equation, Eq. (4.7). Solving $\dot{\theta}$ requires inverting a discrete ill-posed problem, where even small perturbations can lead to large deviations. This instability comes from the structure of the geometric tensor in Eq. (4.8), which can be written as

$$\mathbf{M} = \langle \partial_{\theta_k} \Psi | \, (\mathbb{1} - \Pi_\Psi) \, | \partial_{\theta_l} \Psi \rangle \,, \tag{5.7}$$

where $(\mathbb{1} - \Pi_\Psi) = \Pi_{\Psi^\perp}$ project onto the subspace orthogonal to $|\Psi(\theta)\rangle$. As a result, Eq. (4.7) belongs to a well-known class of ill-posed problems of the form

$$\Pi x = B. \tag{5.8}$$

Such problems are unstable even in the continuum, and discretisation usually amplifies this behavior, further aggravating the instability. This makes a robust regularization strategy essential. In particular, we seek a method that remains reliable across different initial conditions and does not require delicate parameter fine-tuning to function properly. The motivation is straightforward: for large systems, it is impractical to perform multiple trial simulations solely to identify suitable parameters. In contrast, if there is a regularization scheme that depends only on the intrinsic dynamics of the system and not its size, then its parameters can be calibrated in smaller test simulations and subsequently applied to larger runs.

In the previous chapter, we solved Eq. (4.7) using a routine from the `scipy` library. We start the search for the optimal cutoff strategy by examining how this method operates and what assumptions it implicitly relies on.

Truncated singular value decomposition (TSVD)

The `scipy` implementation is based on truncated singular value decomposition (TSVD), a widely used technique for stabilising ill-conditioned linear systems.

The singular value decomposition (SVD) of $\mathbf{M}$ is

$$\mathbf{M} = \mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^\top, \tag{5.9}$$

where $\mathbf{U} \in \mathbb{R}^{m\times m}$ and $\mathbf{V} \in \mathbb{R}^{n\times n}$ contain the left and right singular vectors, and $\mathbf{\Sigma}$ stores the singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$, with $r = \mathrm{rank}(\mathbf{M})$. These values quantify how each component contributes to the matrix structure.

The TSVD regularization retains only the largest singular values k, producing the approximation

$$\mathbf{M}_k = \mathbf{U}_k\,\mathbf{\Sigma}_k\,\mathbf{V}_k^\top,$$

where $\mathbf{U}_k$ and $\mathbf{V}_k$ contain the first singular vectors k, and $\mathbf{\Sigma}_k$ the corresponding singular values. This truncated matrix is the best rank-k approximation of $\mathbf{M}$ in the Frobenius norm,

$$\|\mathbf{M} - \mathbf{M}_k\|_F = \min_{\mathrm{rank}(\mathbf{N})\leq k} \|\mathbf{M} - \mathbf{N}\|_F. \tag{5.10}$$

By discarding the smallest singular values,typically the components most sensitive to noise, TSVD improves stability and removes spurious contributions. Provided that the cutoff is chosen appropriately, the regularized system can be solved as

$$\dot{\theta}_j = \sum_{k<r} \frac{\mathbf{U}_{k,j}^\top\,\mathbf{B}}{\sigma_k}\,\mathbf{V}_{k,j}. \tag{5.11}$$

The central difficulty therefore lies in selecting a cutoff strategy that suppresses noise without discarding genuinely relevant information. A poor choice of cutoff either fails to stabilize the system or introduces bias into the evolution.

Without proper regularization, numerical errors rapidly accumulate, ultimately overwhelming the physical signal and rendering the results meaningless. Thus, a reliable cutoff strategy is essential for scalability. Ideally, it should depend neither on the system size nor on specific initial conditions.

To investigate this, we begin with a small five-qubit system using the same initial condition as in the previous chapter. This resolution captures all qualitative features of the dynamics while remaining simple enough to allow systematic testing of different regularization strategies.

In search of a Cutoff Strategy

The first step is to analyze the initial distribution of singular values, looking for jumps, abrupt changes, or patterns that might indicate where to partition the sequence. In the upper right panel of Fig. 5.1, we observe that the singular values form distinct clusters. Specifically, there is one dominant value of $\mathcal{O}(10^2)$, followed by a relatively flat sequence that decreases to around 1. Beyond this, smaller groups of values appear, clustered together, until a sharp drop occurs near 10^{-4}. From this pattern, we infer that the largest singular value represents the primary contribution to the system's evolution, whereas the subsequent values contribute in descending order of relevance, organized into distinct groups.

The same thing does not happen for a different ansatz, as can be seen in the top-left panel of Fig. 5.1, where, however, the largest initial singular value remains.

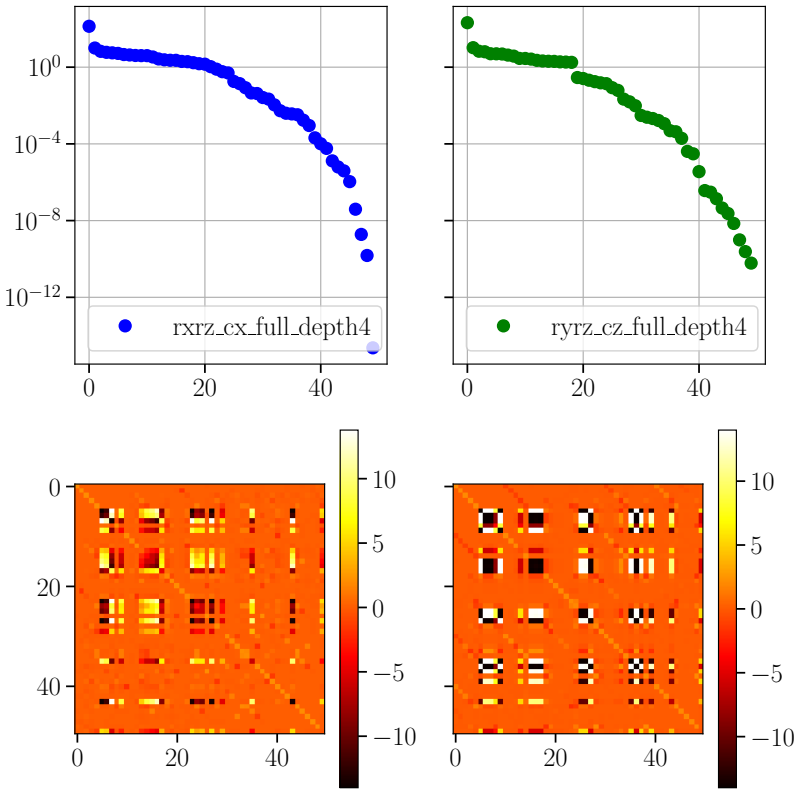


Figure 5.1: Top row: SVD of the initial quantum geometric tensor M for two different ansätze. Bottom row: the same matrix shown as a color plot. The two captions correspond to the ansatz used; here, the first two letters (`rxrz` or `ryrz`) indicate the type of rotational gates used (parameterized RX, RY RZ rotations), the next term (`cx` or `cz`) specifies the entangling gate, `full` indicates full entanglement, and `depth4` denotes that the circuit has 4 layers repeated.

A common strategy involves partitioning the sequence just before major transitions. However, deciding which transition point to use as the cutoff point presents a challenge. In Fig. 5.2, we illustrate the system's behavior when the cut is applied at the last big, discontinuity in the singular values. We notice that the system fails to capture nonlinear effects, indicating a loss of critical information.

It is also important to note that the distribution of singular values changes at each timestep, adding further complexity to the problem. With each step, noise has the potential to grow increasingly significant and influence the system's dynamics.

Looking at Fig. 5.1, we observe that different quantum circuits correspond to distinct initial conditions. This is expected, since the matrix M depends solely on the chosen parameters and the ansatz. However, despite these differences, the system's evolution is anticipated to exhibit common features, as all descriptions represent the same underlying physical dynamics. In other words, the SVD of the matrix M may vary depending on the circuit used to express the evolution, but at each timestep, we solve the linear system in Eq. (5.3), where the information of the Hamiltonian is contained in B. Consequently, any differences in initialization should diminish over time, provided that both descriptions

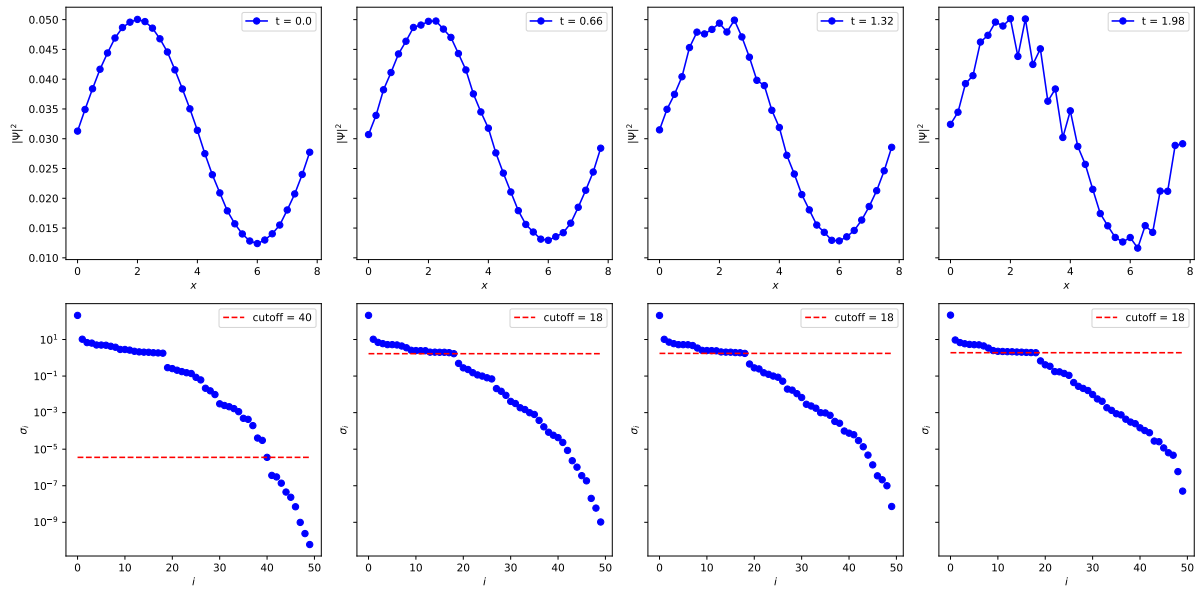


Figure 5.2: (First row) Density distribution obtained with VTE algorithm at different times. The regularization is done by taking only the singular values before the last pronounced discontinuity. (Second row) Singular Values and cutoff line.

accurately reproduce the underlying system.

We utilize a variational approach to obtain a reference trajectory of singular values $\tilde{\sigma}(t) = \{\tilde{\sigma}_1, \ldots, \tilde{\sigma}_{N_t}\}$ by determining, at each timestep, the parameter set that best describes the classical wavefunction $\tilde{\Psi}(t)$ obtained from classical evolution, such that $U(\tilde{\theta}_t)\,|\mathbf{0}\rangle = \Psi(t)$. Specifically, we reinitialize the quantum wavefunction at each timestep by identifying the parameter set that minimizes the fidelity $\mathcal{F}$ between a trial quantum state $|\psi(\theta)\rangle$ and the quantum state corresponding to the classical wavefunction $\left|\tilde{\psi}_t\right\rangle$:

$$\min_{\{\tilde{\theta}\}} \mathcal{F}(\tilde{\psi}_t,\, \psi(\theta)) = \min_{\{\tilde{\theta}\}} \langle\tilde{\psi}_t|\, \psi(\theta)\rangle\,. \tag{5.12}$$

To mitigate the issue of the barren plateau during optimization, the quantum state of the test is initialized using the set of parameters from the previous timestep, $\{\theta_{t-\delta t}\}$.

After determining the parameter set at time t, we perform an SVD to extract the singular values $\tilde{\sigma}_t$. Repeating this procedure at each timestep allows us to construct an approximate representation of the temporal evolution of the singular values, capturing their behavior in an assumed exact evolution.

In Fig. 5.3, we present the evolution of *exact* singular values at different time frames. Initially, noticeable gaps between the singular values highlight distinct dynamics. However, as time progresses, these gaps diminish, with most singular values aligning along a nearly uniform trend, except for the dominant largest value. In particular, after 60 iteration, singular values begin to decrease again while maintaining a linear structure. This observation illustrates why cutoff strategies based on singular value discontinuities fail: as the system approaches shell crossing, these discontinuities vanish.

To understand the impact of fixed cutoffs, we examine how the relevant singular values evolve over time. Since their number changes dynamically, an adaptive cutoff strategy

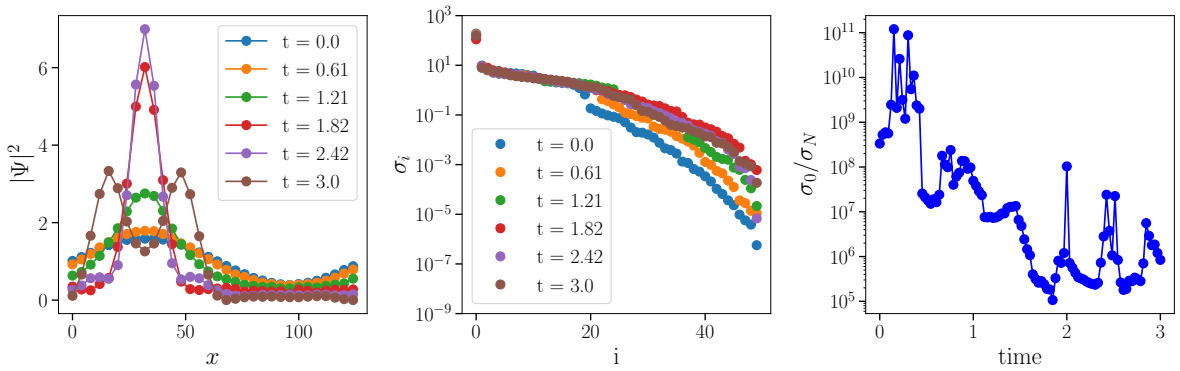


Figure 5.3: The value presented in this picture refers to quantum states obtained by minimizing the fidelity between a trial quantum state and the classical reference at each timestep. (left) Density function as a function of the position at different time steps. (center) Singular values at different time steps. (right) Evolution of the conditioning number of the martix M. This number is the ratio between the largest and the smallest singular values, it defines the grade of ill-condition of the matrix.

appears essential to capture these variations. To test this hypoThesis, we first evaluate the accuracy of fixed cutoff strategies, as they provide a useful benchmark for estimating the error introduced when the cutoff is held constant. If fixed cutoffs yield acceptable results, this would indicate that the number of significant parameters does not deviate substantially from an optimal range.

Figure 5.4 shows the mean squared error (MSE) between the quantum state and the classical reference state over time for different fixed cutoffs. The MSE is defined as

$$\mathrm{E}(t) = \sum_{k=1}^{N} \frac{\left(|\tilde{\Psi}_t|^2 - |\Psi(\theta_t)|^2\right)^2}{\sqrt{N}}, \tag{5.13}$$

where the cutoff is imposed by fixing r, for example, by retaining only the first 39 singular values .

To associate a qualitative interpretation with the MSE values, we wish to identify when a simulation begins to diverge from the expected density. To this end, we step-by-step monitor the density profile for several fixed cutoff choices. Figure 5.4 highlights the divergence times for $r = 33$, $r = 38$, and $r = 44$, indicating that an error threshold of approximately $\mathrm{E} \approx 1$ marks the breakdown of a qualitatively accurate approximation. Indeed, for all values below this threshold, the overall shape of the density distribution is still captured correctly, despite the presence of some numerical noise.

Finally, Fig. 5.5 shows that no fixed cutoff yields a qualitatively accurate simulation up to and beyond $t = 3$, regardless of the chosen value.

All simulations were performed with $C = 1/100$ in roughly 10^3 time steps. However, as shown in Fig. 5.6, increasing the number of time steps does not significantly improve accuracy when fixed cutoffs are used. This occurs because the cumulative noise introduced by a fixed cutoff strategy grows over time, negating any benefits gained from higher temporal resolution.

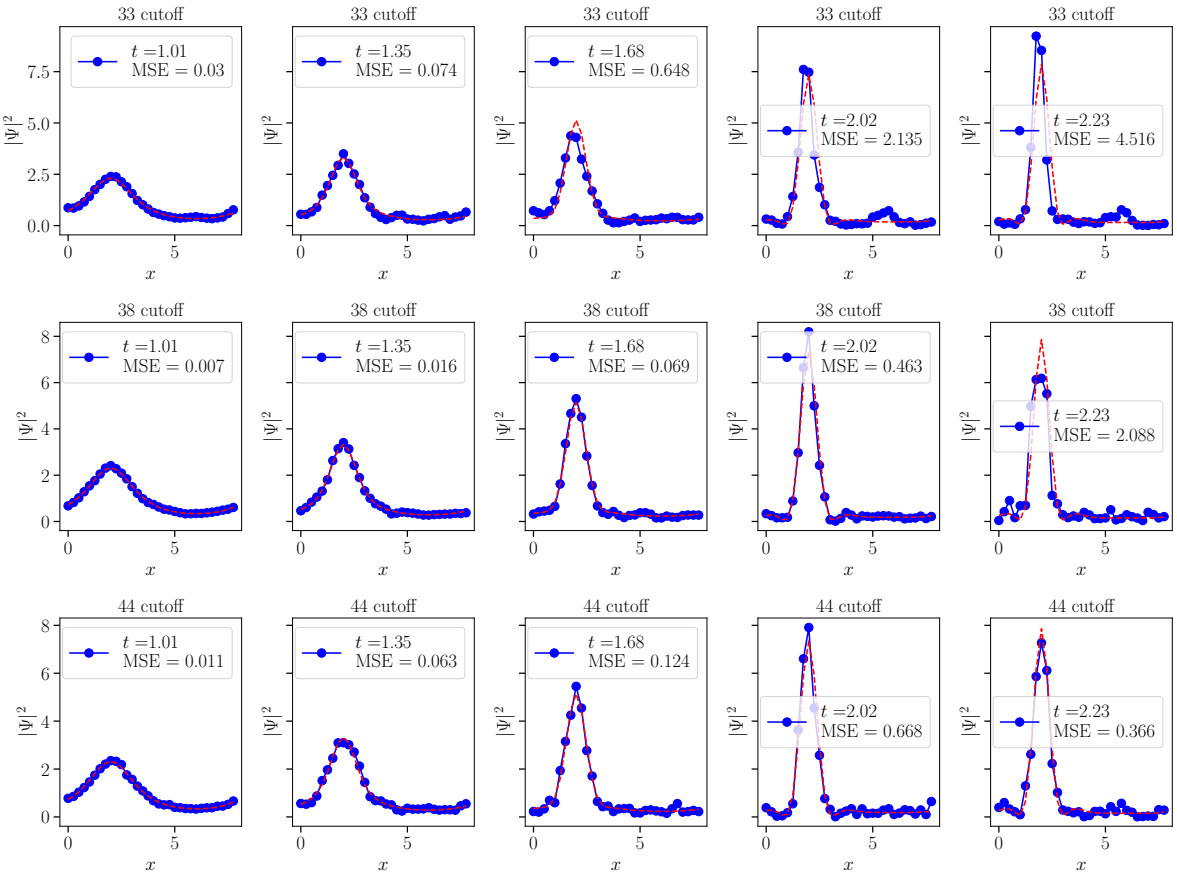


Figure 5.4: Comparison between reference density (dotted red line) and density obtained with VTE using fixed cutoff method, where above each panel wenote how many singular values have been considered. The simulation is done using $C = 0.01$. In the label of each panel we report the time frame and the associated MSE. This figure provides a visual representation of how the density profiles relates to the corresponding MSE values.

Together, these findings confirm that a dynamic cutoff strategy is necessary. This strategy should minimize abrupt changes in the number of significant singular values, particularly before shell crossing, while simultaneously employing adaptive time steps to balance the evolution of the system against the accumulation of noise.

The first example of dynamical cutoff strategy is the one implemented by the function `lstq` in the module `scipy.linalg`. The cutoff threshold is determined by retaining singular values σ that satisfy $\sigma \geq r_c\sigma_0$, where $\sigma_1 > \sigma_2 > \cdots > \sigma_N$ and r_c is an initial parameter. Combined with the adaptive $RK45$ integration scheme, this approach produces reasonable results up to $t = 3$. Optimal performance is achieved with $r_c = 10^{-8}$ and an absolute tolerance of 10^{-7}, as shown in Fig. 5.7, using approximately $N_t \sim 2500$ time steps.

Despite its success in small-scale simulations, this method requires extensive trial and error to fine-tune r_c and absolute tolerance, making it impractical for larger systems. For example, scaling to $n = 30$ qubits would render individual trial runs computationally prohibitive, particularly without a reference density for comparison. This happens because, when the number of qubits increases, so does the number of variational parameters, thus

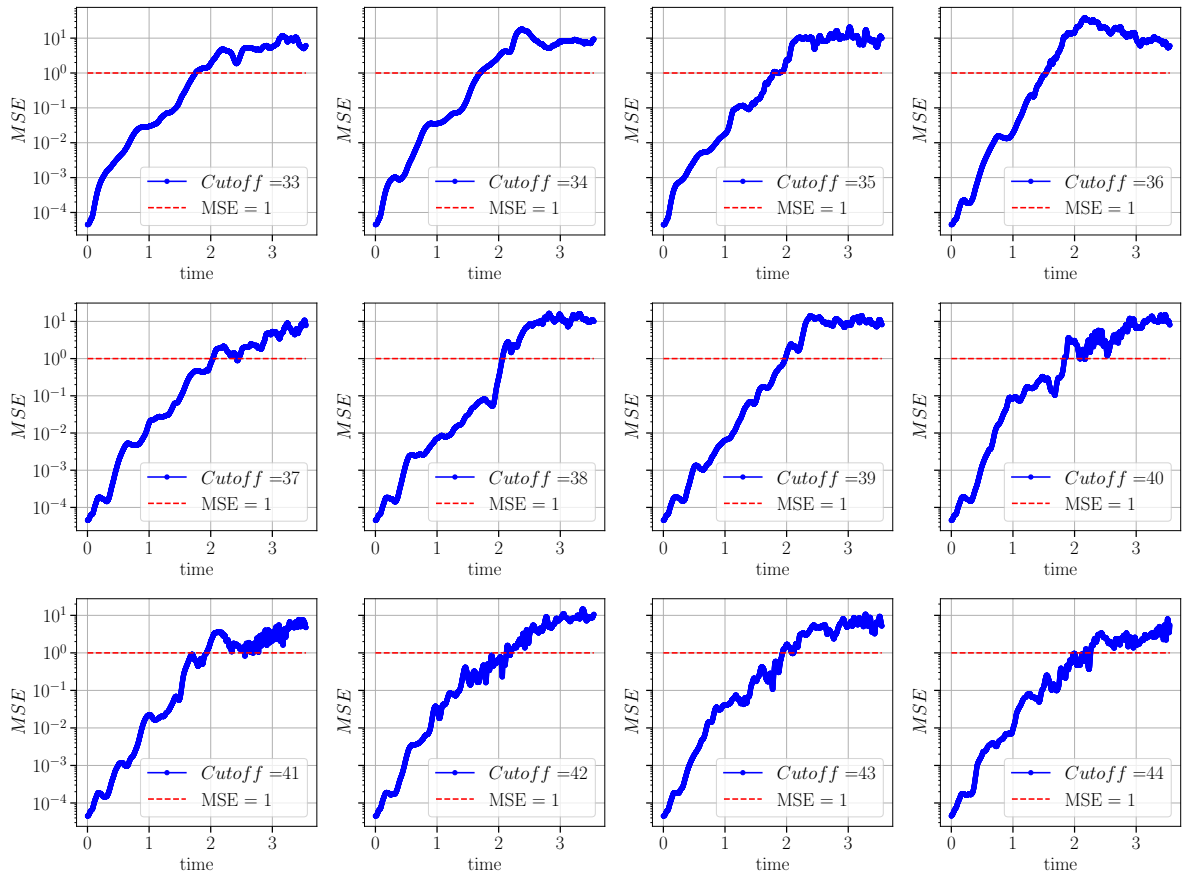


Figure 5.5: Mean Squared Error MSE as a function of time for simulations with fixed cutoff strategies but different cutoff. We consider the threshold $MSE = 1$ to be maximum allowed error for a valid representation of the density.

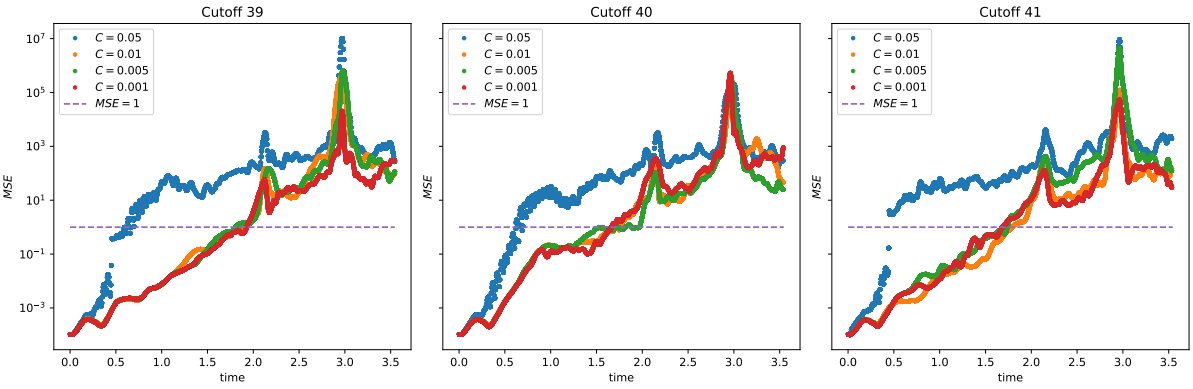


Figure 5.6: MSE for for simulations with different number of time steps with fixed cutoff strategy. The C in the legend refers to maximum cell displacement explained in section Sec. 5.2.1. The simulations presented are regularized by taking only the first 39(left), 40(center) and 41(right) singular values. We have chosen these three examples because they represent the range in which the fixed cutoff works better. The dotted line represent the maximum allowed MSE for which the density manages to capture the qualitative behavior of the target one.

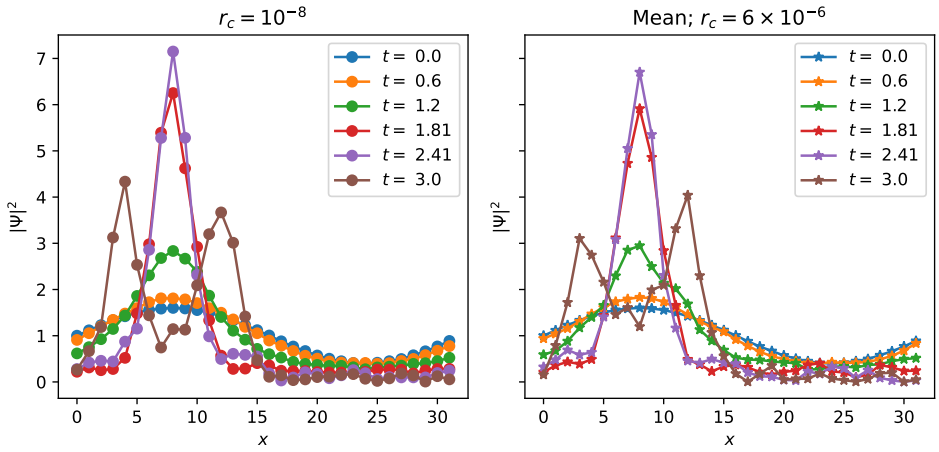


Figure 5.7: Density function at different times for two different simulations, the one on the left uses a cutoff strategy based on the maximum singular value with $r_c = 10^{-8}$, while the one on the right uses the mean of the first five singular values (Eq.(5.14)) with $r_c = 6 \times 10^{-6}$.

the quantum geometric tensor changes, and so does the structure of the problem: the value r_c that proved to be effective for a simulation of 5 qubits might not work for a simulation of 6 qubits.

To address these limitations, we propose a refined regularization strategy that instead of relying solely on the largest singular value σ_1 considers a function of the singular values $f(\Sigma)$ averaging the top five singular values, as depicted in Fig. 5.7. In this scenario, the cutoff threshold ϵ is defined by

$$\epsilon = r_c \, f(\Sigma) = r_c \sum_{j=0}^{4} \frac{\sigma_j}{5} \, . \tag{5.14}$$

The choice of taking the mean of the first five singular values is an attempt to find the right $f(\Sigma)$ that is more relevant to core aspects of the systems' dynamics. However, this attempt proved to be less effective than the approach using only the largest singular value, yielding lower accuracy and requiring more time steps. This tells us that, in this specific problem, the largest singular value is a feature more relevant than the mean of the first five. This can been guessed also by looking at the approximately exact dynamic of the singular values in Fig. 5.3, where the first singular value changes in magnitude while remaining detached to all the others.

We point out again, that both density profiles shown in Fig. 5.7 are not symmetrical with respect to the peak value. This is not an issue of the regularization method or the variational algorithm, but rather of the ansatz chosen for both potential and wavefunction: in order for the system to preserve this property, it must be explicitly encoded in the variational ansatz. Imposing this constraint, however could not obviate to the regularization problem.

In fact, in our case, the small size of the system, allows for multiple trials in order to find which function better capture the core feature of the system dynamics. On the other hand, we will have to look for a strategy that is scale-independent.

Other regularization methods, such as Tikhonov regularization [115], [116], were also tested. By introducing a diagonal penalty term, this technique resembles SVD but remains heuristic and fails to adapt to the time-dependent nature of our problem.

We further explored hybrid approaches, as the one suggested by Gavish and Donoho [117], where to estimate a reference error to decide the magnitude of the correction, both Tikhonov regularization and TSVD are applied. This approach should be able to dynamically adjust the cutoff parameter. However, we verified that this approach is not suitable for our situation, as the evolution did not converge in the first steps. We suppose the reason resides in the fact that the problem itself has multiple solutions and some of those, are simply numerical noise.

In conclusion, identifying a robust and scalable regularization method is inherently challenging. Although promising in smaller scenarios, these approaches often falter as the system scales, underscoring the need for innovative strategies capable of addressing both accuracy and scalability.

In conclusion, what should be done is identify the key feature that better represents the dynamic of the system, independent on the scale and on the initial condition, because in a generic scenario it would not be advantageous to repeat all the process of finding the best regularization strategy. This by itself might prove to be quite a demanding, but not impossible task. However, as we will see, this will prove to be one among different challenges to a large scale efficient implementation.

5.3.2 Measurement precision

To reliably estimate the algorithm's performance in higher-dimensional systems, we must account for measurement readout costs. Specifically, we determine how many measurements are needed to evaluate the terms in Eqs. (4.9) and (4.8). The most expensive terms involve potential and kinetic energy. Note that the cost of measuring the kinetic energy in Eq. (4.28) can be reduced using a Fourier-transform approach instead of finite differences, eliminating the $1/\Delta x^2$ prefactor. The potential measurement cost remains unchanged.

For a desired measurement precision ϵ, inverting Eq. (4.27) gives

$$N = \phi_v^2 \frac{1 - \langle \sigma^z \rangle}{\epsilon^2}, \tag{5.15}$$

where ϕ_v represents the potential norm and scales as $\mathcal{O}(2^{n/2})$. The sampling of the potential therefore exhibits unfavorable scaling for large systems and observables with small expectation values ($\langle \sigma^z \rangle \to 0$). In this regime, the required measurements scale as $\mathcal{O}(2^n \epsilon^{-2})$.

For the matrix terms in Eq. (4.8), regularization imposes a cutoff threshold that sets a maximum achievable precision $\epsilon_{\min}$. Using singular values that are physically meaningful standardizes the procedure. The required measurements then scale as ϕ_v/ϵ^2, where ϵ is the desired precision. For example, a cutoff point of 10^{-4} implies $\mathcal{O}(\phi_v \times 10^8)$ measurements per value, with $\phi_v(n) \sim \mathcal{O}(2^{n/2})$. Such a large number of measurements could pose practical challenges in preserving a quantum advantage over classical algorithms, as the overhead may offset the computational speedup.

However, these considerations do not apply to Eq. (4.9), as those terms do not require regularization.

5.3.3 Timestep

A hybrid variational algorithm consists of both a classical and a quantum component. In the specific case of the VTE, the integration scheme employed is entirely classical. At each time step, the quantum computer is only tasked with measuring the quantities in Eq. (4.14) and Eq. (5.3). The resulting linear system is then solved using a classically chosen integration method. Consequently, the temporal evolution, and thus the total number of time steps, is determined by a classical integration scheme. As an example, with CFL condition the number of time steps scales as $\mathcal{O}(1/\Delta x) \sim \mathcal{O}(2^{-n})$. In other words, there is no quantum advantage in terms of time stepping. In the VTE, the quantum part is solely used to evaluate quantities required for the evolution; the time-stepping procedure itself is not improved. In fact, whereas a classical algorithm may require $\mathcal{O}(N_t)$ time steps, the variational quantum approach can demand an even larger number.

This motivates the investigation of potential improvements in this aspect, such as offloading the time integration, or part of it, to the quantum computer.

5.4 Linearization Approach

We have seen the strength and weaknesses of the VTE approach; while on one hand the limited depth variational circuits enables fast runs and is suited for NISQ devices, on the other, the time integration remains classical, thus providing no advantage regarding the number of timesteps. This implies that in a typical situation of interest, where the required resolution grows, the number of calls to the quantum device increases linearly. Each call has an overhead, thus a possible quantum advantage might be put in shadows. We thus consider an alternative approach to the solution of SP. We are looking for a way to offload the time integration to the quantum device. We expect that this would require circuit with a greater depth, at least in the order of the number of timesteps $\mathcal{O}(N_t)$.

Moving to a similar field, fluid dynamics, it has been shown that an approach based on linearization of the original system might lead to quantum advantage [118]. The work suggests employing a specific technique, called Carleman embedding. It has already been used to simulate the Lattice Boltzmann system in the simplest scenario, showing promising results [119]. Specifically, with one big quantum circuit, the authors are able to retrieve the value of the density at a given time. Following the same path, we will try to adapt the method to SP set of equations.

5.4.1 Carleman Embedding

The Carleman embedding transforms a finite-dimensional nonlinear system into an infinite-dimensional linear system while preserving the intrinsic properties of the system. It does this by deriving an evolution equation for every term in the original system. Each time a new quantity is introduced, additional terms appear, typically products of existing ones, and these higher-order terms require their own evolution laws. These laws are obtained by comparing them with those already derived, creating a hierarchical cascade of terms. In practice, this infinite series of linear equations is truncated at some chosen order, which inevitably introduces a truncation error. Consequently, our goal will be the linearization of SP, in order to obtain a linear system of equation, only then we will search for an

effective quantum algorithm. We start by splitting Schrödinger into real $v_1 = \mathrm{Re}\,\psi$ and imaginary part $v_2 = \mathrm{Im}\,\psi$ so we can write

$$\frac{\partial v_1}{\partial t} = -\frac{D}{2}\Delta v_2 + \frac{1}{D}U v_2\,, \tag{5.16}$$

$$\frac{\partial v_2}{\partial t} = \frac{D}{2}\Delta v_1 - \frac{1}{D}U v_1\,, \tag{5.17}$$

where Δ is Laplace operator ∇^2. This is done to obtain a linear system that evolves strictly in real quantities, since Ψ is a complex wave function while the potential is valued in real terms. The next step to obtain a linear system is to find an expression that describes the evolution of the gravitational potential. Instead of solving the Poisson equation we introduce a smooth constraint γ on the potential equation, similar to a dissipation coefficient, this way we obtain a new set of equations for the potential U

$$\partial_t U = \gamma\left[\Delta U - \alpha(v_1^2 + v_2^2 - 1)\right]\,, \tag{5.18}$$

where $\alpha = 4\pi G\rho^*$. Formally this set includes the version as in Eq. (4.4), we thus aim to find the right value for γ such that the potential is rightly represented.

5.4.2 Classical solutions

Let us first try solving the equations in (5.16),(5.17),(5.18) with a Runge-Kutta integrator. This is a first trial without the Carleman decomposition and the system is, in fact, still non-linear, allowing to set an upper-bound to the solution we can expect to obtain with an infinite order Carleman approach. The only errors made in this scenario are attributed to the new form of the potential equation (5.18).

We first consider the system treated in Chapter 4, where the box dimension is $L = 8$, the diffusion $D = 1$ and $\alpha = 4\pi G\rho^* = 1$; the evolution is carried out until $t = 3$. We also use this test to find the optimal γ for Carleman implementations.

In Fig. 5.9 a comparison between the results obtained using a classical spectral method and the one obtained from our decomposition, that combines the real and imaginary time splitting with the soft constraint on the potential (Eqs. (5.16), (5.17), (5.18)), is shown. We obtain the right results with γ large and positive. Observe in fact that Eq. (5.18) can be seen as the heat equation with the non-homogeneous term $\gamma h(x) = v(x)_1^2 + v(x)_2^2 - 1$ and *periodic* boundary condition

$$\partial_t U(x,t) + \gamma\partial_{xx}U(x,t) = \gamma h(x)\,. \tag{5.19}$$

As γ increases, the solution of Eq.(5.18) converges faster to the forcing term, thus we recover a more refined solution; this can be seen visually in Fig. 5.9 and numerically in Fig. 5.8, where we have shown how the relative mean square error varies in time for simulations with different γ.

The integration is obtained with Scipy's `solve_ivp` module that uses a self adaptive timestep.

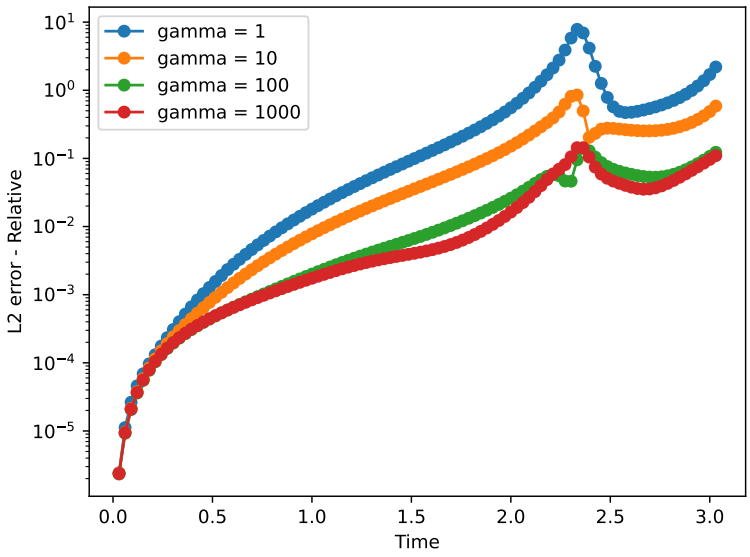


Figure 5.8: Error as function of time for simulation with different γ. The error is evaluated comparing the density ρ with the one obtained using a spectral method that we consider *exact*. We used the mean squared residual relative error $\sqrt{\sum_i (\rho_i/\tilde{\rho}_i - 1)^2}$, where $\tilde{\rho}$ is the exact solution.

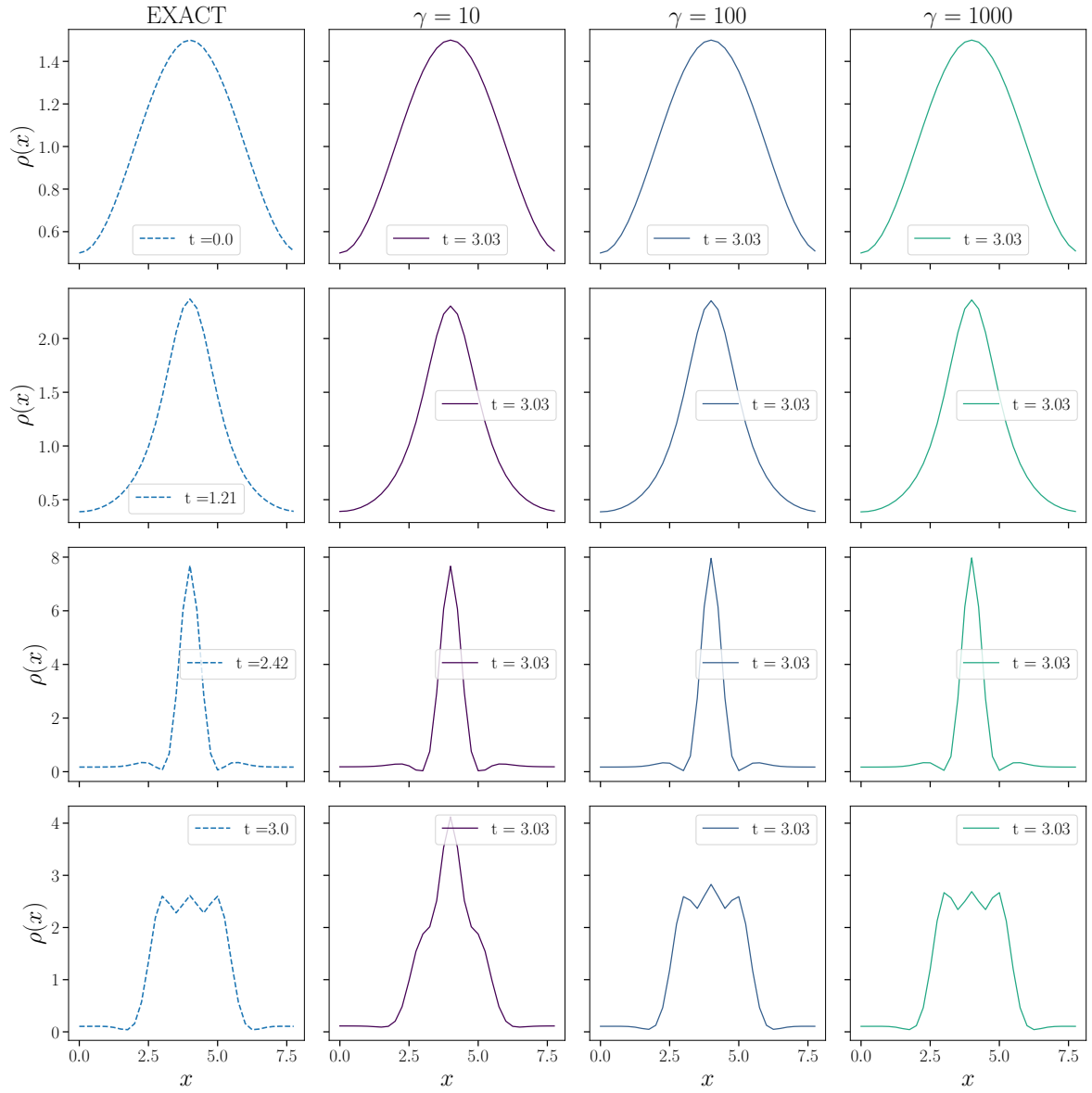


Figure 5.9: Density ρ as a function of space for different γ at different time frames. The figure is to be read in columns: In the first column from the left there is the reference solution obtained with a spectral method for SP; the second column represents the results obtained integrating Eqs. (5.18) for the potential with $\gamma = 1$; in the third $\gamma = 10$ and in the last $\gamma = 10^3$. All the simulation are done in the same setup: 5 qubits, a diffusion coefficient $D = 1$, a box dimension $L = 8$ and $\alpha = 1$.

5.4.3 Second order Carleman Embedding

The Carleman approximation aims to map a nonlinear problem into one of the kind $\partial_t \mathbf{v} = \mathcal{C}\mathbf{v}$, in order to do so we must define the evolution of every term in Eqs. (5.16), (5.17), (5.18),

in particular the quadratic terms. Thus, defining the new variables as components of $\boldsymbol{v}$

$$v_1 = \mathrm{Re}\{\psi\} \tag{5.20}$$
$$v_2 = \mathrm{Im}\{\psi\} \tag{5.21}$$
$$v_3 = U \tag{5.22}$$
$$v_4 = U \otimes \mathrm{Re}\{\psi\} \tag{5.23}$$
$$v_5 = U \otimes \mathrm{Im}\{\psi\} \tag{5.24}$$
$$v_6 = \mathrm{Re}\{\psi\} \otimes \mathrm{Re}\{\psi\} \tag{5.25}$$
$$v_7 = \mathrm{Im}\{\psi\} \otimes \mathrm{Im}\{\psi\} \tag{5.26}$$

we obtain the first version of the Carleman system, where we have chosen a truncation at second order that drops all third order terms, i.e, neglect the terms like Uv_1^2. The full-explicit expression and how it is retrieved is presented in Appendix B.

$$\partial_t v_1 = -\frac{D}{2}\Delta v_2 + \frac{1}{D}Bv_5 \tag{5.27}$$
$$\partial_t v_2 = \frac{D}{2}\Delta v_1 - \frac{1}{D}Bv_4 \tag{5.28}$$
$$\partial_t v_3 = \gamma\left(\Delta v_3 - \alpha B v_6 - \alpha B v_7 + \alpha\right) \tag{5.29}$$
$$\partial_t v_4 = \gamma\alpha \mathrm{O} v_1 + \gamma\left(\Delta \otimes 1\right) v_4 - \frac{D}{2}(1 \otimes \Delta)v_5 \tag{5.30}$$
$$\partial_t v_5 = \gamma\alpha \mathrm{O} v_2 + \gamma(\Delta \otimes 1)v_5 + \frac{D}{2}(1 \otimes \Delta)v_4 \tag{5.31}$$
$$\partial_t v_6 = -\frac{D}{2}\left(v_1 \otimes \Delta v_2 + \Delta v_2 \otimes v_1\right) \tag{5.32}$$
$$\partial_t v_7 = \frac{D}{2}\left(\Delta v_1 \otimes v_2 + \ v_2 \otimes \Delta v_1\right) \tag{5.33}$$

where B is an $N \times N^2$ used to obtain the vector N containing the quadratic term and the product of the wave function and the potential from their tensor products, for example $B_{i,j,k} = \delta_{i,j}\delta_{j,k}$, such that acting on $v_5 = v_3 \otimes v_2$ returns the $i-$th component

$$B_{i,j,k}(v_5)_{j,k} = B_{i,j,k}(v_3)_j(v_2)_k = (v_3)_i(v_2)_i\,. \tag{5.34}$$

In the same manner O acts on a vector v with the purpose of obtaining $1 \otimes v$ which is simply a matrix whose rows are the vector itself. This can be obtained by a rank 3 tensor: a collection of Identity matrices.

We observe that to fully describe the evolution of the system two additional components are needed

$$v_8 = v_1 \otimes v_2 \tag{5.35}$$
$$v_9 = v_2 \otimes v_1 \tag{5.36}$$

and their evolution

$$\partial_t v_8 = \frac{D}{2}\left(v_1 \otimes \Delta v_1 - \Delta v_2 \otimes v_2\right) \tag{5.37}$$
$$\partial_t v_9 = \frac{D}{2}\left(\Delta v_1 \otimes v_1 - v_2 \otimes \Delta v_2\right) \tag{5.38}$$

5.4.4 Third order Carleman Embedding

The third-order Carleman construction follows the same hierarchical approach. We track the time evolution of each term in the original nonlinear equations by applying the derivative rule to express new terms using quantities whose evolution is already known. The key difference is that we truncate at third order, discarding all fourth-order and higher contributions.
Consider terms of the form $v_i \otimes v_j$ from equations (5.27)–(5.29). The product rule gives

$$\partial_t(v_i \otimes v_j) = \partial_t(v_i) \otimes v_j + v_i \otimes \partial_t(v_j). \tag{5.39}$$

Substituting the expressions for $\partial_t v_{i/j}$ yields second-order terms plus third-order contributions $v_i \otimes v_j \otimes v_k$. Differentiating again,

$$\partial_t(v_i \otimes v_j \otimes v_k) = \partial_t(v_i) \otimes v_j \otimes v_k + v_i \otimes \partial_t(v_j) \otimes v_k + v_i \otimes v_j \otimes \partial_t(v_k). \tag{5.40}$$

We retain only first-order components of $\partial_t v_i$, as higher-order components would generate fourth-order terms that violate our truncation scheme. The explicit terms composing the third-order Carleman matrix are given in Appendix B The Carleman vector consist in a combination of 3 N-component vectors, 6 (N, N) tensors and 18 (N, N, N) tensors. We can thus approximate the Carleman matrix to be a $(3N + 6N^2 + 18N^3)^2$ matrix, only for a third order approximation.

5.5 Numerical Simulations

To evaluate the Carleman embedding accuracy, we performed two simulations on a one-dimensional box discretised into 32 equally spaced points. One used second-order truncation, the other third-order. Both adopted a soft constraint with $\gamma = 100$ and a Runge–Kutta integrator with adaptive timestep. Figure 5.10 shows the relative error for both truncation levels. We quantify error using the L_∞ norm, comparing the Carleman solution v_C with a reference state v_{NL} obtained by integrating the original nonlinear equations (Sec. 5.4.2):

$$\frac{\|v_C - v_{\mathrm{NL}}\|_{L_\infty}}{\|v_C\|_{L_\infty}} = \frac{\max_{i\in[0,N]} |(v_C)_i - (v_{\mathrm{NL}})_i|}{\max_{i\in[0,N]} |(v_C)_i|}. \tag{5.41}$$

Figure 5.11 shows how key physical quantities evolve under both truncation schemes, with the nonlinear solution as a reference. The third-order approximation clearly outperforms the second-order one. Together, Figs. 5.10 and 5.11 quantify the observed discrepancies. However, even third-order truncation fails to achieve high-fidelity results for the Schrödinger–Poisson system. The computational cost of Carleman linearization grows exponentially with truncation order. Since our dynamics only extend to $t = 0.3$—before strong nonlinear phenomena such as shell crossing emerge—significantly higher orders would be needed, quickly becoming computationally prohibitive.

5.6 Conclusions

In this chapter, we explored potential improvements to the variational algorithm for solving the SP system introduced in Chapter 4. While adopting a more suitable ansatz and

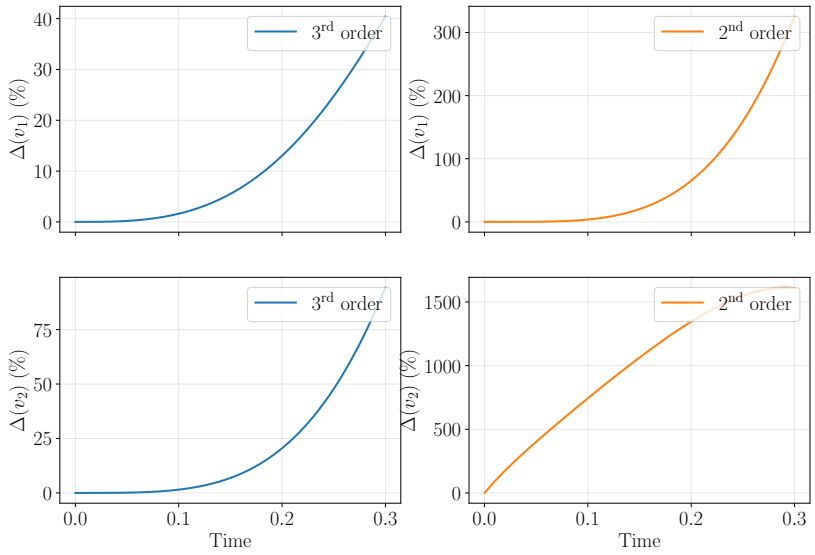


Figure 5.10: Relative error comparison for the real and imaginary part of ψ as a function of the time for different Carleman embeddings. The simulations are obtained with $\gamma = 100$ on a 32 points grid. In the first column the results with a third order approximation; in the second column the result with a second order. The error is obtained using Eq. (5.41).

an enhanced Runge-Kutta integration scheme with adaptive timestep did not allow us to extend the simulation scale, the analysis provides valuable insights into the limitations of current variational strategies for high-dimensional problems.

A key challenge lies in the regularization of the matrix in Eq. (4.8), which governs the system's temporal evolution. Various SVD-based regularization techniques were tested, but no method proved simultaneously robust, general, and independent of the initial conditions. Moreover, the number of measurements required to sample the known term scales as $\mathcal{O}(2^n\epsilon^{-2})$, with ϵ denoting the desired precision, highlighting the practical constraints of the approach.

We also considered an alternative method based on the linearized SP system using Carleman embedding, deriving second- and third-order truncations. Although these results were not immediately promising for large-scale simulations, they provide guidance on which directions may be less effective for direct quantum implementations.

Overall, our analysis highlights the limitations of current variational approaches for the SP system. In particular, the strong nonlinearity of the SP dynamics appears to constrain the potential advantage of quantum computing for direct simulations, both for variational and direct algorithms. At the same time, we do not rule out the possibility that more sophisticated hybrid approaches could exploit quantum resources more effectively in the future. Rather than closing any avenues, this work shifts the focus toward developing refined algorithms that may achieve a practical quantum advantage, as explored in the following chapter for QFRANS. By identifying and understanding these limitations, the study helps guide future efforts toward approaches that could realistically leverage quantum computing for the SP system.

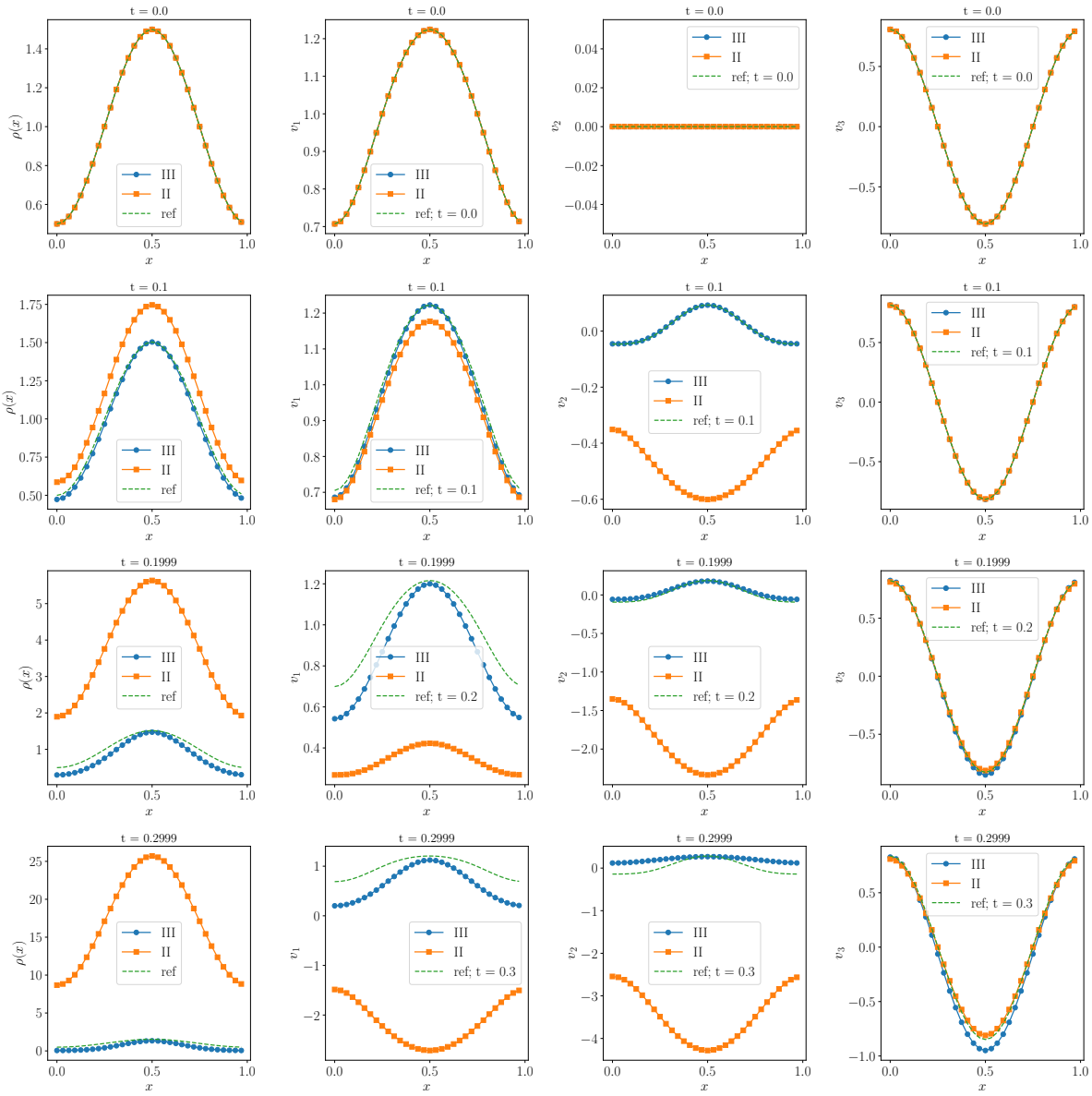


Figure 5.11: The picture is to be read in columns, from left to right we have the evolution of the density, the real and imaginary part of ψ (v_1, v_2) and the potential (v_3). The results are obtained using Carleman at third order ('III'), second order ('II') and solving the nonlinear system (dotted line).

Chapter 6

QFRANS: Quantum Fixed RAdius Neighbor Search

In this chapter, we shift focus from direct dark matter simulations and propose a quantum algorithm capable of accelerating the computationally intensive components of classical cosmological simulations. Our attention is specifically directed toward the Fixed-RAdius Neighbor Search (FRANS) problem. This module is a computationally demanding task, usually both time- and memory-consuming that is present in many N-body methods, and specifically in cosmological simulations. The main issue with this kind of algorithms, from the perspective of HPC performance analysis, is the long execution time due to cache misses. In this chapter, we aim to find a solution to the FRANS problem by employing our own version of the oblivious-fixed-point amplification algorithm [120], [121], which is very adaptable to different initial states while being agnostic to the target state. In Section 6.2 we introduce the problem and describe its difficulties when treated by a classical computer. In Section 6.3 we describe the Grover's algorithm and its Fixed-Point-Search (FPS) and Oblivious-Amplitude-Amplification (OAA) versions. In Section 6.4 we describe our quantum FRANS algorithm and we highlight the similarities and differences with the previous algorithms. Hence, we explicitly show the quantum circuit and the gate decomposition of the various operations, analyzing the computational complexity and the gate complexity. Furthermore, we propose a stopping criterion based on Bayesian's posterior probability. In Section 6.5 we perform numerical tests to verify the goodness of our stopping criterion. In Section 6.6 we introduce the bit-flip and the readout measurement errors in our simulation, calculating the threshold for the amount of noise required to preserve the quantum advantage of the algorithm, without introducing particular error correction schemes. Finally, we draw the outlooks and the conclusions of this work in Section 6.7.

6.1 Introduction

In computational science, a seemingly straightforward task has emerged as a fundamental limitation: finding all particles within a fixed distance between each other. This Fixed-RAdius Neighbor Search (FRANS) [122] is strictly related to N-body methods and forms the computational backbone of numerous simulation methods across physics, chemistry, and astronomy [123]–[127], yet consistently dominates runtime despite decades of algo-

rithmic advancements.

The naive approach to finding all particles within a cutoff distance ξ requires examining every possible particle pair, resulting in $\mathcal{O}(N^2)$ scaling that quickly becomes unfavorable as the system size grows. Modern simulations employ specialized spatial data structures such as uniform grids, Verlet lists [123], [126], [128], and hierarchical trees [65], [125], which can reduce the complexity to $\mathcal{O}(N \log N)$ by considering only a portion of all the possible pairs.

Fast algorithms used to explicitly find all the particle pairs within the fixed-radius scale as $\mathcal{O}(M + N)$, where M is the number of neighboring pairs, with a preprocessing of $\mathcal{O}(N \log N)$ and a $\mathcal{O}(N)$ space complexity [129]. Yet, in spite of the improved computational cost, this optimized neighbor search still constitutes the primary computational expense across diverse scientific applications.

In the context of Molecular Dynamics (MD), Verlet lists or cell indices [123] are used to evaluate interactions between atoms for the study of a wide range of phenomena such as drug binding [130], [131], membrane dynamics [132], [133] and material science [134]. These data structure keeps track of the atoms that are spatially close, in order to avoid considering atoms far beyond the interaction radius. The FRANS procedure and maintenance of these lists typically consumes from 30% to 60% of total runtime in large-scale simulations [135], [136]. Another interesting field of MD concerns protein folding. Research in this area focuses on how proteins evolve towards an equilibrium state (folding) and interact with one another. A noticeable application is found in neuroscience, where abnormal folding patterns often signal neurological diseases [137]. While earlier studies relied on neighbor search [138], the field has evolved considerably with new approaches. AlphaFold [139] stands out as a particularly valuable tool, serving as a detailed simulator and providing a database of 200 million proteins' structure. In this case, neighbor search algorithms remain useful for homology detection [140], helping researchers identify structural similarities between proteins to trace evolutionary origins and primitive forms. These methods also assess how well different proteins might interact based on their geometric properties [141]. Non-equilibrium protein folding [142]–[144] is an open research field, also related to neurological disorder [145], [146] and represents another area of interest for neighbor search. Here environmental influences cause proteins to adopt conformations outside thermal equilibrium. Since ground-state assumptions do not apply in these systems, different methods are used to model the folding mechanism such as coarse-grained molecular dynamics simulations combined with FRANS.

In both cosmology and fluid dynamics, Smoothed Particle Hydrodynamics (SPH) [124] can be used to simulate the evolution of fluids and gases. This approach represents continuous media as discrete particles, each carrying fundamental properties such as mass, momentum and energy. Particle interactions emerge through a weighting scheme that employs localized kernel functions to determine the influence of neighboring particles. The method achieves conservation of field quantities (e.g, pressure and density) by spatially averaging contributions from all particles within the kernel's smoothing length radius. The evaluation of neighboring pair and interaction between particles takes up most of the execution time [147]. The challenge becomes particularly acute in simulations with free surfaces or multiphase flows, where particle distributions become highly non-uniform.
In the context of astrophysical N-Body simulations, SPH is used to simulate dust dynamics using tree-based methods for the neighbor search part. It is common to consider in these

scenarios the nearest n neighbors (typically $n = 64$) rather than all particles within a fixed radius [148]. While this can introduce limitations to the convergence of the scheme [149], it remains a practical approach.
For purely gravitational N-body simulations, hybrid Tree-Particle-Mesh methods are often employed to achieve $\mathcal{O}(N \log N)$ scaling, leveraging techniques such as space-filling Peano-Hilbert ordering, Oct-trees, and multi-pole expansions. However, these methods do not explicitly compute interactions for all neighbor pairs within a fixed radius, and thus do not fully fall into the FRANS category. In both cases, the traversal of these spatial data structures to identify nearby particles still dominates computational expense, consuming most of simulation's time in cosmological models with billions of bodies [125], [150]. Highly clustered mass distributions – characteristic of galaxy formation – further amplify these costs by necessitating frequent tree rebuilds.

What makes this problem particularly resistant to optimization is not merely its algorithmic complexity, but rather the challenging computational patterns it creates. Specifically, the combination of a large data structure combined with the frequent and irregular memory access patterns.

To facilitate the data transfer to and from the compute cores, modern computing architectures are built around a memory hierarchy with components of growing size and decreasing bandwidth, that performs best when data access follows predictable patterns. We report as reference the data gathered from two modern supercomputers [151], [152]: CPU registers $\rightarrow$ L1 cache, with $32 - 64$ KiB per core; $\rightarrow$ L2 cache, with 512KiB -8 MiB per core; $\rightarrow$ L3 cache, with $120 - 256$ MiB per core; $\rightarrow$ main memory However, neighbor search fundamentally involves unpredictable memory access and particles that are close in space configuration might be stored far apart in memory, particularly after many simulation time steps have caused the particles to move from their initial positions. This creates numerous cache misses, as the processor constantly needs to fetch data from slower memory tiers. For reference, in both SPH and MD the number of neighbors M depends on the application, as high resolution simulations might require higher values. As an example a high-resolution 3D SPH simulation requires approximately $N = 10^{10}$, with up to 100 target neighbors per particle [153], while an MD simulation might require $N = 10^9$ with the number of neighbors in the order of hundreds [126], [154], [155].

In Fig. 6.1 a toy model representing the cache miss problem is presented. When searching for neighbors of particle B, the algorithm needs to access data for particles E, G, and H that happen to be spatially nearby. However, due to limited cache memory capacity, only particles within B's sub-quadrant are currently stored in cache. To retrieve information for the remaining particles, the algorithm must fetch data from main memory—a significantly slower operation. The frequent repetition of this process creates substantial computational overhead and dramatically increases execution time. Thus, the mismatch between physical proximity and memory proximity causes the FRANS to be the bottleneck in many different scenarios. In practice, however, this issue can be mitigated by reorganizing particle data in memory, with the possibility of placing spatially nearby particles in contiguous memory locations.

At the dawn of quantum computing, Grover's search algorithm [156] showed one of the first example of a quantum algorithm that can run faster than any classical algorithm designed for the same task. While the computational complexity of a classical search is $\mathcal{O}(N)$, Grover's algorithm can do the search with only $\mathcal{O}(\sqrt{N})$ steps, thus offering a

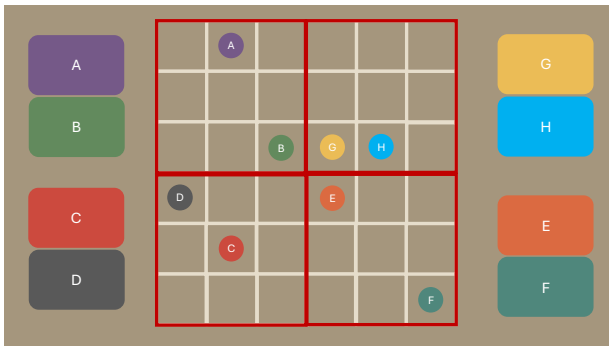


Figure 6.1: Toy model illustrating the cache miss problem in neighbor search algorithms. An evenly spaced grid contains eight particles distributed across four memory blocks (indicated by red boundaries).Each particle's data is stored in memory locations denoted by matching colors. Note that spatially adjacent particles might be stored in different memory locations.

quadratic advantage. After its appearance in the literature, Grover's algorithm has been analyzed and modified several times to match more specific tasks and to tackle the inherited issues of the original algorithm. Among those, we mention the Fixed-point search (FPS) algorithms [120], [157], [158], thought to tackle the periodic increase and decrease in the success probability of the search, and the Oblivious Amplitude Amplification algorithms(OAA) [121], [159], [160] intended to overcome the issue of not knowing the target state in advance. We will discuss both approaches later in the main text. In more recent years, it has been shown how Grover's search algorithm and all of its many versions belong to the class of Quantum Singular-value Transform (QSVT) algorithms [161], [162], which reveals the principles behind the most widely used (and cited) quantum algorithms.

6.2 The fixed-radius neighbor search

At the heart of many computational models in physics lies a conceptually simple yet computationally intensive operation: determining which objects in space are close to one another. This procedure, known as neighbor search, forms the foundation for virtually all particle-based simulation methods across the physical sciences.

In its most fundamental form, the fixed-radius variant, neighbor search answers a straightforward question: given a collection of particles distributed throughout space, which particles lie within a fixed distance of each other? This critical calculation enables the modeling of interactions that occur only between objects in close proximity—forces that diminish rapidly with distance, collisions between bodies, or influence that spreads within a limited radius.

The mathematical formulation is deceptively simple. For each particle i in a system of N particles, we must identify all other particles j such that the distance between them $d(r_i, r_j)$ is less than some threshold distance ξ. This distance threshold might represent the cutoff radius of a potential energy function in MD, the smoothing length in fluid simulations.

Despite its conceptual simplicity, this operation presents an extraordinary computational challenge. The most direct approach—checking every possible pair of parti-

cles—would require examining N^2 distance calculations, a prohibitive scaling for systems of scientific interest that may contain millions or billions of particles. Moreover, as particles move throughout a simulation, these neighborhood relationships continually change, requiring repeated recalculation.

To address this challenge, researchers have developed sophisticated spatial data structures that organize particles based on their positions in space. Cell lists divide the simulation domain into a grid, allowing the search to focus only on particles in adjacent cells. Verlet lists maintain a precomputed set of neighbors for each particle, including a buffer zone to reduce update frequency. Tree-based methods hierarchically partition space, enabling rapid elimination of distant regions from consideration.

These techniques reduce the theoretical complexity from $O(N^2)$ to $O(N \log N)$, representing an enormous computational saving. Nevertheless, neighbor search still typically accounts for $30 - 70\%$ of total computation time in production simulations across disciplines [135], [136], [147]. The persistent challenge stems from a fundamental disconnect between two different types of proximity: spatial proximity in the physical simulation and memory proximity in the computing hardware.

When particles are close to each other in the simulation space, their data should ideally be close to each other in the computer's memory for efficient processing. However, this alignment rarely occurs, especially as particles move throughout the simulation. A particle's neighbors in physical space are often scattered across distant memory locations, forcing the processor to constantly fetch data from widely separated memory addresses.

Compounding this problem is the repetitive nature of the search operation. For each of the N particles in the system, the algorithm must perform a separate neighbor search, repeatedly accessing memory locations across the entire data structure.

The combined effect of these two factors – spatial-memory misalignment and repeated broad memory access – frequently results in cache misses, where the processor must wait to retrieve data from slower memory tiers rather than finding it in fast cache memory. These waiting periods significantly increase actual execution time, often by an order of magnitude or more compared to theoretical predictions.

Thus, the primary challenge in modern neighbor search implementations has shifted from algorithmic complexity to hardware efficiency. Even algorithms with optimal theoretical scaling ($O(N)$) can perform poorly in practice due to their incompatibility with contemporary computing architectures and the true complexity now lies not in the mathematical operation count but in navigating the complex memory hierarchy of modern processors to minimize data movement costs.

Estimating the impact of cache misses on the FRANS solution is a complex issue that falls beyond the scope of this work. Such an assessment depends not only on the algorithmic implementation but also on the underlying hardware architecture and specific software realization. Additionally, factors such as dataset size, memory management strategies, and data distribution across computing resources vary significantly between different versions and implementations. Nevertheless, to provide context on how this issue impacts modern simulations, we present a simplified estimate.
Consider a large cosmological simulation with 10^{10} particles executed on Fugaku supercomputer [151]. It has been reported that in a 32^3 particles SPH simulation optimized for single-node execution on Fugaku exhibits cache miss probabilities for $L1 = 1.2\%$ and for $L2 = 0.12\%$ [163]. Building on these data, we aim to estimate cache behavior for a

10^{10}-particle simulation under a simplified capacity-dominated scaling model, where the miss rate scales proportionally with the number of particles per node.
Fugaku comprises 158,976 compute nodes [151]. A reasonable allocation for a simulation of this magnitude would utilize approximately one-tenth of the available nodes (∼15,898 nodes) [164], with each node processing $\sim 6.29 \times 10^5$ particles. Applying a naive linear scaling law, the estimated miss rates increase to $L1 \simeq 23\%$ and $L2 \simeq 2.4\%$.
These estimates provide a first-order approximation of cache behavior, assuming that the per-particle data dimension remains independent of the working set size. However, for larger datasets, a significantly greater fraction of particles resides in RAM compared to the 32^3 particles-per-node case, thereby increasing the probability of accessing particles stored in main memory rather than cache. This type of miss is particularly costly, as retrieving data from main memory requires substantially more CPU cycles than cache access. During such memory fetches, the entire computational pipeline stalls while awaiting the memory refill operation.

Given these considerations, we regard our estimate as an upper bound on the actual cache miss percentage. We emphasize that actual performance may differ substantially depending on code-specific factors, particularly the strategies employed for managing and accessing larger datasets in practice. For instance, codes written for molecular dynamics do not translate well to the astrophysical domain due to fundamentally different data layouts: Verlet lists are optimized for short-range interactions, whereas in cosmological simulations data are clustered in distinct spatial regions separated by cosmic voids. Moreover, state-of-the-art codes employ sophisticated techniques such as Peano-Hilbert space-filling curves [157], [165] to ensure that spatially proximate data occupy adjacent memory locations, thereby minimizing cache misses. However, the focus of this work is not to compete with these highly optimized classical methods, rather, we aim to introduce the fundamental elements of a quantum algorithm for the FRANS problem. The purpose of this analysis is precisely to lay the groundwork for a quantum algorithm. Classical methods have been studied and refined for many years, and attempting to directly compete with them at this stage would be unwise. Instead, our approach establishes the foundations for a future scenario where the combination of technological development and algorithmic design could potentially lead to a quantum advantage over classical routines.

6.3 Quantum search algorithms

In this section, we describe a few versions of the Grover's algorithm as they are relevant to this paper. In the original work, the N elements of a database D are encoded into a quantum state $|i\rangle$ and we want to select a specific target state $|\psi_T\rangle$. The initial state is set as the uniform superposition of all possible states

$$|\psi_0\rangle = \frac{1}{\sqrt{N}} \sum_{i \in D} |i\rangle. \tag{6.1}$$

The algorithm consists in applying to the state $|\psi_0\rangle$ first an oracle operator $\hat{O}$, which is capable of recognizing the solution and whose action is to flip the sign of the target state $|\psi_T\rangle$, defined as

$$\hat{O} = \mathbb{I} - 2|\psi_T\rangle\langle\psi_T|, \tag{6.2}$$

and later applying the reflection operator $\hat{R}_0$, which reflects the resulting state with respect to the initial state $|\psi_0\rangle$, namely

$$\hat{R}_0 = 2|\psi_0\rangle\langle\psi_0| - \mathbb{I}. \tag{6.3}$$

Conveniently, the algorithm can be represented in a two-dimensional space spanned by the solution $|\psi_T\rangle$ and its orthogonal part $|\psi^\perp\rangle$ in the initial state, defined such that

$$|\psi_0\rangle = \cos\theta|\psi^\perp\rangle + \sin\theta|\psi_T\rangle, \tag{6.4}$$

where the angle $\theta = \arcsin\left[\sqrt{\frac{M}{N}}\right]$ and M is the number of target states in the database. The application of the Grover operator $\hat{G} = \hat{R}_0\hat{O}$ rotates the vector in the $|\psi_T\rangle, |\psi^\perp\rangle$ plane by an angle 2θ increasing the amplitude of the target state. By applying G a sufficiently large number of times $k \sim \sqrt{\frac{N}{M}}$, the final state $G^k|\psi_0\rangle$ will have the maximum amplitude for the component $|\psi_T\rangle$.

Now we see that there are two main hurdles when one aims to apply the Grover's search algorithm to its database. First, if the value M is unknown, the risk is to apply G too many times, with the result that the final state passes over the $|\psi_T\rangle$ axis and the amplitude probability of measuring the target state reduces. This is known as the *soufflé problem*, since applying the Grover's operator too few times undercooks the state, whereas applying it too many times overcooks it, with the resulting deflation of the quantum cake. This obstacle was overcome by the FPS algorithm [120], which increases monotonically the probability of measuring a target state. Second, as expressed in Eq. (6.2), the target state has to be known in order to construct the correct oracle operator $\hat{O}$. Although this may be sensible in the context of quantum simulation to construct a particular quantum state, it loses its significance when the quantum algorithm is intended to search over a classical database or to solve classical optimization problems. The OAA algorithm was designed to overcome this problem. In the remainder of this section we are going to summarize these two algorithms.

6.3.1 FPS algorithm

The FPS algorithm [120] introduces an ancilla qubit to reproduce a non-unitary dynamics that damps out the oscillations of the results of Grover's algorithm between the target and non-target states. The algorithm uses a series of parametric rotations

$$R_y(\alpha_i) = \begin{pmatrix} \cos\frac{\alpha_i}{2} & -\sin\frac{\alpha_i}{2} \\ \sin\frac{\alpha_i}{2} & \cos\frac{\alpha_i}{2} \end{pmatrix}, \tag{6.5}$$

where the optimal value of the α_i angles depends on the specifics of the problem. The algorithm is presented in Fig. 6.2 and reads as follows

$$\Pi_{i=1}^{\mathrm{K}}\left[\left(|0\rangle\langle 0|\otimes\mathbb{I}+|1\rangle\langle 1|\otimes\hat{R}_0\right)\left(R_y(-\alpha_i)\otimes\mathbb{I}\right)\right)\left(|0\rangle\langle 0|\otimes\mathbb{I}+|1\rangle\langle 1|\otimes\hat{O}\right)\left(R_y(\alpha_i)\otimes\mathbb{I}\right)\right)\Big]|1\rangle|\psi_0\rangle, \tag{6.6}$$

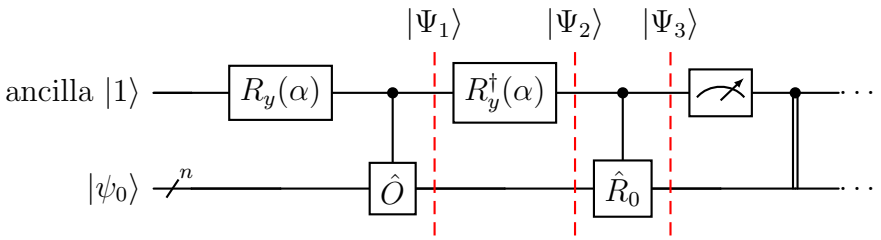


Figure 6.2: Circuit used for the amplitude amplification process. Based on Mizel search algorithm. The parameter α is adjusted accordingly to the iteration. The measurement on the ancilla qubit serves as control, if $|0\rangle$ is the outcome, it means that the quantum register is ready for measurement, otherwise repeat the process, varying the angle α.

where the circuit is run a total number of K times depending on the result of a measurement performed on the ancilla qubit as we see next.

The first step of the algorithm rotates the ancilla qubit to $-\sin\frac{\alpha_i}{2}|0\rangle + \cos\frac{\alpha_i}{2}|1\rangle$. The second term performs a controlled operation which yields the state

$$|\Psi_1\rangle = -\sin\frac{\alpha_i}{2}|0\rangle|\psi_0\rangle + \cos\frac{\alpha_i}{2}|1\rangle(\cos\theta|\psi^\perp\rangle - \sin\theta|\psi_T\rangle). \tag{6.7}$$

The application of the second single-qubit rotation $R_y(-\alpha_i)$ leads to

$$\begin{aligned}|\Psi_2\rangle &= -\sin\alpha_i\sin\theta|0\rangle|\psi_T\rangle + \sin^2\frac{\alpha}{2}|1\rangle|\psi_0\rangle + \cos^2\frac{\alpha}{2}|1\rangle\big(\cos\theta|\psi^\perp\rangle - \sin\theta|\psi_T\rangle\big)\\ &= -\sin\alpha_i\sin\theta|0\rangle|\psi_T\rangle - \cos\alpha_i\sin\theta|1\rangle|\psi_T\rangle + \cos\theta|1\rangle|\psi^\perp\rangle.\end{aligned} \tag{6.8}$$

Now the action of the last controlled operation $\big(|0\rangle\langle 0|\otimes\mathbb{I} + |1\rangle\langle 1|\otimes\hat{R}_0\big)$ on $|\Psi_2\rangle$ is given by

$$\begin{aligned}|\Psi_3\rangle = &-\sqrt{p_{i+1}}|0\rangle(|\psi_T\rangle) + \sqrt{1-p_{i+1}}|1\rangle\big((c_i\cos 2\theta - s_i\cos\alpha_i\sin 2\theta)|\psi^\perp\rangle\\ &+ (c_i\sin 2\theta + s_i\cos\alpha_i\cos 2\theta)|\psi_T\rangle\big),\end{aligned} \tag{6.9}$$

where we have used $p_{i+1} = \sin^2\alpha_i\sin^2\theta$, $c_i = \cos\theta$ and $s_i = \sin\theta$. Finally, depending on the outcome of the measurement performed on the ancilla qubit, we decide to stop or continue the algorithm. In fact, if we measure the state $|0\rangle$ on the ancilla qubit, the system collapses into the state $|\psi_T\rangle$. Conversely, if the outcome is $|1\rangle$, the system becomes a new superposition of the states $|\psi_T\rangle, |\psi^\perp\rangle$.

At the i-th repetition, the superposition state associated with $|1\rangle$ is

$$|\psi_{i+1}\rangle = c_{i+1}|\psi^\perp\rangle + s_{i+1}|\psi_T\rangle, \tag{6.10}$$

with the new coefficients defined as

$$\begin{aligned}s_{i+1} &= \frac{c_i\sin 2\theta + s_i\cos\alpha_i\cos 2\theta}{\sqrt{1-p_{i+1}}}\\ c_{i+1} &= \frac{c_i\cos 2\theta - s_i\cos\alpha_i\sin 2\theta}{\sqrt{1-p_{i+1}}}\end{aligned} \tag{6.11}$$

for $i = 1, \ldots, \mathrm{K}$ and

$$p_{i+1} = s_i^2 \sin^2 \alpha_i, \tag{6.12}$$

being the probability of measuring the ancilla qubit on $|0\rangle$.

Note that, depending on the choice for the sequence of the angles α_i, the instantaneous probability p_i may oscillates or even reduce at each iteration, but the cumulative probability p_{cum}, that is the probability of getting the outcome $|0\rangle$ at least once after after i iterations, is always non-decreasing. This is obtained by considering the probability of the complementary event, i.e. not measuring the desired outcome in the first i iterations.

$$p_{\mathrm{cum}}(i) = 1 - \prod_{j=1}^{i} \left(1 - p_{j+1}\right) \quad \text{for } i = 1, ..., K \tag{6.13}$$

The average number of oracle calls before success is given by:

$$< n_{\mathrm{calls}} >= \sum_i i \times \left[p_{i+1} \prod_{j=1}^{i-1} (1 - p_{j+1})\right], \tag{6.14}$$

where the term between squared brackets is the probability of having success exactly at the i-th iteration.

A sensible choice of the α_i leads the cumulative probability of success to rapidly grow to 1. In the case of knowing beforehand the value of M, and consequently of θ, the best choice is given by the constant critical value

$$\alpha_C = \arccos\left(\frac{1 - \sin 2\theta}{1 + \sin 2\theta}\right), \tag{6.15}$$

which has been found to be the optimal angle dampening the oscillations of Grover's algorithm between target and nontarget states[120]. When M is not known in advance but $M \ll N$, then the critical angle computed for $M = 1$ is an effective choice for a large set of values of M near 1. It is noteworthy to mention that the quantum counting algorithm has been proposed as a method to efficiently count the number of solutions that attain specific requirements of a given quantum search problem or simply counting the total number of solutions [166]. More recently, a new method has been proposed to estimate M by exploiting the relationship between this value and the probability of measuring one solution after a given number of iterations of the standard Grover's algorithm [167].

We show next, for the case of $M = 1$ and $N = 1000$, the evolution of the coefficients c_i, s_i together with the instantaneous and cumulative probabilities $p_i, p_{\mathrm{cum}}(i)$ for different choices of α_i. In the case of using a constant critical angle, shown in Fig. 6.3c, the probability of success at each iteration p_i tends to increase. However, when the number of target states M is not known but it is supposedly very large, an effective choice is given by varying the angle in the following decreasing way

$$\alpha_1 = \pi/2, \quad \alpha_i = \arccos\left(\frac{1 - \sin(\pi/2i)}{1 + \sin(\pi/2i)}\right) \quad \text{for } i > 1, \tag{6.16}$$

for which the dynamics is plotted in Fig. 6.3d. In Fig. 6.3b, we compare the cumulative probabilities and the average number of oracle calls $< n_{calls} >$ before success (vertical lines) for different choices of α (critical and decreasing) with the probability of classical

random extraction (dashed line). Comparing the two different choices of α, the cost of not knowing M weights at most by a factor of 1.5 in the average number of oracle calls. Summing everything up, in Fig. 6.3a we compare the average number of oracle calls for the case $M = 1$ as a function of N for the critical and decreasing cases with respect to the references $\sqrt{N/M}$ and N/M. As expected, we find that the fixed point search algorithm has a scaling $\mathcal{O}(\sqrt{N/M})$, an improvement with respect to the classical $\mathcal{O}(N/M)$ full stop.

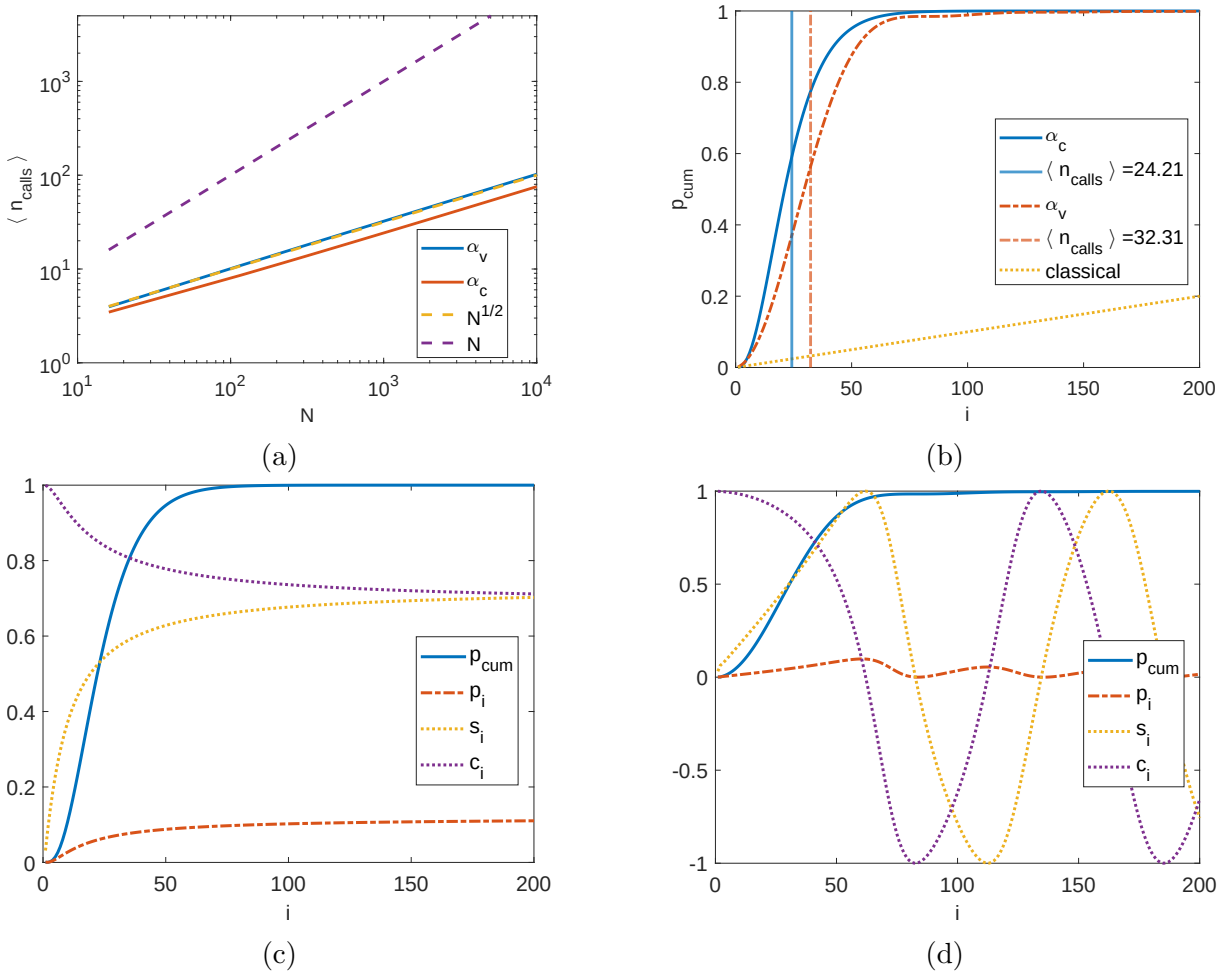


Figure 6.3: In (a), the average number of oracle calls before success (using $M = 1$ and changing the database size N) for the constant critical value of $\alpha_C = \alpha_i$ and for the variable decreasing sequence, defined in (6.16) and referenced as α_V. This is compared with the references $\sqrt{N}$ and N using a logarithmic scale on both the horizontal and vertical axes. The fixed point search algorithm provides a quadratic advantage with respect to classical search, for the considered angle's schedules. In (b), a comparison between the cumulative probabilities for the two choices with respect to a classical algorithm. The vertical lines represent the average number of queries to the oracle for the respective cases. In (c) and (d) the time evolution of the coefficients in Eq. (6.11), the instantaneous and cumulative probabilities p_i, p_{cum} for the case $M = 1$, $N = 1000$ with α_C and α_V respectively.

6.3.2 OAA algorithm

In order to apply either the Grover or the FPS algorithm, we need to know how to prepare the oracle $\hat{O}$ as expressed in Eq. (6.2), which assumes some knowledge of the target state $|\psi_T\rangle$. Most of the times, we do not have all the pieces of information about the target state, but some of its values, which are stored in a subspace that labels if the quantum state is (or it is not) the target. Namely, we can think of a unitary operation $\hat{U}$ such that

$$\hat{U}|0\rangle|\psi\rangle \quad = \quad |0\rangle V|\psi\rangle + |\Phi^{\perp}\rangle \tag{6.17}$$

where $|\psi_T\rangle = V|\psi\rangle$ has the label qubit set to $|0\rangle$ and $|\Phi^{\perp}\rangle$ is a state orthogonal to $|0\rangle|\psi_T\rangle$ with different value of the label qubit. The OAA algorithm allows the amplification of the state labeled by $|0\rangle$ regardless of the quantum state $|\psi_T\rangle$. This is a typical condition that we encounter in block-encoding algorithms [161] where the relevant dynamics is labeled by the $|0\rangle$ state.

We refer to the original paper [159] for the description of the algorithm. However, as a caveat, we remind that the OAA algorithm is exact only if the V operator is unitary [160], otherwise introducing errors into the target state.

6.4 Quantum FRANS algorithm

In this section we present our Quantum algorithm for the FRANS problem (QFRANS), which is obtained as an efficient and ad-hoc modification of the FPS algorithm. Later in this section we present a full analysis of each component of the quantum circuit.

We consider a dataset $\mathbb{X} = \{i, x_i\}_{i=1}^{N}$ which collects the label i given to N particles and the respective position x_i. Our goal is to find all the pairs (i, j) such that the distance $d(i, j)$ is lower than the chosen fixed radius h.

The first part of the algorithm encodes two copies of the dataset in four quantum registers, by using bit-encoding. In particular, two quantum registers with $q_0 = \lceil \log_2 N \rceil$ qubits are used to encode the label of the particles, with the state $|i\rangle$ being the binary representation of the integer i, while q_1 qubits are used to encode their position by discretizing the whole space into a 2^{q_1}-points lattice, and letting the state $|x_i\rangle$ to represent the coordinates.

The preparation of the state is accomplished by employing the PREP operator, which acts on the position and label registers as

$$|\phi\rangle = \mathrm{PREP}|0\rangle_{q_0}|0\rangle_{q_1} = \frac{1}{\sqrt{N}} \sum_i |i\rangle_{q_0} |x_i\rangle_{q_1} , \tag{6.18}$$

where the suffix given to the ket details the number of qubits. QFRANS applies PREP to both the particles register, yielding

$$|\psi_0\rangle = \mathrm{PREP} \otimes \mathrm{PREP}|0\rangle_{q_0}|0\rangle_{q_1}|0\rangle_{q_0}|0\rangle_{q_1} = \frac{1}{N} \sum_{i,j} |i\rangle_{q_0} |x_i\rangle_{q_1} |j\rangle_{q_0} |x_j\rangle_{q_1} , \tag{6.19}$$

Once the dataset is encoded we evaluate the distance between all the points in the dataset $d_{ij} = d(x_i, x_j)$. This can be done by introducing a set a_1 of ancillary qubits

initialized in $|0\rangle$ and by applying a generic distance operator $\hat{D}$ on the two q_1-qubits registers

$$|\psi_1\rangle = (\mathbb{I}_{2q_0} \otimes \hat{D})|\psi_0\rangle|0\rangle_{a_1} = \frac{1}{N}\sum_{i,j} |i\rangle_{q_0} |x_i\rangle_{q_1} |j\rangle_{q_0} |d_{ij}\rangle_{q_1} |\cdot\rangle_{a_1}; \tag{6.20}$$

an efficient implementation of the operator $\hat{D}$ in case of euclidean geometry is given in Sec. 6.4.1.

The value $|\cdot\rangle_{a_1}$ of the ancilla qubits is not relevant at this stage of the calculation, but we cannot discard it as it will be useful later.

Once the state is prepared, according to Grover's algorithm and its FPS counterpart, now we construct the oracle operator that applies a negative phase to the states where $d_{ij} \leq h$.

Here resides the novelty of this work: we do not know what is our target state $|\psi_T\rangle$, so we do not aim to build the oracle operator as described in Eq. (6.2), nevertheless we build the oracle by constructing a diagonal operator which is able to invert the sign of all the possible solutions.

$$\hat{O}(h) = \mathrm{diag}\Big(\underbrace{-1,\ldots,-1}_{h \text{ times}}, \underbrace{1,\ldots,1}_{2^{(q_1+1)}-h \text{ times}}\Big). \tag{6.21}$$

When applied to $|\psi_1\rangle$ this yields

$$\begin{aligned} |\psi_2\rangle &= (\mathbb{I}_{2q_0+q_1} \otimes \hat{O}(h) \otimes \mathbb{I}_{a_1})|\psi_1\rangle \\ &= \sqrt{\frac{N-M}{N}} \sum_{d_{ij}>h} |i\rangle_{q_0}|x_i\rangle_{q_1}|j\rangle_{q_0}|d_{ij}\rangle_{q_1}|\cdot\rangle_{a_1} - \sqrt{\frac{M}{N}} \sum_{d_{ij}\leq h} |i\rangle_{q_0}|x_i\rangle_{q_1}|j\rangle_{q_0}|d_{ij}\rangle_{q_1}|\cdot\rangle_{a_1} \\ &= \cos\theta|\psi^{\perp}\rangle - \sin\theta|\psi_T\rangle, \end{aligned} \tag{6.22}$$

where M is the number of neighboring pairs and in the last line of Eq. (6.22) we restored the convention that uses $|\psi^{\perp}\rangle, |\psi_T\rangle$ and the relative angle θ.

Finally we build the reflection over the initial state $|\psi_1\rangle$, $\hat{R}_1$, similar to Eq. (6.3) as

$$\hat{R}_1 = 2|\psi_1\rangle\langle\psi_1| - \mathbb{I} = \hat{U}_1(2|0\rangle\langle 0| - \mathbb{I})\hat{U_1}^{\dagger} = \hat{U}_1\bar{Z}\hat{U_1}^{\dagger}, \tag{6.23}$$

where we defined $\hat{U}_1 = \hat{D}\hat{U}$, and $\bar{Z}$ as the expanded $\hat{Z}$ operation which inverts the sign of the qubits different from $|0\rangle$. The application of $\hat{R}_1$ to the quantum state leads to $|\psi_3\rangle = \hat{R}_1|\psi_2\rangle$. Upon this premises, we are able to build the QFRANS algorithm as depicted in Fig. 6.4, which is analogue to the FPS algorithm, with an explicit state preparation for the FRANS problem, and a modified oracle operator. Although those differences, the result can be written as in Eq. (6.9), with

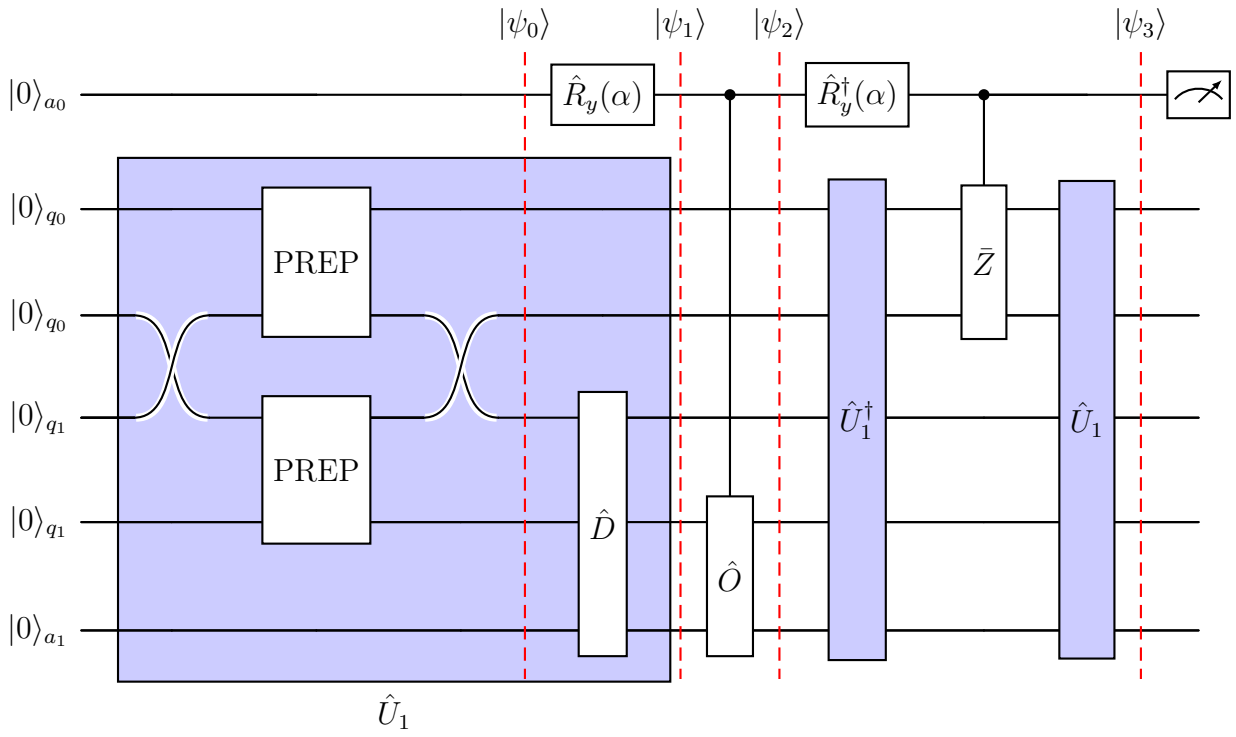


Figure 6.4: The circuit for the QFRANS algorithm, which has to be repeated until the state measured in the register a_0 is $|0\rangle$.

$$
\begin{aligned}
|\Psi_3\rangle = &-\sqrt{p_{k+1}}\sqrt{\frac{M}{N}}\sum_{d_{ij}\leq h}\left(|0\rangle_{a_0}|i\rangle_{q_0}|x_i\rangle_{q_1}|j\rangle_{q_0}|d_{ij}\rangle_{q_1}|\cdot\rangle_{a_1}\right)\\
&+\sqrt{1-p_{k+1}}\,|1\rangle_{a_0}\Big[(c_k\cos 2\theta - s_k\cos\alpha_k\sin 2\theta)\sqrt{\frac{N-M}{N}}\sum_{d_{ij}>h}(|i\rangle_{q_0}|x_i\rangle_{q_1}|j\rangle_{q_0}|d_{ij}\rangle_{q_1}|\cdot\rangle_{a_1})\\
&+(c_k\sin 2\theta + s_k\cos\alpha_k\cos 2\theta)\sqrt{\frac{M}{N}}\sum_{d_{ij}\leq h}(|i\rangle_{q_0}|x_i\rangle_{q_1}|j\rangle_{q_0}|d_{ij}\rangle_{q_1}|\cdot\rangle_{a_1})\Big],
\end{aligned}
\tag{6.24}
$$

In the remaining of this section we analyze each of the components of the algorithm separately.

6.4.1 State preparation

Here we describe the $\hat{U}_1$ operator needed to encode and process the pieces of information available in the database.

The fundamental premise of our algorithm relies on establishing a correspondence between dataset elements $\mathrm{X_i}$ and their integer representations x_i. When working with floating-point precision, this correspondence can always be achieved by defining a discretization grid based on the numerical precision of the computing system. With 64 qubits si possible to reproduce each number in the format float64, with precision 10^{-16}. It is in fact possible to convert a floating point into an integer in a classical machine; the

number of qubits necessary to reproduce such number are expressed by $\log_2$ of the number of classical bits. However, more sophisticated grid selection schemes can be tailored to specific problem requirements and search radius ξ to optimize performance.

To identify and mark the target states that satisfy the proximity condition in Eq. (6.30), we must first determine the integer representation of the chosen distance threshold ξ. This conversion is feasible precisely because we have established a bit-encoding scheme that associates each dataset point with a unique integer value.

A critical requirement of our approach is that the input dataset must be evenly spaced, or equivalently, that the data space can be accurately described using a regular grid structure. Under this constraint, we can interpret the integer representation as the number of discrete distance units required to span the threshold distance ξ. For illustration, consider the one-dimensional case where this relationship is expressed as:

$$h = \left\lceil \frac{\xi}{\Delta x} \right\rceil , \tag{6.25}$$

where Δx represents the grid spacing. This discretization procedure effectively transforms the original continuous proximity problem into an equivalent integer formulation:

$$d(x_i, x_j) \leq h \, , \quad x_i, x_j, h \in \mathbb{N} \, . \tag{6.26}$$

Hence, We assume that both the label and the position of each particle are described by integers. Note that the ceiling function in Eq. (6.25) is suitable to a proximity condition that uses the inclusive inequality as in Eq. (6.26), whereas, for applications requiring a strict inequality, the floor function should favorable.

The PREP operator

In the following we treat PREP as a black-box operator that is able to prepare the state $|\phi\rangle$ as defined in Eq. (6.18). The construction of PREP is often disregarded but it is actually one of the hardest element to deal with. The quantum superposition of N states can be obtained by employing a circuit with depth $\mathcal{O}(N)$, thus exponential with the number of qubits q_0 of the label register [168], [169]. This can be interpreted as the sequential implementation of each particle directly on the quantum circuit, with the $\log N$ ancilla qubits facilitating the decomposition of multi-controlled gates. Note that by using the optimized algorithm of Ref. [168], we can implement PREP as two separate circuit, as shown in Fig. 6.5. The first operator, $\hat{L}$, defines the balanced superposition of N elements in the register with q_0 qubits $\hat{L}|0\rangle_{q_0} = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle$. The second operator, $\hat{E}$, assigns to each particle its own position, $\hat{E}|i\rangle_{q_0}|0\rangle_{q_1} = |i\rangle_{q_0}|x_i\rangle_{q_1}$.
This implementation of PREP allows us to use a reduced $\hat{R}_1$ operator, as we will see later.

One can drastically reduce the depth of the circuit by increasing its width, i.e. by employing ancilla qubits [170]–[172], an approach that culminates with a circuit that has width$\sim \mathcal{O}(N)$ while keeping the depth$\sim \mathcal{O}(\log N)$. This constructs a register of N ancilla qubits containing the information for initial state preparation, effectively forming a database. Access to this repository scales logarithmically with the number of data points N, enabled by a tree-based data access structure. In our case, since the initial state has nearly uniform coefficients, the classical overhead for amplitude preparation is both constant and efficient. In general, different algorithms play with the tradeoff

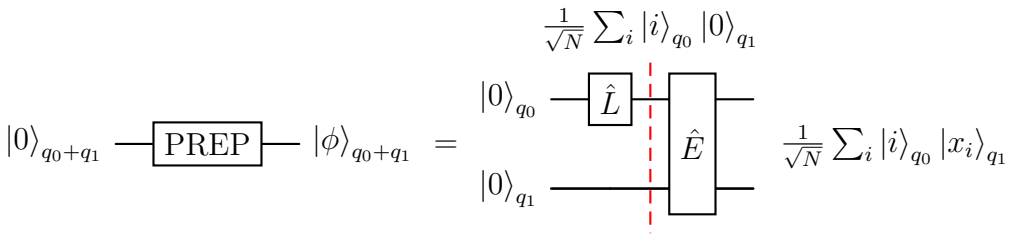


Figure 6.5: A possible implementation of the PREP operator, where $\hat{L}$ creates a balanced superposition of the labels of the particles and $\hat{E}$ assigns to each particle its position.

between depth and number of ancillas [171], while keeping the product of depth and width constant. In order to build a functioning PREP operator, one should adapt the depth and the width to the specifics of the quantum computer available. Neither of the approaches at the extremes of this trade-off require all-to-all qubit connectivity; consequently, we do not expect connectivity constraints to significantly impact the algorithm's scaling.

Conversely, if the position of the particles follows some structure, that can be described by a mathematical function, we can use matrix access oracles with depth scaling approximately as $\sim \mathcal{O}(\text{poly}\log N)$ [119], [172], [173].

The distance operator $\hat{D}$

In a generic scenario, the distance operator $\hat{D}$ is applied on a combination of two identical states superpositions $|\phi\rangle|\phi\rangle$

$$\hat{D}\sum_{i,j}|x_i\rangle_{q_1}|x_j\rangle_{q_1}|0\rangle_{a_1} = \sum_{i,j}|x_i\rangle_{q_1}|d_{ij}\rangle_{q_1+1}|0\rangle_{a_1-1}\,, \tag{6.27}$$

where $|d_{ij}\rangle_{q_1+1}$ is the state encoding the signed distance between the particle x_i and x_j. The extra qubit q_1+1 is borrowed from the ancillary register and used to represent the sign of the distance.

Consider a three-dimensional Euclidean space with L_2 distance

$$d(r_i, r_j) = \sqrt{(x_i-x_j)^2+(y_i-y_j)^2+(z_i-z_j)^2}\,. \tag{6.28}$$

The integer FRANS problem in Eq. (6.26) reduces to identifying particle pairs contained within a sphere of radius h. However, in certain scenarios, such as cosmological simulations [125], it is sufficient to determine whether particles fall within a cube of side length h (Chebyshev distance). This approach effectively assigns particles to discrete spatial cells, allowing one to quickly identify nearby particles without computing full Euclidean distances. In these simulations, the number of cells is typically so large that calculating the Euclidean norm for each comparison provides little practical benefit in terms of algorithmic logic, while significantly increasing computational cost. This is exactly the approach used in classical simulations, where checking which particles occupy the same or adjacent cubes efficiently determines potential interactions without evaluating all pairwise distances. This approach decomposes the problem into three independent one-dimensional searches along each Cartesian axis, where the distance equals the absolute difference of the coordinates.

$$d(r_i, r_j) \leq \epsilon \mapsto |x_i-x_j| \leq h \;\wedge\; |y_i-y_j| \leq h \;\wedge\; |z_i-z_j| \leq h\,. \tag{6.29}$$

This the case

Henceforth, we focus on the one-dimensional case, as the problem reduces to this simpler scenario when employing the Chebyshev distance.

Before delving into details regarding the implementation of the absolute difference, we must explain how negative integers are represented in binary. Negative integer values are represented in binary form using two's complement: the binary representation of the integer is negated and 1 is added, with the most significant bit (MSB) encoding the sign. For example, consider the integer 10 with binary representation 01010. We first negate the value to obtain 10101, then add 1 to yield 10110, the binary representation of -10. To recover the absolute value of a negative integer, we apply two's complement again: negate the bits to obtain 01001, then add 1 to recover 01010, corresponding to 10. Thus, obtaining the absolute value requires negation followed by adding 1 when the distance is negative.

In a quantum circuit, the difference between two binary values a and b is computed using the inverse Ripple Carry Adder [174], which requires $\mathcal{O}(q_1)$ CNOT gates and only 2 ancilla qubits: one for the carry of the subtraction and another as a clean ancillary qubit. When $a > b$, the carry qubit remains in state $|0\rangle$ and yields the correct result. When $a < b$, the carry qubit flips to $|1\rangle$, producing the two's complement representation of $a-b$. We thus treat the carry bit as the sign qubit and effectively as part of the distance.

To obtain the absolute value, when the sign qubit is $|1\rangle$, the incrementer $\widehat{+1}$ is applied to the distance register, performing the addition of unity: $\widehat{+1}\,|d_{ij}\rangle = |d_{ij}+1\rangle$. This controlled incrementer can be implemented with circuit depth $\mathcal{O}(q_1)$ [175], [176]. Subsequently, the value is negated via a series of q_1 $CNOT$ gates, each controlled by the sign qubit and acting on $|d_{ij}+1\rangle$.

However, instead of using the absolute difference between particles, we propose to work with the signed difference. The negative values will be then disregarded as non-target states by an appropriate choice of the oracle; this will be explained in Sec 6.4.2. This operation is considerably simpler, requiring only the inverse adder and avoiding the controlled operations. Moreover, the number of target states is reduced by half, as each pair is counted only once.

Summing up, the choice of the Chebyshev distance over the L_2 norm reduces the overall depth of the distance operator $\hat{D}$, as it allows the use of signed differences along each spatial dimension. In this context, $\hat{D}$ comprises three inverse adder operators, one acting along each spatial coordinate. This eliminates the squaring and summing operations required by the Euclidean distance [177], thereby reducing the overall complexity of the distance operator.

6.4.2 The oracle $\hat{O}$

The oracle presented in Equation (6.21) represents the most naive implementation, as it incorporates elements with zero distances. This becomes particularly valuable when different particles are mapped to identical positions due to finite precision constraints. It is also the cheapest option in terms of ancilla qubits, number of CNOT and circuit depth.

The most direct approach to implementing the oracle $\hat{O}$ employs the diagonal decomposition method described in Reference [178]. However, this approach exhibits unfavorable scaling properties, with both the number of CNOT gates and circuit depth scaling

as $\mathcal{O}(2^{q_1+1})$, rendering it impractical for large-scale implementations.

An alternative approach would be to implement the oracle as a series of multi-controlled Z gates (MCZ). In this scenario, we would require $\mathcal{O}(h)$ MCZ gates. Using the approach described in [179], [180] the global scaling of the oracle would be $\mathcal{O}(hq_1)$.

We can achieve a substantial improvement, reducing the complexity to $\mathcal{O}(q_1)$, by exploiting the inherent structure of our problem. In fact, we propose an oracle implementation that utilizes an integer comparator circuit (COMP), which requires $q_1 - 1$ clean ancilla qubits and one additional qubit to store the comparison result $d_{ij} \geq h$. When this condition is satisfied, the target qubit is found in state $|1\rangle$. The comparator's operation can be formally expressed as $\mathrm{COMP}\left(\sum_{ij} |d_{ij}\rangle_{q_1+1} |0\rangle\right) = \sum_{ij} |d_{ij}\rangle_{q_1+1} |d_{ij} \geq h\rangle$. Choosing the most significant qubit as the sign bit, we create a natural filtering mechanism. By adopting this choice, when we have a negative distance d_{ij}, the sign qubit is set to $|1\rangle$, which automatically makes the entire numerical value at least 2^{q_1}. Since all negative values are now greater than $2^{q_1} - 1$, the comparator naturally excludes them without any additional circuit. This elegant consequence of our sign bit placement simplifies the overall implementation.

To optimize circuit complexity, we implement the inverse oracle $-\hat{O}$ rather than the oracle defined in Equation (6.21). This is achieved through a phase kickback mechanism: we first apply the quantum comparator to the distance register, then apply a Z gate to the comparator's target qubit, and finally reset the target qubit to $|0\rangle$ by applying $\mathrm{COMP}^\dagger$. This procedure introduces a negative phase to all elements whose distance satisfies $d_{ij} \geq h$. Using this approach the Oracle requires only q_1 clean ancilla qubits.

However, when one is faced with large datasets, e.g, $N = 10^9$ particles, we suggest to employ a slightly different oracle

$$\hat{O}(h) = \mathrm{diag}\Big(1, \underbrace{-1, \ldots, -1}_{h-1 \text{ times}}, \underbrace{1, \ldots, 1}_{2^{(q_1+1)}-h \text{ times}}\Big). \tag{6.30}$$

where we introduced a modification to exclude zero-distance values $|0\rangle$. This eliminates comparisons between identical data points, reducing the number of solutions by N. In practical applications, e.g, cosmological simulations where $N \sim 10^{10}$, this optimization provides significant computational advantages.

Figure 6.6 illustrates the quantum circuit implementation of $-\hat{O}$ from Equation (6.30). Zero values are excluded from the marked states by replacing the standard comparator COMP with $\mathrm{COMP}\,(\widehat{-1})$, where $\widehat{-1}$ represents the inverse quantum incrementer based on a logarithmic depth quantum adder [176]. To maintain linear scaling in both CNOT count and circuit depth with respect to q_1, this modified approach requires $2q_1 - \log_2 q_1$ clean ancilla qubits [175], which allows us to use the same ancilla qubits for both the comparator and the incrementer.

In Fig. 6.7 we show a comparison of the depth and number of ancilla qubits required to implement both version of the oracle. We have to consider also that the fewer the number of target states M, the harder is to find them. Specifically the number of required repetitions is expected to scale as $\mathcal{O}(N/\sqrt{M})$ (cf. Sec 6.3), as N^2 is the number of all the possible pairs of the dataset with N elements.

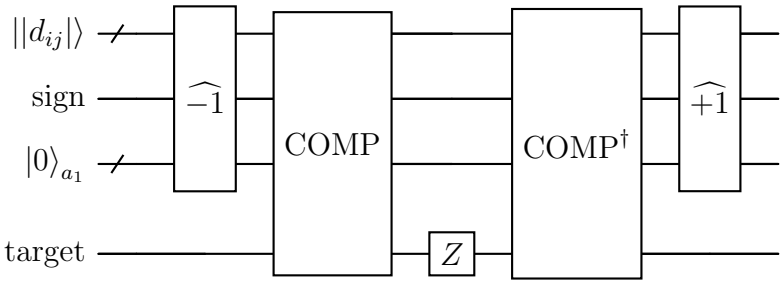


Figure 6.6: Quantum Oracle. The circuit realizes the inverse oracle $-\hat{O}$ defined in Equation (6.30). It is important for the circuit to work correctly that the sign qubit is the most significant qubit. The target register stores the comparison result $d_{ij} \geq h$. The operators $\widehat{\pm 1}$ respectively add and subtract unity to the distance value; they can be implemented with a logarithmic depth quantum adder [176]. Removing the incrementer $\widehat{+1}$ and decrementer $\widehat{-1}$ operators recovers the expression in Equation (6.21). The original oracle $\hat{O}$ can be recovered by replacing the Z gate with $-Z$.

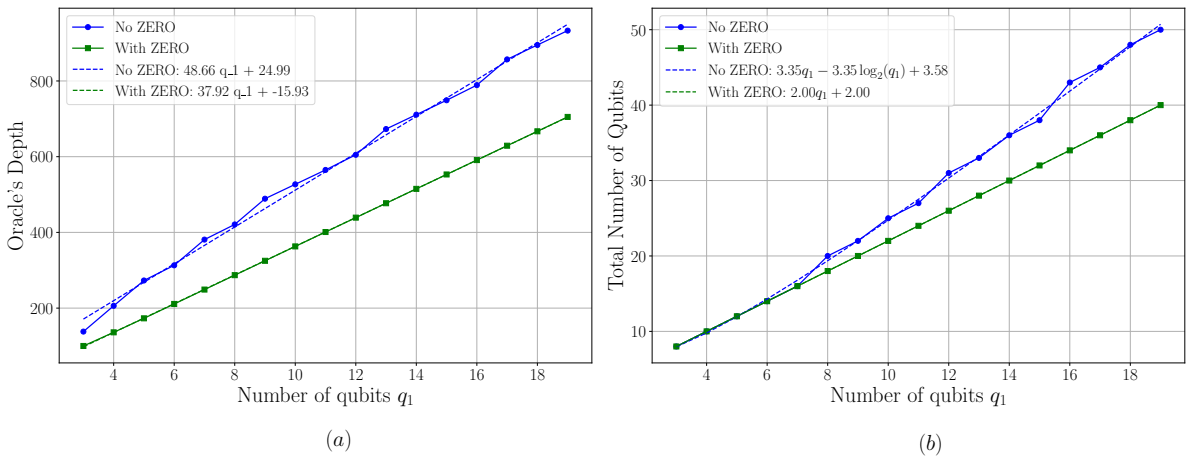


Figure 6.7: Depth (a) and total number of qubits (b) as a function of the number of qubits q_1. The green line refers to the oracle built using the comparator for the case including Zero as in Eq. (6.30)); the blue line to the case without zero as in Eq. (6.21).

6.4.3 The reflection

The final step of the QFRANS algorithm requires performing an inversion around the initial state $|\psi_1\rangle$. As demonstrated in Eq. (6.23) and illustrated in Figs. 6.4 and 6.8, this can be achieved through the implementation of a single operator $\bar{Z}$.
To maintain consistency with our circuit optimization strategy introduced in Section 6.4.2, where we implemented the negative oracle $-\hat{O}$ rather than the positive version, we adopt the same approach here by constructing $-\hat{R}_1$ instead of $\hat{R}_1$. This design choice ensures that the accumulated negative phases from both the oracle and reflection operators cancel out, ultimately producing results that are identical to the standard implementation presented in Section 6.4. Under this modified framework, the required operator becomes $-\bar{Z} = \mathbb{I} - 2\,|0\rangle\langle 0|$, which can be efficiently realized as a multi-controlled Z gate that activates when all control qubits are in the $|0\rangle$ state.

Figure 6.8 presents an explicit representation of the R_0 implementation. As established in Section 6.4.1 and illustrated in Figure 6.5, PREP can be factorized as the product of label creation operator $\hat{L}$ and element creation operator $\hat{E}$. Consequently, following application of $\hat{\bar{Z}}$, the complete $\hat{U}$ operator need not be applied in all cases. When readout measurements target only the labels of dataset elements, applying $\hat{L}$ suffices to create the particle labels, whereas $\hat{U}$ would reconstruct a superposition of particle elements and distances. However, when the ancilla measurement yields $|1\rangle$, correct algorithmic behavior requires the full $\hat{U}$ operator. This is achieved by applying the remaining components $\hat{E}$ and $\hat{D}$ to the appropriate registers.

The scaling of $\hat{R}_0$ is dominated by the PREP operator. In fact, [180], [181] shows that is possible to implement a multi-controlled X gate using only $12n + \mathcal{O}(1)$ qubits and one dirty ancilla, where in the QFRANS scenario $n = 2q_0$. Consequentially, even with an advantageous decomposition of the MCZ, the implementation of $\hat{R}_0$ remains the bottleneck of the algorithm, as the depth and complexity scales in the same manner as PREP.

6.4.4 The readout measurement

The protocols described in Sec. 6.4.2 and Sec. 6.4.3 are repeated until the ancilla qubit is measured in the state $|0\rangle_{a_0}$. When this occurs, assuming error-free quantum hardware, the system has collapsed into the target subspace corresponding to a valid solution. Repeating the procedure allows one to sample multiple solutions.

We treat the total number of solution states M as an unknown random variable and estimate it using Bayesian inference. Under the assumption $N \gg M$, we adopt a Poisson prior distribution

$$p(M) = \frac{\mu^M e^{-\mu_0}}{M!}, \tag{6.31}$$

where $\mu_0 = \sum_M M p(M)$ denotes the prior mean number of solutions. The initial value, μ_0 can be inferred from previous simulations or a priori physical considerations. In the absence of such information, we set μ_0 to the expected number of neighboring pairs in a uniform particle distribution, $\left(\frac{2h}{L}\right)^d N^2$.

For a given run j, let m_j denote the number of oracle queries required before the ancilla is first measured in the state $|0\rangle_{a_0}$. For a problem with M solutions, the probability

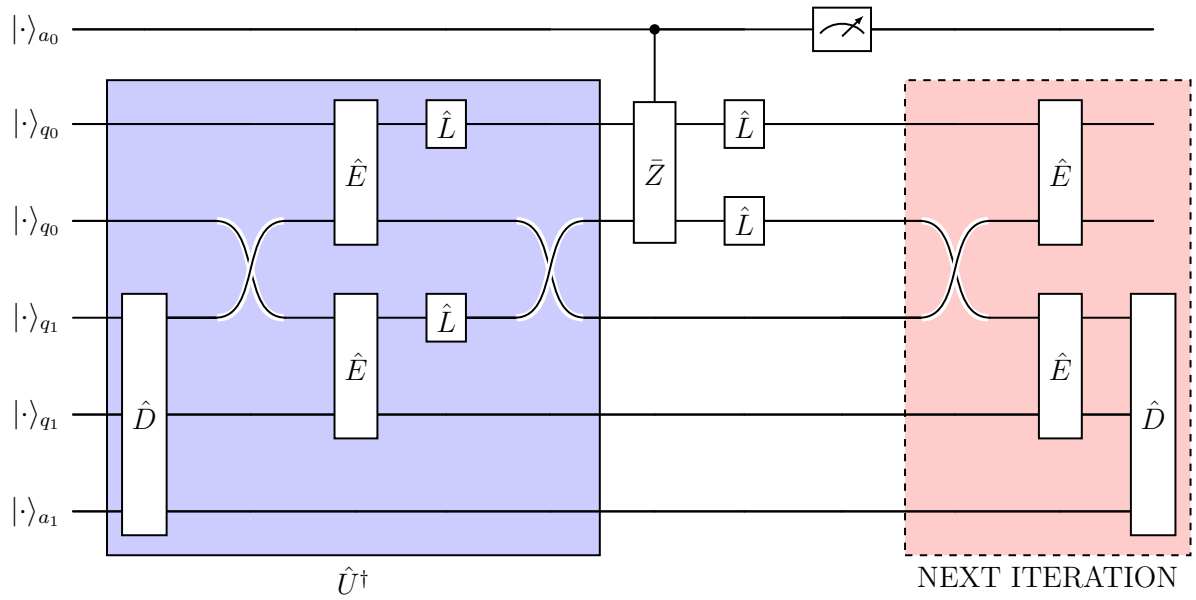


Figure 6.8: Explicit representation of the $\hat{R}_0$ operator. The part included by the red-dashed line is applied only if the ancilla measurement is $|1\rangle$.

that the algorithm succeeds within m_j oracle queries is given by the cumulative success probability

$$P_{\text{cum}}(m_j \mid M) = 1 - \prod_{i=1}^{m_j} \left(1 - p_i(M)\right), \tag{6.32}$$

where $p_i(M)$ is given in Eq. (6.12) denotes the instantaneous success probability at the i-th iteration. The dependence on M enters through the Grover angle $\theta = \arcsin\sqrt{M/N}$.

In addition to the time-to-success information, each run provides information on whether the observed solution is new or has been previously sampled. Let K_{j-1} be the number of distinct solutions identified prior to run j. Assuming uniform sampling over the M solutions, the likelihood associated with run j is

$$\mathcal{L}_j(M) = P_{\text{cum}}(m_j \mid M) \times \begin{cases} \left(\dfrac{M - K_{j-1}}{M}\right)^{\gamma}, & \text{if new}, \\ \left(\dfrac{K_{j-1}}{M}\right)^{\gamma}, & \text{if known}; \end{cases} \tag{6.33}$$

where the exponent $\gamma > 1$ is included to speed up prior convergence to the right value of M, giving more weight to the information about the novelty of the measured outcome.

Following Bayes' rule, the posterior distribution after j runs is updated sequentially as

$$p(M \mid D_j) = \frac{\mathcal{L}_j(M)\, p(M \mid D_{j-1})}{\sum_{M'} \mathcal{L}_j(M')\, p(M' \mid D_{j-1})}, \tag{6.34}$$

where D_j denotes the data collected up to run j. We remark that all the information we can gathered from previous runs is contained in the posterior, containing both data form the convergence behavior of the algorithm and from repeated sampling of known solutions.

To determine when the QFRANS sampling procedure should terminate, we adopt a Bayesian stopping criterion based on the posterior probability that undiscovered solutions still exist. We expect the posterior distribution $P(M|D_j)$ to converge towards a steep distribution peaked on the true value of M; this information, combined with the the number of distinct solutions observed after j runs K_j, should give us the certainty to have measured all the possible outcomes. The algorithm stops when

$$P(M > K_j \mid D_j) = \sum_{M=K_j+1}^{N} p(M \mid D_j) < \varepsilon, \tag{6.35}$$

here ε is a user-defined confidence threshold. The value of ε should be chosen in accordance with the selected convergence speed factor γ: faster convergence requires a more stringent condition (i.e, smaller ε) to ensure that all admissible solutions are captured. It should be noted that applying a power factor $\gamma > 1$ to the likelihood effectively sharpens the posterior, so it can no longer be interpreted as a strictly calibrated probability. Consequently, ε does not directly represent the true probability of unobserved solutions; rather, it serves as a confidence threshold indicating when the posterior is sufficiently concentrated around the most likely values of M. This criterion is similar to requiring that the posterior probability mass is sufficiently concentrated around the true value of M within a prescribed confidence level (i.e, more than $P(M_{true}|D_j) > 1 - \varepsilon$).

A final remark concerns the maximum number of queries allowed. In principle, one could measure indefinitely; however, when implementing the algorithm, it is advisable to set an upper limit on the number of queries. The algorithm is expected to converge in $\mathcal{O}\left(\sqrt{N^2/M}\right)$ queries. In practice, however, M is not known a priori, and only an estimate or guess μ_j is available at the j-th iteration. To provide a more standardized and broadly applicable procedure, we therefore propose to estimate the maximum number of allowed queries according to

$$N_{\text{max queries}}(j) = \left\lceil 1.5 \sqrt{\frac{N^2}{\mu_j}} \right\rceil . \tag{6.36}$$

6.5 Numerical Tests

We aim to verify that the stopping criterion implemented in QFRANS is robust and allows us to recover all solutions, regardless of the scenario or the chosen parameters, and in a reasonable amount of iterations. To this end, we considered a 1 dimensional setup, where 6 particles are arranged in a box of dimension $L = 8$: $\boldsymbol{x} = \{0; 1; 3; 4; 6; 7\}$. We first choose an integer smoothing length $h = 1$ such that $M = 3$ with neighboring result couple being $(x_0, x_1); (x_2, x_3); (x_4, x_5)$.

In this setup, we aim to verify that convergence can be reliably achieved regardless of the initial estimate of the number of solutions, μ_0. To do this, we select a (γ, ε) configuration that ensures convergence within a reasonable number of steps and run 10 simulations to assess the impact of different choices of μ_0. We then test the same configuration on a modified problem, setting $h = 2$ (corresponding to $M = 5$), to confirm that the results remain consistent.

In Tab 6.1 are show the results obtained with $(\gamma, \varepsilon) = (3, 4 \times 10^{-3})$. This choice of parameters favors convergence speed over the guarantee of recovering all solutions in every

Table 6.1: Mean number of found solutions and iterations for the cases with $M = 3$ and $M = 5$ with different initial guess μ_0. We used a convergence speed and tolerance $(\gamma, \varepsilon) = (3, 0.004)$. The means and errors are evaluated on a sample of 10 data.

	$M = 3$		$M = 5$	
μ_0	Solution	Iteration	Solution	Iteration
1	2.80 ± 0.31	10.50 ± 0.79	4.40 ± 0.92	16.00 ± 2.92
2	2.90 ± 0.23	11.50 ± 0.52	4.80 ± 0.46	18.20 ± 1.84
8	2.90 ± 0.23	14.40 ± 0.51	5.00 ± 0.00	22.40 ± 0.56
16	3.00 ± 0.00	16.20 ± 1.08	4.80 ± 0.46	23.40 ± 2.01

case. With these parameters QFRANS finds all the solutions in most of the cases within a restricted number of iterations. Although the test sample is too small for statistically robust conclusions, we find that, regardless of the initial estimate μ_0, this (γ, ε) configuration recovers all solutions in nine out of ten cases, missing only one. Absolute certainty (100%) is impossible, as it would require $\varepsilon = 0$. However, an appropriate choice of ε can substantially increase the likelihood of recovering all solutions, at the cost of more iterations.

An example of this behavior is reported in Tab. 6.2, for $(\gamma, \varepsilon) = (2, 10^{-3})$. In this case, all solutions are recovered more consistently, at the cost of a higher average number of iterations compared to Tab. 6.1. Even with this different parameter choice, convergence depends solely on the (γ, ε) configuration.

At the same time, both case studies show that the initial choice of μ_0 determines the maximum number of iterations required. This behavior originates from the stopping criterion itself. In particular, the condition expressed in Eq. (6.35) is biased toward initial priors that underestimate the total number of solutions, $\mu_0 < M$. Since only probabilities with $M > K$ are included in the calculation, the integration is effectively restricted to the right tail of the probability distributions. As a consequence, posteriors that approach the solution from above (i.e., $\mu_0 > M$) are penalized, as illustrated in Figs. 6.9 and 6.10. In addition, a large overestimation of μ reduces the number of available queries, thereby increasing the likelihood of failure. Despite these effects, both configurations converge within a reasonable number of iterations when the parameters are kept fixed.

Finally, a direct comparison of the posterior evolution in Figs. 6.9 and 6.10 highlights the impact of a more aggressive choice of γ. This effect is particularly evident for $\mu_0 = 16$, where a more conservative choice leads to a significantly slower convergence toward the target state. In summary, this study shows that the (γ, ε) configuration is the decisive factor in recovering the solution, largely independent of the true number of solutions in the problem.

6.6 Analysis of noise resilience

Today's quantum hardware is affected by noise and implementing any search algorithm in a scalable system is a significant challenge [182], [183]. Only recently, a novel noise

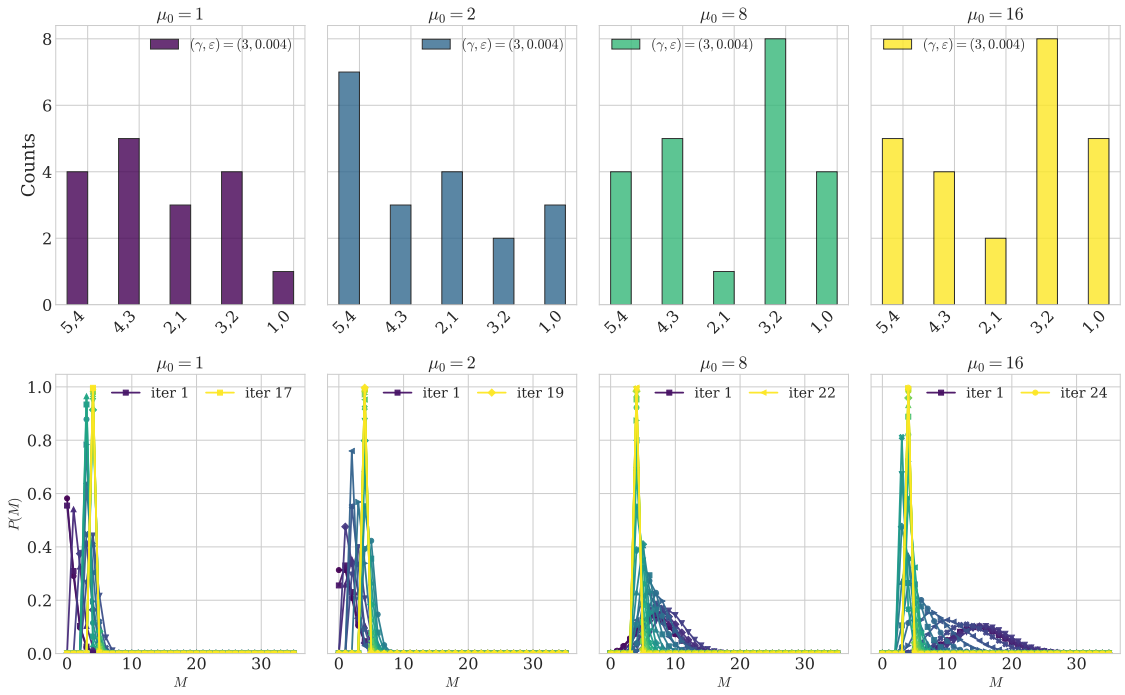


Figure 6.9: Outcomes and posterior probabilities from a simulation with $N = 36, M = 5$ and $(\gamma, \varepsilon) = (3, 0.004)$. The results are presented for different initial mean guesses μ_0 in the prior distribution of Eq. (6.31). In the top row is shown the outcome of the readout, listing name of the particles on the x axis and their respective occurrences. In the bottom row the evolution of the posterior probability in Eq. (6.34) as s function of M; the function peaks around $M = 5$.

Table 6.2: Mean number of found solutions and iterations for the cases with $M = 3$ and $M = 5$ with different initial guess μ_0. We used a convergence speed and tolerance $(\gamma, \varepsilon) = (2, 0.001)$. The means and errors are evaluated on a sample of 10 data.

	$M = 3$		$M = 5$	
μ_0	Solution	Iteration	Solution	Iteration
1	3.00 ± 0.00	15.20 ± 0.31	5.00 ± 0.00	25.00 ± 0.00
2	2.80 ± 0.46	16.00 ± 2.31	4.90 ± 0.23	26.50 ± 1.15
8	3.00 ± 0.00	20.80 ± 0.31	5.00 ± 0.00	31.90 ± 0.41
16	3.00 ± 0.00	22.90 ± 0.23	5.00 ± 0.00	35.50 ± 0.38

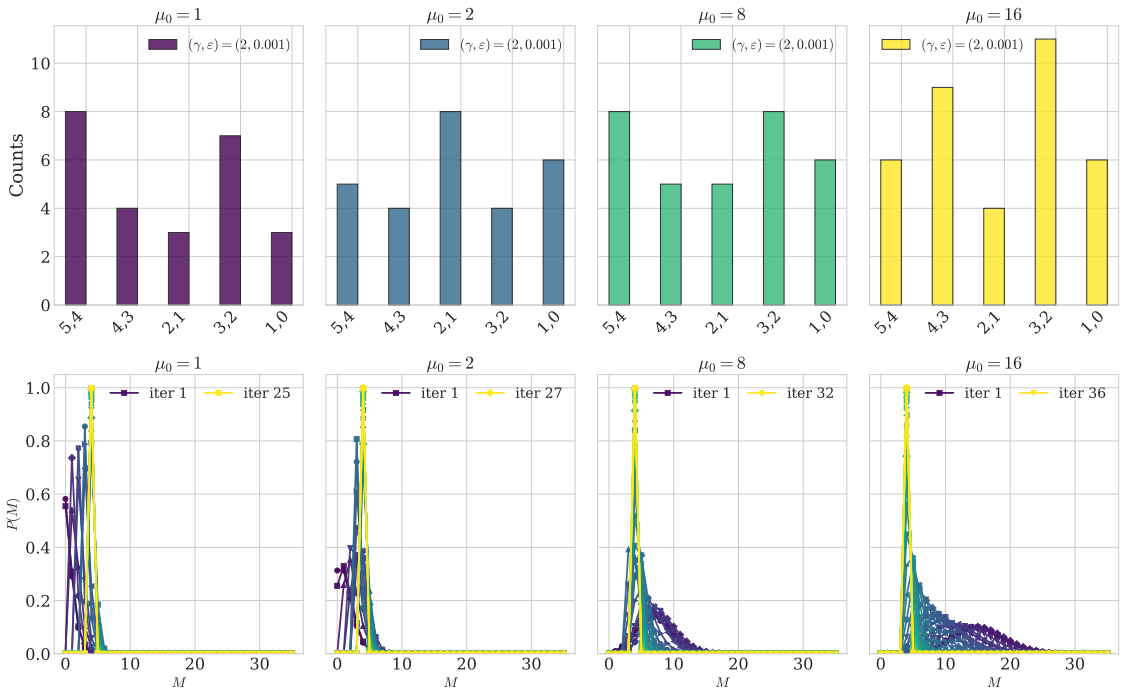


Figure 6.10: Outcomes and posterior probabilities from a simulation with $N = 36, M = 5$ and $(\gamma, \varepsilon) = (2, 10^{-3})$. The results are presented for different initial mean guesses μ_0 in the prior distribution of Eq. (6.31). In the top row is shown the outcome of the readout, listing name of the particles on the x axis and their respective occurrences; all the target states are found. In the bottom row the evolution of the posterior probability in Eq. (6.34) as a function of M; the function peaks around $M = 5$.

tolerant method has been developed to reduce the error threshold for Grover's search by optimizing the number of iterations [184].

In this section we employ a simple model in order to analyze the effect of the noise on the result of the QFRANS algorithm assuming to have a quantum computer that can be characterized only by a bit-flip readout error. This assumption, though quite restrictive, can be used to characterize the noise resilience of the proposed algorithm. Therefore, we consider that the state, after successful measurement of the ancilla qubit onto $|1\rangle$, has collapsed into the correct target state $|\psi_T\rangle$ and that noise affects the algorithm only in the readout measurement part. We will see that this assumption does not change our proposed strategy which does not rely on error correction codes.

The bit-flip channel noise flips the state of a qubit with a probability $1-p =$ error rate, where p is the probability of getting the correct result. This error can affect the two registers that are actually measured, both with q_0 qubits, which give the labels corresponding to the two particles. Starting from the states $|i\rangle_{q_0}, |j\rangle_{q_0}$, the overall probability of successfully measuring the exact indexes i and j is given by p^{2q_0}. As a result, if we set a tolerance value $\texttt{TOL}$ for this probability, i.e. $p^{2q_0} \geq \texttt{TOL}$, the error rate must satisfy the following inequality:

$$\text{error rate} \leq 1 - \texttt{TOL}^{\frac{1}{2q_0}}. \tag{6.37}$$

Using reasonable values for the tolerance $\texttt{TOL} = 0.99$ and the number of qubits $q_0 = 30$ to encode a large database of $N = 2^{30}$ particles, the required error rate is in the order of 10^{-4}. Within this simplified model, the proposed algorithm provides sufficiently accurate results within currently available error rates. To improve the robustness of the proposed algorithm to noise, post-readout classical measurement techniques can be employed. For instance, if the full reflection operator is applied at the last iteration, measuring the state of the distance register q_1 provides a direct method for error detection: an error has occurred if the measured distance exceeds the integer smoothing length h. Moreover, a mismatch between the position register q_1 and the label of the first particle can also reveal erroneous circuit behavior, indicating that the readout should be rejected.

6.7 Discussion

One of the main bottleneck in N-body methods is the FRANS subroutine. This is due to cache misses that arise when data that are close in space resides in distant memory locations. In this chapter we looked for a possible solution using quantum computers.

Building on fixed-point amplitude amplification we developed QFRANS, a quantum algorithm that finds all the elements whose distance is less than a fixed radius ξ with the use of an ancilla qubit that signals if the quantum state has reached the target. The combination of the search algorithm and effective stoppage criterion based on Bayesian interference allows for a sure recovery of all the target state once the learning rate γ and the tolerance ε are chosen. We developed an explicit encoding of the data that, by leveraging quantum superposition, simultaneously evaluates all the N^2 distances between particles. This allows to recover with $\mathcal{O}(M \log M)$ measures and a single quantum circuit all the pairs of close neighbors, as opposed to classical FRANS algorithms that repeat the search process for each particle with the risk of incurring in cache miss.

One of the novelty of this work resides in the oracle, which is efficiently implemented

on a quantum circuit by using two comparators and with a depth scaling linearly with the number of qubits. We proposed two different versions of such oracle, that can be used in different scenarios. We introduced the possibility of using the Chebyshev distance in place of the Euclidean one, bringing the complexity of the oracle to $\mathcal{O}(q_1)$ where 2^{q_1} is the number of grid points used to discretize one dimension. Another central feature of QFRANS is its stopping criterion based on Bayesian interference. By introducing the learning rate γ and the confidence threshold ε, the Bayesian posterior can be made to concentrate around the true number of solutions. With a suitable choice of these parameters, the algorithm consistently recovers all—or nearly all—admissible solutions with high confidence, though at the cost of slower convergence. This approach provides a practical and theoretically grounded way to balance speed and reliability, ensuring that the target states are identified within the desired confidence level. Our algorithm is robust against bit-flip errors in the readout measurements process thanks to the encoding of data labels and distances that allows an easy-to-implement error detection. We proposed and tested QFRANS on a $1D$ grid with a total of 36 possible outcomes. In our tests, we found that convergence depends solely on the choice of (γ, ε) and not on the total number of solutions. In particular, using $(\gamma, \varepsilon) = (2, 10^{-3})$, we conducted a total of 80 trials and recovered all solutions in all but two cases, which can therefore be considered outliers.

Even though we employed state of the art algorithms, state preparation represents the major contribution to the circuit depth of our algorithm. For unstructured datasets, current methods face a fundamental depth-width trade-off: implementations requiring $\mathcal{O}(N)$ circuit depth utilize $\mathcal{O}(\log N)$ ancilla qubits, while those achieving $\mathcal{O}(\log N)$ depth demand $\mathcal{O}(N)$ ancilla qubits, where N is the number of particles. Intermediate strategies exist between these extremes, though all maintain the constraint depth $\times$ width $= \mathcal{O}(N \log N)$.

The fixed-point algorithm necessitates $\sqrt{N^2/M}$ queries to achieve the target state, which, when combined with the cost of implementing the oracle and diffusion operator, yields overall depth $\mathcal{O}(M^{-1/2}N^2)$ with $\mathcal{O}(\log N)$ ancilla qubits, or $\mathcal{O}((M \log M)^{1/2} N \log N)$ depth with $\mathcal{O}(N)$ ancilla qubits, depending on the encoding protocol. In addition we need to count for the readout requiring $\mathcal{O}(M \log M)$ measurements. Although at first glance the computational complexity of QFRANS does not appear to offer substantial advantages over the best classical methods ($\mathcal{O}(N \log N)$), a couple of considerations should be weighed before drawing firm conclusions. The first concerns cache-misses in classical FRANS codes. This issue primarily slows down execution time because the volume of data the CPU must process is too large to be kept in cache and therefore resides in RAM. Those classical codes are then heavily parallelized to reduce the amount of data streamed to the CPU, which shortens run times and lowers overall complexity.
By analogy, the main obstacle to implementing QFRANS is the enormous cost of state preparation. In particular, choosing to work with $\mathcal{O}(N)$ ancilla qubits and depth $\mathcal{O}(\log N)$ is equivalent to having a particles' database with logarithmic access. If the algorithm were parallelized, for example using domain decomposition, each QPU would handle only a fraction of the total N particles rather than the entire set. This approach necessitates the use of multiple QPUs, which can significantly reduce the computational load per device; for instance, deploying 10 QPUs would decrease the per-QPU workload to $N/10$, thereby enhancing scalability and enabling more efficient utilization of quantum resources. Once again, QFRANS does not suffer from cache misses as there is no QRAM [185] and the

burden falls entirely on the QPU. This will be analogous to a classical cache only implementation. Nevertheless, the limiting factor here is still the sheer amount of data, thus for the moment, QFRANS should be regarded as a proof of concept, as well as a cache only classical implementation.

In future research more efficient state preparation algorithms will be investigated. As anticipated, a more detailed study on domain decomposition strategies may offer computational advantages, though at the cost of requiring multiple circuit implementations. Additionally, detailed studies of specific applications—such as cosmological simulations or molecular dynamics may be used to validate the algorithm's practical utility. Understanding error propagation and developing robust mitigation strategies will be considered in future studies. QFRANS represents a promising foundation for quantum algorithms addressing neighbor search problems. This work suggests that quantum computing may offer genuine advantages in computational domains where classical methods face fundamental memory hierarchy limitations rather than purely algorithmic constraints.

In this Thesis, QFRANS is investigated primarily through quantum emulation, reflecting the current limitations due to the limitation of both state preparation and quantum hardware. The long-term goal, however, is to integrate quantum processors as accelerators within hybrid high-performance computing systems. In such architectures, quantum devices are not intended as standalone replacements for classical computation, but rather as specialized coprocessors aimed at alleviating performance bottlenecks related to memory hierarchy or data movement.

Conclusions

In this Thesis, we have examined the potential of quantum computing in cosmological simulations, with the primary aim of assessing whether quantum algorithms can contribute to modeling dark matter (DM) evolution. To this end, we focused on two main avenues: developing quantum methods for dark matter dynamics and using quantum computing to accelerate the most computationally demanding aspects of traditional simulation frameworks.

Schrödinger-Poisson and Vlasov-Poisson Simulations

To address the first objective, we built on recent work [53], which suggested that the evolution of DM could be described using the Schrödinger equation. In the limit of a small quantum diffusion scale, $\hbar/m \to 0$, the Schrödinger-Poisson (SP) equation should reproduce the dynamics of the Vlasov-Poisson (VP) system. This formulation makes the SP system a natural candidate for quantum simulation. We investigated this correspondence in the context of Cold Dark Matter (CDM) by comparing three approaches to solving the VP equation: two methods based on direct VP integration (these two are part of the same approach), a Particle-Mesh (PM) solver, and a spectral solver for the SP system.

In Section 3.2 We found that existing VP solvers are not well suited to CDM-like initial conditions, where the velocity distribution is discontinuous and effectively described by a Dirac delta function. This discontinuity leads to numerical difficulties in phase-space-based methods. To overcome this issue, we employed a spectral approach in which a very small velocity dispersion is introduced, transforming the delta-like distribution into a narrow Gaussian. For example, on a line divided in 2048 grid points, the narrow Gaussian has a width of approximately 13 grid points. This modification preserves the correct physical evolution and reproduces the right density profile apart from minor numerical oscillations in regions near zero valued-points. These results are published in [22].

Using this improved VP solver, we verified in Section 3.3 that the SP formulation accurately reproduces both VP and PM results in the case of a one-dimensional sinusoidal perturbation and a two-dimensional Gaussian collapse. Reaching this agreement requires higher spatial resolution in SP simulations, followed by a coarse-graining to the scale at which the comparison with VP has to be done. This procedure allows the quantum diffusion scale $\hbar/m$ to be chosen sufficiently small, ensuring accurate physical evolution. In practice, the one-dimensional spectral SP simulations required resolutions of up to 2^{15} grid points before coarse-graining to a resolution of 2^{10} points, with a diffusion scale $\hbar/m = 2.5 \times 10^{-5}$.

Building on these results, we also confirmed that, for the simple test cases considered

here, the PM method provides reliable results for the simple CDM-like problems considered Section 3.3. In such tests, the phase-space structure is straightforward, and the VP solver is not strictly necessary; in fact, VP performs less effectively under these simple initial conditions, which do not fully exploit its capabilities. Consequently, PM was treated as the reference solution in these tests. However, it is important to emphasize that these conclusions are limited to the simple scenarios studied here. More complex phase-space configurations, such as those encountered in full cosmological simulations, could reveal advantages of the VP approach that are not captured in these preliminary tests.

Quantum Algorithms for Schrödinger-Poisson

Having verified the conditions of application of the SP description for the evolution of a self-gravitating collisionless fluid, we investigated the development of quantum algorithms to solve the SP system. In Chapter 5, we proposed a variational algorithm [19] designed for near-term quantum (NISQ) devices [20].

In the SP equation, the potential depends on the wavefunction itself, introducing a non-linear aspect that cannot be directly handled by quantum computing, which operates linearly. There are various ways to address this issue, but in this work, we adopted a simple approach: each timestep was divided into two operations, first solving the Poisson equation to determine the potential and then evolving the wavefunction in time. This strategy allowed us to implement the SP dynamics within the constraints of current quantum algorithms. The gravitational potential was obtained using a standard variational method from the literature, which minimizes the distance between the trial state and the exact solution of the Poisson equation. The main innovation of the approach introduced in this Thesis lies in the time evolution. Using a scheme derived from McLachlan's variational principle [186], the problem is reduced to solving the linear system $\mathrm{M}\dot{\theta} = \mathrm{B}(V)$, with

$$\mathrm{M}_{ij} = \mathrm{Re}\left\{\left\langle \partial_{\theta_i}\Psi \mid \partial_{\theta_j}\Psi\right\rangle - \left\langle \partial_{\theta_i}\Psi \mid \Psi\right\rangle \left\langle \Psi \mid \partial_{\theta_j}\Psi\right\rangle\right\} \tag{6.38}$$

$$\mathrm{B}_i = \mathrm{Im}\left\{\langle \partial_{\theta_i}\Psi|\mathcal{H}|\Psi\rangle - \langle \partial_{\theta_i}\Psi \mid \Psi\rangle \langle \Psi|\mathcal{H}|\Psi\rangle\right\} \tag{6.39}$$

where each element of M and $\mathrm{B}(V)$ is evaluated via quantum circuits presented in Sec. 4.2. In this framework, we presented an original circuit design for computing the potential term in $\langle \partial_{\theta_i}\Psi| \mathcal{H} |\Psi\rangle$, $\mathcal{H}$ being the Hamiltonian of the system. For a single timestep, the algorithm requires a number of circuits that scales quadratically with the number of variational parameters (Tab. 4.2). Using n qubits to represent 2^n grid points and a number of parameters $M_p < N$, we found that circuits with a depth scaling linearly with n can achieve a fidelity of approximately 0.96; results are reported in Fig. 4.4 and Tab. 4.1. At the same time, the number of timesteps needed to reach this fidelity increases as the number of variational parameters decreases. Furthermore, classical simulations revealed an empirical logarithmic scaling law for the number of qubits required as the system approaches the classical limit ($\hbar/m \to 0$), highlighting the potential efficiency of this approach in larger-scale or more classical regimes. These results are reported in Fig. 6.3a The results of this work have been published in [25].

In Section 5.3 we explored several techniques to regularize the matrix $\mathcal{M}$, but none proved fully successful. The challenge arises from the need to separate numerical noise

from meaningful information, while the system itself continuously evolves over time. As a result, we highlighted the challenge of developing a strategy that would allow the matrix to be regularized in a manner independent of the specific resolution or initial conditions. In addition, the number of measurements required to estimate $B(V)$ scales as $N\mathcal{O}(1/\varepsilon^2)$, where ε is the desired numerical precision and N the number of grid points.

In Section 5.4 we also explored an alternative strategy based on linearizing the SP equations using Carleman embedding [187]. We applied a third-order Carleman truncation to linearize the SP system, which resulted in a system of 27 linear equations. Although this approach allowed us to formulate the problem in a linear framework, the accuracy achieved did not suffice to describe the evolution of the SP equations, as shown in Fig. 5.11. Extending the truncation to higher orders would require an exponentially larger number of terms, making the method impractical with current techniques. These results highlight the intrinsic difficulty of directly simulating the SP system on a quantum device. The results of this work are currently under revision for publication.

Quantum Acceleration of Classical Codes

Given the limitations encountered in direct quantum simulation, In Chapter 6 we explored the use of QC as an accelerator for classical cosmological codes. In large-scale cosmological simulations, neighbor search routines represent a major computational bottleneck, as they are central to both gravitational force calculations and Smoothed Particle Hydrodynamics (SPH). This motivated the development of QFRANS, a quantum algorithm for the Fixed-Radius Near Neighbor Search (FRANS).

QFRANS identifies all elements within a fixed search radius in a dataset of size N using amplitude amplification. The algorithm presented in Sec. 6.4 requires a total of $\mathcal{O}(N\sqrt{M})$ queries to find all M neighboring pairs. The depth of a single query scales as $\mathcal{O}(\log N)$ and requires N ancilla qubits. Each query consists of two components: the oracle and a diffusion step. The overall scaling is dominated by the diffusion step, which must apply the same unitary as used in the state preparation. In contrast, the oracle itself exhibits favorable scaling, $\mathcal{O}(q_1)$, where 2^{q_1} corresponds to the number of grid points. In realistic cosmological simulations, the number of particles can reach $N \sim 10^{10}$, which implies an equally large number of qubits. Although this requirement can be partially mitigated by spatial partitioning, as is commonly done in classical codes, it remains far beyond the capabilities of current and near-term quantum hardware. Nevertheless, QFRANS provides a valuable proof of principle and highlights how quantum acceleration could target specific computational bottlenecks in classical simulations.

In summary, we have identified critical constraints that guide future research on both direct quantum simulation of SP and quantum acceleration. In particular, our analysis shows that direct simulation remains highly challenging due to the intrinsic difficulty of encoding strong nonlinearities within a fundamentally linear quantum framework.

In the variational approach we employed, the main limitation is scaling, primarily related to the regularization of the matrix for time evolution. Similarly, Carleman methods based on linearization encounter fundamental obstacles even before implementing the actual quantum circuit, producing systems of linear equations of exponential size. From our analysis, we conclude that even with alternative variational algorithms, to our knowl-

edge, the number of timesteps remains constrained by the classical numerical integration scheme chosen, mitigating any potential advantage gained in solving a single timestep. This is an inherent feature of variational algorithms that do not offload the time evolution. However, even if a linearized system could be produced, its size would be enormous, and the time integration of such a system would impose a prohibitive cost in both circuit depth and number of qubits.

Although the circuit presented here for solving FRANS has practical limitations, particularly due to the resources required for preparing the initial state, we consider that further investigations in this area are warranted, as the analysis was not as in-depth as for direct simulation. Moreover, employing quantum computing as an accelerator appears more practical, especially as cosmological simulations increasingly face bottlenecks related to data access and memory movement rather than raw floating-point operations. In this context, quantum acceleration could offer significant advantages.

Future Perspectives

Building on the insights gained in this Thesis, the most promising direction for future research is the strategic use of quantum computing to accelerate the most computationally demanding subroutines in classical cosmological simulations. This approach leverages the strengths of quantum techniques without attempting a full quantum simulation of the SP system, which remains highly challenging due to the embedding of strong nonlinearities and the scaling limitations of variational and linearization methods.

Concrete targets for quantum acceleration include neighbor search algorithms and long-range interaction computations. Quantum algorithms capable of efficiently identifying the M nearest neighbors would directly support SPH routines, while quantum implementations of fast-multipole methods could accelerate long-range gravitational interactions. Focusing on these computationally intensive subroutines allows quantum computing to provide practical benefits within classical simulation frameworks.

Direct quantum simulation of the SP system remains highly challenging. As this Thesis did not exhaust all possible approaches, some speculative directions could be explored in future work. For example, hybrid strategies such as continuous-variables or open quantum system techniques could be used to simulate limited portions of SP evolution.

Overall, this Thesis demonstrates that while fully quantum simulations of cosmological dynamics remain beyond current reach, strategically applying quantum algorithms to accelerate key subroutines offers a promising path forward. Continued exploration of quantum acceleration has the potential to significantly enhance the efficiency and scalability of large-scale cosmological simulations, opening new avenues for high-precision modeling.

Appendix A

Supplement material for the Schrödinger-Poisson quantum algorithm

We start by showing in Sec. A.1 how to obtain the form of the Shrödinger-Poisson (SP) equation used in the main text. In Sec. A.2 we show the explicit form of the cost function used in the minimization process used for the solution of the Poisson equation and the circuit necessary to build it. Sec.A.3 gives more detail about the adder circuit proposed in [23]. Finally, in Sec. A.4, is given a proof of concept on how the circuit used in the main text works.

A.1 Dimension Re-Scaling of Schrödinger-Poisson equation

Let us consider the Schrödinger-Poisson (SP) equation in a format similar to the one presented in [53]:

$$\begin{aligned} i\hbar\frac{\partial\Psi}{\partial t} &= -\frac{\hbar^2}{2m}\nabla^2\Psi + mU\Psi\,; \\ \nabla^2 U &= 4\pi G(\rho\ -\bar{\rho})\,, \end{aligned} \tag{A.1}$$

where $\bar{\rho}$ is a reference density. If we chose it to be the average density over the volume, the normalization of the wavefunction as given in the main text directly follows.

We can transform the potential so that

$$\nabla^2 U = 4\pi G\bar{\rho}\left(\frac{\rho}{\bar{\rho}} - 1\right)\,. \tag{A.2}$$

If we now define $|\Psi|^2 := \rho/\bar{\rho}$ and $\lambda := \hbar/m$, Eqs.(A.1) can be recast in the form

$$\begin{aligned} i\frac{\partial\Psi}{\partial t} &= -\frac{\lambda}{2}\nabla^2\Psi + \frac{1}{\lambda}U\Psi\,; \\ \nabla^2 U &= 4\pi G\bar{\rho}(|\Psi|^2 - 1)\,. \end{aligned} \tag{A.3}$$

The potential we used in the simulation is redefined so that the Poisson equation is adimensional. This is done defining a function $V := U/\alpha$, so that

$$\nabla^2 V = |\Psi|^2 - 1\,, \tag{A.4}$$

with $\alpha = 4\pi G\bar{\rho}$. We have now to substitute $U(V)$ in Eq. (A.3):

$$\begin{aligned} i\frac{\partial\Psi}{\partial t} &= -\frac{\lambda}{2}\nabla^2\Psi + \frac{\alpha}{\lambda}V\Psi\,; \\ \nabla^2 V &= |\Psi|^2 - 1. \end{aligned} \tag{A.5}$$

Finally the equation we worked with is obtained if and only if

$$\alpha = 1 \iff \bar{\rho} = \frac{1}{4\pi G} \iff \frac{4\pi G}{L}\int_0^L \rho(x)dx = 1\,, \tag{A.6}$$

where we used the definition of $\bar{\rho}$. This sets the computational units as combinations of $[G, \bar{\rho}, L]$.

Is worth noticing that from a dimensional point of view, the first equation in Eq. (A.5) holds only if the density ρ is defined in 3D. If this is not the case, a correcting factor is needed. When we work in one dimension, we assume spherical symmetry, so that the dependence of the functions involved is one dimensional, i.e. there is only dependence on a radial coordinate r, but the density remains three-dimensional (i.e, $[\rho(r)] = 1/L^3$).

A.2 Potential cost function

In the main text, the cost function used to find the solution of the Poisson equation,(see Eq. (4.14) in main text) is a Euclidean norm of a vector

$$||\nabla^2 V(\phi) - |\Psi(\theta)|^2 + 1||^2 = \sum_{j=0}^{N-1}\left(\nabla^2 V_j(\boldsymbol{\phi}) - |\Psi_j(\boldsymbol{\theta})|^2 + 1\right)^2 . \tag{A.7}$$

Developing this relation, grouping all terms and taking into consideration only those explicitly dependent on $\boldsymbol{\phi}$ we found

$$min_{\boldsymbol{\phi}}\Big\{|\nabla^2 V(\boldsymbol{\phi})|^2 - 2\,\mathrm{Re}\{\nabla^2 V(\boldsymbol{\phi})\cdot|\Psi|^2\} + 2\,\mathrm{Re}\{\nabla^2 V(\boldsymbol{\phi})\cdot\mathbf{1}\}\Big\}\,, \tag{A.8}$$

where we used for conciseness $\Psi =: \Psi(\boldsymbol{\theta})$.
Due to the periodic boundary conditions (PBC) of the problem, the last term of the previous equation vanish if we use a finite differences approach for the evaluation of the Laplacian $\nabla^2 V_j = (V_{j+1} - 2V_j + V_{j-1})/\Delta x^2$, where we omitted the parameters dependence of V just for brevity.
Let us focus in the first term of Eq.(A.8). Remembering the normalisation of the potential (Eq. (4.12) in main text) and the PBC one finds that

$$\sum_{j=0}^{N-1}\phi_V^2\left(\frac{\tilde{V}_{j+1} - 2\tilde{V}_j + \tilde{V}_{j-1}}{\Delta x^4}\right)^2\frac{2\phi_V^2}{\Delta x^4}\left[4\left(1 - \sum_{j=0}^{N-1}\tilde{V}_j\tilde{V}_{j+1}\right) - \left(1 - \sum_{j=0}^{N-1}\tilde{V}_j\tilde{V}_{j+2}\right)\right] . \tag{A.9}$$

This term can be computed using a circuit like the one in Fig. 4.3 in the main text, where instead of $F(\boldsymbol{\theta})$ one uses the unitary ansatz for the potential $U(\tilde{\boldsymbol{\phi}})$ (Fig. 4.1 in main

text). This is possible because the potential is real valued, and by evaluating $\langle\langle\sigma_z\rangle\rangle$ on the ancilla qubit

$$1 - \langle\sigma_z\rangle = 1 - \sum_{j=0}^{N-1} V_j V_{j+1}\,. \tag{A.10}$$

If two adder circuits A are used instead, from the previous relation is possible to retrieve $1 - \sum_{j=0}^{N-1} V_j V_{j+2}$.

Switching to the second term in Eq.(A.8) and unraveling the Laplacian we find that three terms need to be evaluated.

$$\sum_{j=0}^{N-1} \phi_V |\Psi_j|^2 \frac{V_{j+1} - 2V_j + V_{j-1}}{\Delta x^2} \cdot \Delta x^2 = \langle\Psi| V_+ |\Psi\rangle - 2\langle\Psi| V |\Psi\rangle + \langle\Psi| V_- |\Psi\rangle\ , \tag{A.11}$$

where $V_\pm$ are shifted version of the potential (analogous to the one in Eq. (4.21) in main text). These three expectation value can be evaluated with a similar circuit to the one in Fig. 4.3 in the main text, where instead of $F(\boldsymbol{\theta})$ one uses the unitary ansatz for the wavefunction $U(\boldsymbol{\theta})$. The shifted potential is obtained with the adder circuit and its inverse.

A.3 Adder circuit

This circuit scheme is taken from [23]. Here we present it with some more detail on how it works.
This unitary action on the j-th basis produces a negative shift $A : |bin(j)\rangle \mapsto |bin(j-1)\rangle$. When this is applied to a generic wavefunction it results in a shift of the coefficient that takes into consideration periodic boundary conditions $A|\psi\rangle = \sum_{j=0}^{N-1} \psi_{j+1} |bin(j)\rangle$. In this work, the action of the unitary A is controlled by an ancillary qubit.

The rationale behind its working principle is that, to produce the desired shift, (*i*) first the least significant (LSQ) qubit have to be negated using one CX gate, (*ii*) then is added to the second LSQ using a Toffoli controlled by the ancilla and the LSQ. (*iii*) Going up in the hierarchy, to the following qubit is added a product of the previous state. This is repeated until the most significant qubit. To store the product of the previous states are used $n-2$ ancillary qubits. The *loading* process is carried out by n-2 Toffoli gates, while the *adding* uses $n-2$ CX gates and Two Toffoli. (*iv*) To finish the procedure, the ancillary register needs to be set back to the initial state $|\mathbf{0}\rangle$ using $n-2$ Toffoli. The implementation of the adder circuit requires a total of $2n-2$ Toffoli, $n-2$ CX gates and $n-2$ ancillary qubits. In Fig. A.1 is shown an example for the case of a 4 qubits system. The unitary A^{-1} produces a positive shift and is obtained reverting the adder (i.e, from right to left).

A.4 Circuit proofs

This section gives an idea on how the circuits proposed in Fig. 4.3 of the main text work.

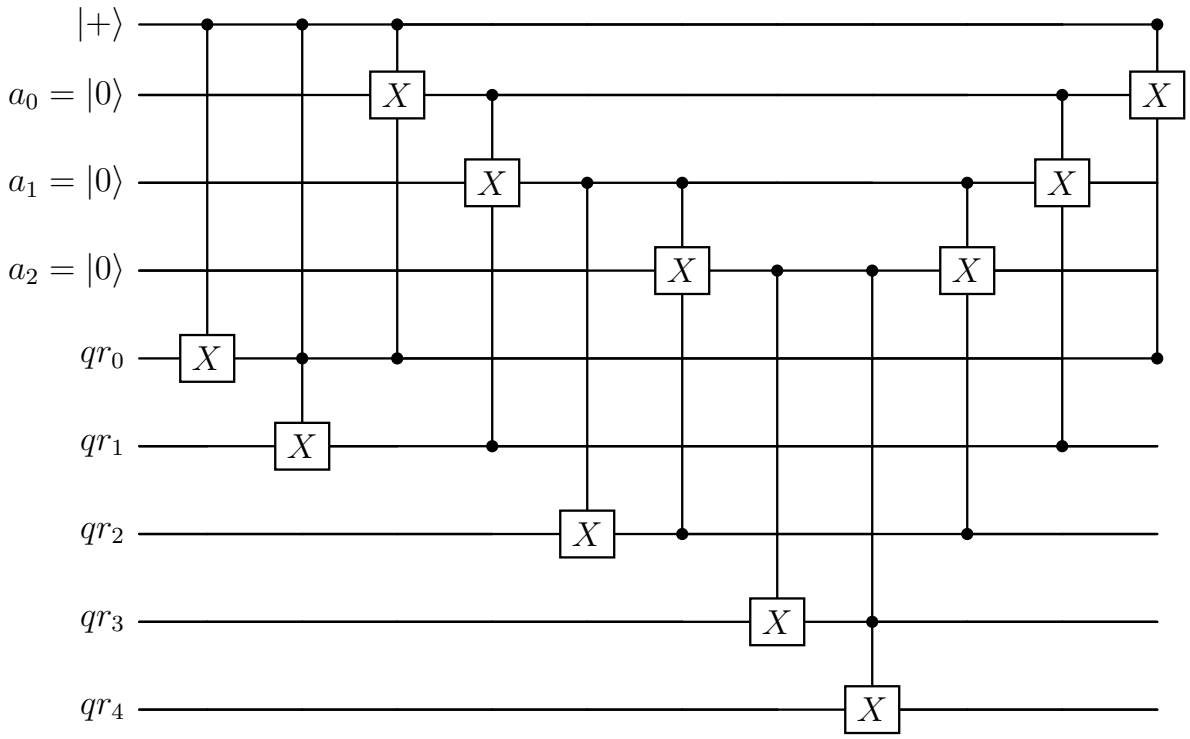


Figure A.1: Example of an adder for a 5-qubits system. The first qubit on the top is the control. The most significant qubit is the one at the bottom.

Potential

Consider the circuit used for the evaluation of $\langle \partial_{\theta_k} \psi | \tilde{V} | \psi \rangle$. We refer to the the quantum state before the application of the Toffoli ladder with

$$|\Xi\rangle_0 = \frac{1}{\sqrt{2}} \left(2i \, |\mathbf{0}\rangle \, |\partial_{\theta_k} \psi\rangle \, |0\rangle + |\Phi_{\tilde{V}}\rangle \, |\psi\rangle \, |1\rangle \right), \tag{A.12}$$

where we used Eq. (4.17) of the main text and the fact that we can encode the wavefunction and the potential using parameterized circuits. If we omit the explicit parameters' dependence we have that the wavefunction can be encoded as

$$|\psi\rangle = \sum_{j=0}^{N-1} \psi_j \, |bin(j)\rangle \ , \tag{A.13}$$

and the potential (see Eq. (4.12) in the main text)

$$|\Phi_{\tilde{V}}\rangle = \sum_{j=0}^{N-1} \tilde{V}_j \, |bin(j)\rangle \ . \tag{A.14}$$

With $bin(j)$ we refer to the binary conversion of the decimal number j. The Toffoli gate adds to the control qubit the product of the two control states.

$$|\Xi\rangle_1 = \frac{1}{\sqrt{2}} \Big(2i \, |\mathbf{0}\rangle \, |\partial_{\theta_k} \psi\rangle \, |0\rangle + \sum_{j,l=0}^{N-1} \psi_l \tilde{V}_{\tilde{j}(j,l)} \, |bin(j)\rangle \, |bin(l)\rangle \, |1\rangle \Big), \tag{A.15}$$

where we defined $\tilde{J}(j,l) = dec(bin(j) + bin(l))$ as the decimal conversion of the binary sum of the indices j and l, with periodic boundary conditions (e,g. with $N = 4$, $\tilde{J}(1,3) = dec(01 + 11) = dec(00) = 0$; with $N = 8$, $\tilde{J}(1,3) = dec(001 + 011) = dec(100) = 4$). We point out that $\tilde{J}(0,l) = l$. With this relation in mind one can write the quantum state before the measurement as

$$|\Xi\rangle_2 = \frac{1}{2}\sum_{l=0}^{N-1}\Big[\Big(2i\partial_k\psi_l \pm \tilde{V}_l\psi_l)\,|\mathbf{0}\rangle\Big) \pm \sum_{j=1}^{N-1}\psi_l\tilde{V}_{\tilde{J}(j,l)}\,|bin(j)\rangle\Big]\,|bin(l)\rangle\,|0/1\rangle\ , \tag{A.16}$$

where the sign + (or −) is used when the ancilla qubit in the state $|0\rangle$ (or $|1\rangle$).
Evaluating σ^z with this quantum state is equivalent to find the probability of having outcome 0 minus the one of outcome 1.

$$P(0/1) = \frac{1}{4}\sum_{l=0}^{N-1}\Big[4|\partial_k\psi_l|^2 + |\psi_l|^2\big(\tilde{V}_l^2 + \sum_{j=1}^{N-1}\tilde{V}^2_{\tilde{J}(j,l)}\big) \mp 2i\left(\partial_k\psi_l^*\tilde{V}_l\psi_l - \partial_k\psi_l\tilde{V}_l\psi_l^*\right)\Big]\,. \tag{A.17}$$

We observe that $\tilde{V}_l^2$ corresponds to $\tilde{V}^2_{\tilde{J}(j=0,l)}$; this way $\tilde{V}_l^2 + \sum_{j=1}^{N-1}\tilde{V}^2_{\tilde{J}(j,l)} = \sum_{j=0}^{N-1}\tilde{V}^2_{\tilde{J}(j,l)}$, where the periodic boundary conditions assure us that, for a given l, this is equivalent to the squared module of the quantum state $|\Phi_{\tilde{V}}\rangle$. Remembering now the normalization of quantum states we can write

$$\langle\sigma^z_V\rangle = P(0) - P(1) = -i\sum_{l=0}^{N-1}(\partial_k\psi_l^*\tilde{V}_l\psi_l - \partial_k\psi_l\tilde{V}_l\psi_l^*) = 2\mathrm{Im}\left\{\sum_{l=0}^{N-1}\partial_k\psi_l^*\tilde{V}_l\psi_l\right\}\,. \tag{A.18}$$

Kinetic term

Consider Eq. (4.21) of the main text. The evaluation of the overlap $\langle\partial_{\theta_k}\psi|\psi|\partial_{\theta_k}\psi|\psi\rangle$ is easy and has already been tackled in the main text as well as in [24]. This subsection analyzes the implementation of the circuits in Fig. 4.3 in the main text.
Let us consider the case in which the derivative is controlled by the ancilla qubit in the state $|0\rangle$ ($F_k^{(0)}$ is used). Since the adder circuit (A.3) is controlled by the ancilla state $|1\rangle$, the quantum state after its application is

$$|\Xi\rangle_0 = \frac{1}{\sqrt{2}}\sum_{j=0}^{N-1}\left(2i\ \partial_k\psi_j\,|bin(j)\rangle\,|0\rangle + \psi_{j+1}\,|bin(j)\rangle\,|1\rangle\right)\,. \tag{A.19}$$

The quantum state on which $\langle\sigma^z_-\rangle$ is evaluated is given by

$$|\Xi\rangle_1 = \frac{1}{2}\sum_{j=0}^{N-1}\left(2i\ \partial_k\psi_j \pm \psi_{j+1}\right)|bin(j)\rangle\,|0/1\rangle\ . \tag{A.20}$$

Now, in a similar manner to what has been done in the case of the potential is easy to find that

$$\langle\sigma^z_-\rangle = P(0) - P(1) = 2\,\mathrm{Im}\left\{\sum_j\partial_k\psi_j^*\psi_{j+1}\right\} \tag{A.21}$$

Following the same procedure, but in the case when the derivative and the adder are controlled by the same ancilla state (e.g, $|1\rangle$ if using $F_k^{(1)}$) one find that the final state is given by

$$|\Xi\rangle_1 = \frac{1}{2}\sum_{j=0}^{N-1}\left(\psi_j \pm 2i\ \partial_k\psi_{j+1}\right)|bin(j)\rangle\,|0/1\rangle\ . \tag{A.22}$$

Since the indices follow periodic boundary conditions, what really matter is the relative shift between ψ and $\partial_k\psi$. We thus find that

$$\langle\sigma_+^z\rangle = P(0) - P(1) = 2\,\mathrm{Im}\left\{\sum_j \partial_k\psi_j^*\psi_{j-1}\right\}\ . \tag{A.23}$$

Appendix B

Supplement material to the Carleman linearization

Here we report the explicit form of the linearized Schrödinger-Poisson equation using Carleman embedding. The results repersent the trunation at second and third order respectiverly

B.1 Second order Carleman-SP system

This is the second order Carleman matrix for the Schrödinger-Poisson problem, is a $(3N + 6N^2) \times (3N + 6N^2)$ sparse matrix (The tensor product has dimension N^2)

$$\partial_t v_1 = -\frac{D}{2}\Delta v_2 + \frac{1}{D}Bv_5 \tag{B.1}$$

$$\partial_t v_2 = \frac{D}{2}\Delta v_1 - \frac{1}{D}Bv_4 \tag{B.2}$$

$$\partial_t v_3 = \gamma\left(\Delta v_3 - \alpha B v_6 - \alpha B v_7 + \alpha\right) \tag{B.3}$$

$$\partial_t v_4 = \gamma\alpha \mathrm{O} v_1 + \gamma\left(\Delta \otimes 1\right) v_4 - \frac{D}{2}(1 \otimes \Delta)v_5 \tag{B.4}$$

$$\partial_t v_5 = \gamma\alpha \mathrm{O} v_2 + \gamma(\Delta \otimes 1)v_5 + \frac{D}{2}(1 \otimes \Delta)v_4 \tag{B.5}$$

$$\partial_t v_6 = -\frac{D}{2}\left[(1 \otimes \Delta)v_8 + (\Delta \otimes 1)v_9\right] \tag{B.6}$$

$$\partial_t v_7 = \frac{D}{2}\left[(\Delta \otimes 1)v_8 + (1 \otimes \Delta)v_9\right] \tag{B.7}$$

$$\partial_t v_8 = \frac{D}{2}\left[(1 \otimes \Delta)v_6 - (\Delta \otimes 1)v_7\right] \tag{B.8}$$

$$\partial_t v_9 = \frac{D}{2}\left[(\Delta \otimes 1)v_6 - (1 \otimes \Delta)v_7\right] \tag{B.9}$$

Notice in fact, that $v_4 = v_3 \otimes v_1$ and $v_5 = v_3 \otimes v_2$.

Moreover, B is an $N \times N^2$ used to obtain the vector N containing the quadratic term and the product of the wave function and the potential from their tensor products, for example $B_{i,j,k} = \delta_{i,j}\delta_{j,k}$, such that acting on $v_5 = v_3 \otimes v_2$ returns the $i-$th component

$$B_{i,j,k}(v_5)_{j,k} = B_{i,j,k}(v_3)_j(v_2)_k = (v_3)_i(v_2)_i\,. \tag{B.10}$$

In the same manner O acts on a vector v with the purpose of obtaining $1 \otimes v$ which is simply a matrix whose rows are the vector itself. This Can be obtained by a rank 3 tensor: a collection of Identity matrices.

B.2 Third order Carleman-SP system

The Carleman vector consist in a combination of 3 N-component vectors, 6 (N, N) tensors and 18 (N, N, N) tensors. We can thus approximate the Carleman matrix to be a $(3N + 6N^2 + 18N^3)^2$ matrix, only for a third order approximation. The explicit terms composing the third-order Carleman matrix are:

$$
\begin{aligned}
\partial_t v_4 &= \gamma\alpha \mathrm{O} v_1 + \gamma\left(\Delta \otimes \mathbb{I}\right) v_4 - \frac{D}{2}(\mathbb{I} \otimes \Delta) v_5 + \frac{1}{D}(\mathbb{I} \otimes B) v_{19} - \gamma(B \otimes \mathbb{I})(v_{26} + v_{20}) \\
\partial_t v_5 &= \gamma\alpha \mathrm{O} v_2 + \gamma(\Delta \otimes \mathbb{I}) v_5 + \frac{D}{2}(\mathbb{I} \otimes \Delta) v_4 - \frac{1}{D}(\mathbb{I} \otimes B) v_{18} - \gamma(B \otimes \mathbb{I})(v_{23} + v_{27}) \\
\partial_t v_6 &= -\frac{D}{2}\left[(1 \otimes \Delta) v_8 + (\Delta \otimes 1) v_9\right] + \frac{1}{D}\big[(B \otimes \mathbb{I}) v_{10} + (\mathbb{I} \otimes B) v_{14}\big] \\
\partial_t v_7 &= \frac{D}{2}\left[(\Delta \otimes 1) v_8 + (1 \otimes \Delta) v_9\right] - \frac{1}{D}\big[(B \otimes \mathbb{I}) v_{11} + (\mathbb{I} \otimes B) v_{17}\big] \\
\partial_t v_8 &= \frac{D}{2}\left[(1 \otimes \Delta) v_6 - (\Delta \otimes 1) v_7\right] + \frac{1}{D}\big[(B \otimes \mathbb{I}) v_{10} - (\mathbb{I} \otimes B) v_{14}\big] \\
\partial_t v_9 &= \frac{D}{2}\left[(\Delta \otimes 1) v_6 - (1 \otimes \Delta) v_7\right] - \frac{1}{D}\big[(B \otimes \mathbb{I}) v_{12} - (\mathbb{I} \otimes B) v_{15}\big] . \\
\partial_t v_{10} &= \gamma \boldsymbol{\Delta}^{(1)} v_{10} + \frac{D}{2}(\boldsymbol{\Delta}^{(2)} v_{12} - \boldsymbol{\Delta}^{(3)} v_{13}) + \gamma\alpha(O \otimes \mathbb{I}) v_9 \\
\partial_t v_{11} &= \gamma \boldsymbol{\Delta}^{(1)} v_{11} + \frac{D}{2}(-\boldsymbol{\Delta}^{(2)} v_{13} + \boldsymbol{\Delta}^{(3)} v_{12}) + \gamma\alpha(O \otimes \mathbb{I}) v_8 \\
\partial_t v_{12} &= \gamma \boldsymbol{\Delta}^{(1)} v_{12} - \frac{D}{2}(-\boldsymbol{\Delta}^{(2)} v_{10} + \boldsymbol{\Delta}^{(3)} v_{11}) + \gamma\alpha(O \otimes \mathbb{I}) v_6 \\
\partial_t v_{13} &= \gamma \boldsymbol{\Delta}^{(1)} v_{13} + \frac{D}{2}(-\boldsymbol{\Delta}^{(2)} v_{11} - \boldsymbol{\Delta}^{(3)} v_{10}) + \gamma\alpha(O \otimes \mathbb{I}) v_7 \\
\partial_t v_{14} &= -\frac{D}{2}\boldsymbol{\Delta}^{(1)} v_{15} + \gamma \boldsymbol{\Delta}^{(2)} v_{14} + \frac{D}{2}\boldsymbol{\Delta}^{(3)} v_{16}) + \gamma\alpha(\mathbb{I} \otimes O) v_8 \\
\partial_t v_{15} &= \frac{D}{2}\boldsymbol{\Delta}^{(1)} v_{15} + \gamma \boldsymbol{\Delta}^{(2)} v_{15} + \frac{D}{2}\boldsymbol{\Delta}^{(3)} v_{17}) + \gamma\alpha(\mathbb{I} \otimes O) v_7 \\
\partial_t v_{16} &= -\frac{D}{2}\boldsymbol{\Delta}^{(1)} v_{17} + \gamma \boldsymbol{\Delta}^{(2)} v_{16} - \frac{D}{2}\boldsymbol{\Delta}^{(3)} v_{14}) + \gamma\alpha(\mathbb{I} \otimes O) v_6 \\
\partial_t v_{17} &= \frac{D}{2}\boldsymbol{\Delta}^{(1)} v_{16} + \gamma \boldsymbol{\Delta}^{(2)} v_{17} - \frac{D}{2}\boldsymbol{\Delta}^{(3)} v_{15}) + \gamma\alpha(\mathbb{I} \otimes O) v_9 \\
\partial_t v_{18} &= \gamma(\boldsymbol{\Delta}^{(1)} + \boldsymbol{\Delta}^{(2)}) v_{18} - \frac{D}{2}\boldsymbol{\Delta}^{(3)} v_{19} + \gamma\alpha O^2 v_1 \\
\partial_t v_{19} &= \gamma(\boldsymbol{\Delta}^{(1)} + \boldsymbol{\Delta}^{(2)}) v_{19} + \frac{D}{2}\boldsymbol{\Delta}^{(3)} v_{18} + \gamma\alpha O^2 v_2 \\
\partial_t v_{20} &= -\frac{D}{2}\big(\boldsymbol{\Delta}^{(1)} v_{21} + \boldsymbol{\Delta}^{(2)} v_{22} + \boldsymbol{\Delta}^{(3)} v_{23}\big) \\
\partial_t v_{21} &= \frac{D}{2}\big(\boldsymbol{\Delta}^{(1)} v_{20} - \boldsymbol{\Delta}^{(2)} v_{26} - \boldsymbol{\Delta}^{(3)} v_{25}\big) \\
\partial_t v_{22} &= \frac{D}{2}\big(- \boldsymbol{\Delta}^{(1)} v_{26} + \boldsymbol{\Delta}^{(2)} v_{20} - \boldsymbol{\Delta}^{(3)} v_{23}\big) \\
\partial_t v_{23} &= \frac{D}{2}\big(- \boldsymbol{\Delta}^{(1)} v_{25} - \boldsymbol{\Delta}^{(2)} v_{24} + \boldsymbol{\Delta}^{(3)} v_{20}\big) \\
\partial_t v_{24} &= \frac{D}{2}\big(- \boldsymbol{\Delta}^{(1)} v_{27} + \boldsymbol{\Delta}^{(2)} v_{23} + \boldsymbol{\Delta}^{(3)} v_{22}\big) \\
\partial_t v_{25} &= \frac{D}{2}\big(\boldsymbol{\Delta}^{(1)} v_{23} - \boldsymbol{\Delta}^{(2)} v_{27} + \boldsymbol{\Delta}^{(3)} v_{21}\big) \\
\partial_t v_{26} &= \frac{D}{2}\big(\boldsymbol{\Delta}^{(1)} v_{22} + \boldsymbol{\Delta}^{(2)} v_{21} - \boldsymbol{\Delta}^{(3)} v_{27}\big) \\
\partial_t v_{27} &= \frac{D}{2}\big(- \boldsymbol{\Delta}^{(1)} v_{24} + \boldsymbol{\Delta}^{(2)} v_{25} + \boldsymbol{\Delta}^{(3)} v_{26}\big)
\end{aligned}
\tag{B.11}
$$

References

[1] R. E. Angulo and O. Hahn, "Large-scale dark matter simulations," *Living Reviews in Computational Astrophysics*, vol. 8, p. 1, 2022. DOI: 10.1007/s41115-021-00013-z.

[2] F. Bernardeau, S. Colombi, E. Gaztañaga, and R. Scoccimarro, "Large-scale structure of the universe and cosmological perturbation theory," *Physics Reports*, vol. 367, no. 1-3, pp. 1–248, 2002. DOI: 10.1016/S0370-1573(02)00135-7.

[3] W. Cui, A. Knebe, G. Yepes, *et al.*, "The large-scale environment from cosmological simulations – i. the baryonic cosmic web," *Monthly Notices of the Royal Astronomical Society*, vol. 473, pp. 68–79, 2018. DOI: 10.1093/mnras/stx2323.

[4] P. Natarajan, U. Chadayammuri, M. Jauzac, *et al.*, "Mapping substructure in the hst frontier fields cluster lenses and in cosmological simulations," *Monthly Notices of the Royal Astronomical Society*, vol. 468, no. 2, pp. 1962–1980, Feb. 2017, ISSN: 0035-8711. DOI: 10.1093/mnras/stw3385. eprint: https://academic.oup.com/mnras/article-pdf/468/2/1962/11210742/stw3385.pdf. [Online]. Available: https://doi.org/10.1093/mnras/stw3385.

[5] V. Springel, S. D. M. White, A. Jenkins, *et al.*, "Simulations of the formation, evolution and clustering of galaxies and quasars," *Nature*, vol. 435, pp. 629–636, 2005. DOI: 10.1038/nature03597.

[6] M. Vogelsberger, F. Marinacci, P. Torrey, and E. Puchwein, "Cosmological simulations of galaxy formation," *Nature Reviews Physics*, vol. 2, pp. 42–66, 2020. DOI: 10.1038/s42254-019-0127-2.

[7] Planck Collaboration, N. Aghanim, and other, "Planck 2018 results. VI. Cosmological parameters," *Astronomy and Astrophysics*, vol. 641, A6, A6, Sep. 2020. DOI: 10.1051/0004-6361/201833910. arXiv: 1807.06209 [astro-ph.CO].

[8] A. G. Riess and et al., "Observational evidence from supernovae for an accelerating universe and a cosmological constant," *The Astronomical Journal*, vol. 116, pp. 1009–1038, 1998. DOI: 10.1086/300499.

[9] S. Perlmutter and et al., "Measurements of Ω and Λ from 42 high-redshift supernovae," *The Astrophysical Journal*, vol. 517, pp. 565–586, 1999. DOI: 10.1086/307221.

[10] M. Tegmark and et al., "Cosmological parameters from sdss and wmap," *Physical Review D*, vol. 69, p. 103 501, 2004. DOI: 10.1103/PhysRevD.69.103501.

[11] H. Hoekstra and B. Jain, "Weak gravitational lensing and its cosmological applications," *Annual Review of Nuclear and Particle Science*, vol. 58, pp. 99–123, 2008. DOI: 10.1146/annurev.nucl.58.110707.171151.

[12] H. Mo, F. C. van den Bosch, and S. White, *Galaxy Formation and Evolution*. Cambridge University Press, 2010.

[13] S. Veilleux, G. Cecil, and J. Bland-Hawthorn, "Galactic winds," *Annual Review of Astronomy and Astrophysics*, vol. 43, pp. 769–826, 2005. DOI: 10.1146/annurev.astro.43.072103.150610. [Online]. Available: https://doi.org/10.1146/annurev.astro.43.072103.150610.

[14] A. C. Fabian, "Observational evidence of active galactic nuclei feedback," *Annual Review of Astronomy and Astrophysics*, vol. 50, pp. 455–489, 2012. DOI: 10.1146/annurev-astro-081811-125521. [Online]. Available: https://doi.org/10.1146/annurev-astro-081811-125521.

[15] V. Springel and L. Hernquist, "Cosmological smoothed particle hydrodynamics simulations: A hybrid multiphase model for star formation," *Monthly Notices of the Royal Astronomical Society*, vol. 339, pp. 289–311, 2003. DOI: 10.1046/j.1365-8711.2003.06206.x.

[16] J. Jasche and B. D. Wandelt, "Bayesian physical reconstruction of initial conditions from large-scale structure surveys," *Monthly Notices of the Royal Astronomical Society*, vol. 432, pp. 894–913, 2013. DOI: 10.1093/mnras/stt449.

[17] J. G. Sorce, H. M. Courtois, S. Gottlöber, Y. Hoffman, and R. B. Tully, "Simulations of the local universe constrained by observational peculiar velocities," *Monthly Notices of the Royal Astronomical Society*, vol. 437, no. 4, pp. 3586–3595, Dec. 2013, ISSN: 0035-8711. DOI: 10.1093/mnras/stt2153. eprint: https://academic.oup.com/mnras/article-pdf/437/4/3586/18498456/stt2153.pdf. [Online]. Available: https://doi.org/10.1093/mnras/stt2153.

[18] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010. DOI: 10.1017/CBO9780511976667.

[19] M. Cerezo, A. Arrasmith, R. Babbush, *et al.*, "Variational quantum algorithms," *Nature Reviews Physics*, vol. 3, no. 9, pp. 625–644, 2021.

[20] J. Preskill, "Quantum computing in the nisq era and beyond," *Quantum*, vol. 2, p. 79, 2018. DOI: 10.22331/q-2018-08-06-79.

[21] Y. Cao, J. Romero, J. P. Olson, *et al.*, "Quantum chemistry in the age of quantum computing," *Chemical Reviews*, vol. 119, no. 19, pp. 10 856–10 915, 2019. DOI: 10.1021/acs.chemrev.8b00803. [Online]. Available: https://doi.org/10.1021/acs.chemrev.8b00803.

[22] L. Cappelli, G. Murante, and S. Borgani, "Numerical limits in the integration of vlasov-poisson equation for cold dark matter," in *2025 33rd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP)*, 2025, pp. 431–438. DOI: 10.1109/PDP66500.2025.00067.

[23] M. Lubasch, J. Joo, P. Moinier, M. Kiffner, and D. Jaksch, "Variational quantum algorithms for nonlinear problems," *Physical Review A*, vol. 101, no. 1, 010301(R), Jan. 2020, arXiv:1907.09032 [quant-ph], ISSN: 2469-9926, 2469-9934. DOI: 10.1103/PhysRevA.101.010301. [Online]. Available: http://arxiv.org/abs/1907.09032 (visited on 05/16/2022).

[24] P. J. Ollitrault, S. Jandura, A. Miessen, *et al.*, "Quantum algorithms for grid-based variational time evolution," Mar. 2022, Number: arXiv:2203.02521 arXiv:2203.02521 [quant-ph]. [Online]. Available: http://arxiv.org/abs/2203.02521 (visited on 05/16/2022).

[25] L. Cappelli, F. Tacchino, G. Murante, S. Borgani, and I. Tavernelli, "From vlasov-poisson to schrödinger-poisson: Dark matter simulation with a quantum variational time evolution algorithm," *Phys. Rev. Res.*, vol. 6, p. 013 282, 1 Mar. 2024. DOI: 10.1103/PhysRevResearch.6.013282. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevResearch.6.013282.

[26] J. Einasto and M. Einasto, "Dark Matter in Groups and Clusters of Galaxies," *IAU Colloquium 174: Small Galaxy Groups*, 2000.

[27] D. H. Weinberg, M. J. Mortonson, D. J. Eisenstein, C. Hirata, A. G. Riess, and E. Rozo, "Observational probes of cosmic acceleration," *Physical Reports*, vol. 530, no. 2, pp. 87–255, Sep. 2013. DOI: 10.1016/j.physrep.2013.05.001. arXiv: 1201.2434 [astro-ph.CO].

[28] P. J. E. Peebles, *The Large-Scale Structure of the Universe*. Princeton, NJ: Princeton University Press, 1980.

[29] P. J. E. Peebles, *Principles of Physical Cosmology*. Princeton, NJ: Princeton University Press, 1993.

[30] J. R. Bond, L. Kofman, and D. Pogosyan, "How filaments are woven into the cosmic web," *Nature*, vol. 380, pp. 603–606, 1996. DOI: 10.1038/380603a0.

[31] P. Coles and F. Lucchin, *Cosmology: The Origin and Evolution of Cosmic Structure*. Chichester, UK: Wiley, 2002.

[32] P. Mocz and A. Szasz, "Towards Cosmological Simulations of Dark Matter on Quantum Computers," *The Astrophysical Journal*, vol. 910, no. 1, p. 29, Mar. 2021, arXiv:2101.05821 [astro-ph, physics:quant-ph], ISSN: 0004-637X, 1538-4357. DOI: 10.3847/1538-4357/abe6ac. [Online]. Available: http://arxiv.org/abs/2101.05821 (visited on 05/16/2022).

[33] L. Cappelli, F. Tacchino, G. Murante, S. Borgani, and I. Tavernelli, "From vlasov-poisson to schrödinger-poisson: Dark matter simulation with a quantum variational time evolution algorithm," *Phys. Rev. Res.*, vol. 6, p. 013 282, 1 Mar. 2024. DOI: 10.1103/PhysRevResearch.6.013282. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevResearch.6.013282.

[34] J. S. Bullock and M. Boylan-Kolchin, "Small-scale challenges to the Λcdm paradigm," *Annual Review of Astronomy and Astrophysics*, vol. 55, pp. 343–387, 2017. DOI: 10.1146/annurev-astro-091916-055313.

[35] W. J. G. de Blok, "The core-cusp problem," *Advances in Astronomy*, vol. 2010, p. 789 293, 2010. DOI: 10.1155/2010/789293.

[36] J. S. Bullock, "Notes on the missing satellites problem," in *Local Group Cosmology*, D. Martínez-Delgado, Ed. Cambridge University Press, 2013, pp. 95–122. DOI: 10.1017/CBO9781139152303.004.

[37] M. Boylan-Kolchin, J. S. Bullock, and M. Kaplinghat, "Too big to fail? The puzzling darkness of massive milky way subhaloes," *Monthly Notices of the Royal Astronomical Society: Letters*, vol. 415, no. 1, pp. L40–L44, 2011. DOI: 10.1111/j.1745-3933.2011.01074.x. arXiv: 1103.0007 [astro-ph.CO].

[38] T. K. Chan, D. Keres, J. Onorbe, *et al.*, "The impact of baryonic physics on the structure of dark matter halos: The view from the fire cosmological simulations," *Monthly Notices of the Royal Astronomical Society*, vol. 454, Jul. 2015. DOI: 10.1093/mnras/stv2165.

[39] A. M. Brooks, M. Kuhlen, A. Zolotov, and D. Hooper, "A baryonic solution to the missing satellites problem," *The Astrophysical Journal*, vol. 765, no. 1, p. 22, 2013. DOI: 10.1088/0004-637X/765/1/22.

[40] L. Hui, J. P. Ostriker, S. Tremaine, and E. Witten, "Ultralight scalars as cosmological dark matter," *Physical Review D*, vol. 95, no. 4, p. 043 541, 2017. DOI: 10.1103/PhysRevD.95.043541. arXiv: 1610.08297 [astro-ph.CO].

[41] A. G. Riess, W. Yuan, L. M. Macri, *et al.*, "A comprehensive measurement of the local value of the hubble constant with 1 km s-1 mpc-1 uncertainty from the hubble space telescope and the sh0es team," *The Astrophysical Journal Letters*, vol. 934, no. 1, p. L7, Jul. 2022. DOI: 10.3847/2041-8213/ac5c5b. [Online]. Available: https://doi.org/10.3847/2041-8213/ac5c5b.

[42] H. Hildebrandt, M. Viola, C. Heymans, *et al.*, "Kids-450: Cosmological parameter constraints from tomographic weak gravitational lensing," *Monthly Notices of the Royal Astronomical Society*, vol. 465, no. 2, pp. 1454–1498, 2017. DOI: 10.1093/mnras/stw2805.

[43] L. Perivolaropoulos and F. Skara, "Challenges for Λcdm: An update," *New Astronomy Reviews*, vol. 95, p. 101 659, 2022. DOI: 10.1016/j.newar.2022.101659. arXiv: 2105.05208 [astro-ph.CO].

[44] J. R. Bond and A. S. Szalay, "The collisionless damping of density fluctuations in an expanding universe," *The Astrophysical Journal*, vol. 274, pp. 443–468, 1980. DOI: 10.1086/161460.

[45] G. R. Blumenthal, S. M. Faber, J. R. Primack, and M. J. Rees, "Formation of galaxies and large-scale structure with cold dark matter," *Nature*, vol. 311, pp. 517–525, 1984. DOI: 10.1038/311517a0.

[46] M. Davis, G. Efstathiou, C. S. Frenk, and S. D. M. White, "The evolution of large-scale structure in a universe dominated by cold dark matter," *The Astrophysical Journal*, vol. 292, pp. 371–394, 1985. DOI: 10.1086/163168.

[47] P. Bode, J. P. Ostriker, and N. Turok, "Halo Formation in Warm Dark Matter Models," *The Astrophysical Journal*, vol. 556, no. 1, pp. 93–107, Jul. 2001. DOI: 10.1086/321541. arXiv: astro-ph/0010389 [astro-ph].

[48] D. N. Spergel and P. J. Steinhardt, "Observational evidence for self-interacting cold dark matter," *Physical Review Letters*, vol. 84, no. 17, pp. 3760–3763, 2000. DOI: 10.1103/PhysRevLett.84.3760.

[49] K. N. Abazajian, "Sterile neutrinos in cosmology," *Physiscs Reports*, vol. 711, pp. 1–28, Nov. 2017. DOI: 10.1016/j.physrep.2017.10.003. arXiv: 1705.01837 [hep-ph].

[50] S. Tulin and H.-B. Yu, "Dark matter self–interactions and small scale structure," *Physics Reports*, vol. 730, pp. 1–57, 2018. DOI: 10.1016/j.physrep.2017.11.004. arXiv: 1705.02358.

[51] J. Binney and S. Tremaine, *Galactic Dynamics: Second Edition*. Priceton University Press, 2008.

[52] R. Hockney and J. Eastwood, *Computer Simulation Using Particles*. CRC Press., 1988. DOI: https://doi.org/10.1201/9780367806934.

[53] P. Mocz, L. Lancaster, A. Fialkov, F. Becerra, and P.-H. Chavanis, "Schrödinger-Poisson–Vlasov-Poisson correspondence," *Physical Review D*, vol. 97, no. 8, p. 083 519, Apr. 2018, ISSN: 2470-0010, 2470-0029. DOI: 10.1103/PhysRevD.97.083519. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevD.97.083519 (visited on 05/16/2022).

[54] E. Bertschinger, "Self-similar secondary infall and accretion in an Einstein-de Sitter universe," *The Astrophysical Journal Supplements Series*, vol. 58, pp. 39–65, May 1985. DOI: 10.1086/191028.

[55] S. F. Shandarin and Y. B. Zeldovich, "The large-scale structure of the universe: Turbulence, intermittency, structures in a self-gravitating medium," *Reviews of Modern Physics*, vol. 61, no. 2, pp. 185–220, Apr. 1989. DOI: 10.1103/RevModPhys.61.185.

[56] O. Hahn and R. E. Angulo, "An adaptively refined phase-space element method for cosmological simulations and collisionless dynamics," *Monthly Notices of the Royal Astronomical Society*, vol. 455, no. 1, pp. 1115–1133, Jan. 2016. DOI: 10.1093/mnras/stv2304. arXiv: 1501.01959 [astro-ph.CO].

[57] K. Dolag, S. Borgani, S. Schindler, A. Diaferio, and A. M. Bykov, "Simulation Techniques for Cosmological Simulations," *Space Science Reviews*, vol. 134, no. 1-4, pp. 229–268, Feb. 2008. DOI: 10.1007/s11214-008-9316-5. arXiv: 0801.1023 [astro-ph].

[58] C. Cheng and G. Knorr, "The integration of the vlasov equation in configuration space," *Journal of Computational Physics*, vol. 22, no. 3, pp. 330–351, 1976, ISSN: 0021-9991. DOI: https://doi.org/10.1016/0021-9991(76)90053-X. [Online]. Available: https://www.sciencedirect.com/science/article/pii/002199917690053X.

[59] E. Sonnendrücker, J. Roche, P. Bertrand, and A. Ghizzo, "The semi-lagrangian method for the numerical resolution of the vlasov equation," *Journal of Computational Physics*, vol. 149, no. 2, pp. 201–220, 1999, ISSN: 0021-9991. DOI: https://doi.org/10.1006/jcph.1998.6148. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999198961484.

[60] F. Filbet, E. Sonnendrücker, and P. Bertrand, "Conservative numerical schemes for the vlasov equation," *Journal of Computational Physics*, vol. 172, no. 1, pp. 166–187, 2001, ISSN: 0021-9991. DOI: https://doi.org/10.1006/jcph.2001.6818. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999101968184.

[61] K. Yoshikawa, N. Yoshida, and M. Umemura, "Direct integration of the collisionless boltzmann equation in six-dimensional phase space: Self-gravitating systems," *The Astrophysical Journal*, vol. 762, no. 2, p. 116, Dec. 2012. DOI: 10.1088/0004-637X/762/2/116. [Online]. Available: https://dx.doi.org/10.1088/0004-637X/762/2/116.

[62] M. Garny and T. Konstandin, "Gravitational collapse in the schrödinger-poisson system," *Journal of Cosmology and Astroparticle Physics*, vol. 2018, no. 01, pp. 009–009, Jan. 2018. DOI: 10.1088/1475-7516/2018/01/009. [Online]. Available: https://doi.org/10.1088/1475-7516/2018/01/009.

[63] V. Springel, S. D. M. White, A. Jenkins, *et al.*, "Simulations of the formation, evolution and clustering of galaxies and quasars," *Nature*, vol. 435, no. 7042, pp. 629–636, Jun. 2005. DOI: 10.1038/nature03597. arXiv: astro-ph/0504097 [astro-ph].

[64] V. Springel, C. S. Frenk, and S. D. White, "The large-scale structure of the universe," *nature*, vol. 440, no. 7088, pp. 1137–1144, 2006.

[65] J. Barnes and P. Hut, "A hierarchical O(N log N) force-calculation algorithm," *Nature*, Dec. 1986, ISSN: 1476-4687. DOI: 10.1038/324446a0. [Online]. Available: https://doi.org/10.1038/324446a0.

[66] L. M. Widrow and N. Kaiser, "Using the Schroedinger Equation to Simulate Collisionless Matter," *The Astrophysical Journal Letters*, vol. 416, p. L71, Oct. 1993. DOI: 10.1086/187073.

[67] B. Schwabe, M. Gosenca, C. Behrens, J. C. Niemeyer, and R. Easther, "Simulating mixed fuzzy and cold dark matter," *Physical Review D*, vol. 102, no. 8, p. 083 518, 2020.

[68] K. Husimi, "Some formal properties of the density matrix," *Proceedings of the Physico-Mathematical Society of Japan. 3rd Series*, vol. 22, no. 4, pp. 264–314, 1940. DOI: 10.11429/ppmsj1919.22.4_264.

[69] P. Bertrand, N. van Tuan, M. Gros, B. Izrar, M. Feix, and J. Gutierrez, "Classical vlasov plasma description through quantum numerical methods," *Journal of Plasma Physics*, vol. 23, no. 3, pp. 401–422, 1980. DOI: 10.1017/S002237780002242X.

[70] M. Nori and M. Baldi, "AX-GADGET: a new code for cosmological simulations of Fuzzy Dark Matter and Axion models," *Monthly Notices of the Royal Astronomical Society*, vol. 478, no. 3, pp. 3935–3951, Aug. 2018. DOI: 10.1093/mnras/sty1224. arXiv: 1801.08144 [astro-ph.CO].

[71] P. Mocz, A. Fialkov, M. Vogelsberger, *et al.*, "First Star-Forming Structures in Fuzzy Cosmic Filaments," *Physical Review Letters*, vol. 123, no. 14, 141301, p. 141 301, Oct. 2019. DOI: 10.1103/PhysRevLett.123.141301. arXiv: 1910.01653 [astro-ph.GA].

[72] A. M. Turing, "On computable numbers, with an application to the entscheidungsproblem," *Proceedings of the London Mathematical Society*, vol. s2-42, no. 1, pp. 230–265, 1937. DOI: 10.1112/plms/s2-42.1.230.

[73] M. M. Mano and M. D. Ciletti, *Digital Design*, 6th. Boston, MA: Pearson, 2017, pp. 45–52, ISBN: 9780134549896.

[74] P. Horowitz and W. Hill, *The Art of Electronics*, 3rd. Cambridge, UK: Cambridge University Press, 2015, ISBN: 9780521809269.

[75] C. Gerry and P. Knight, *Introductory Quantum Optics*, 1st. Cambridge, UK: Cambridge University Press, 2005, ISBN: 9780521835740.

[76] D. Loss and D. P. DiVincenzo, "Quantum computation with quantum dots," *Physical Review A*, vol. 57, no. 1, pp. 120–126, 1998. DOI: 10.1103/PhysRevA.57.120.

[77] H. Häffner, C. Roos, and R. Blatt, "Quantum computing with trapped ions," *Physics Reports*, vol. 469, no. 4, pp. 155–203, 2008, ISSN: 0370-1573. DOI: https://doi.org/10.1016/j.physrep.2008.09.003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0370157308003463.

[78] J. Koch, T. M. Yu, J. Gambetta, *et al.*, "Charge-insensitive qubit design derived from the cooper pair box," *Physical Review A*, vol. 76, no. 4, p. 042 319, 2007. DOI: 10.1103/PhysRevA.76.042319.

[79] J. S. Bell, "On the einstein–podolsky–rosen paradox," *Physics*, vol. 1, no. 3, pp. 195–200, 1964. DOI: 10.1103/PhysicsPhysiqueFizika.1.195. [Online]. Available: https://doi.org/10.1103/PhysicsPhysiqueFizika.1.195.

[80] A. Barenco, C. H. Bennett, R. Cleve, *et al.*, "Elementary gates for quantum computation," *Physical Review A*, vol. 52, no. 5, pp. 3457–3467, Nov. 1995, arXiv:quant-ph/9503016, ISSN: 1050-2947, 1094-1622. DOI: 10.1103/PhysRevA.52.3457. [Online]. Available: http://arxiv.org/abs/quant-ph/9503016 (visited on 05/16/2022).

[81] J. Preskill, "Fault-tolerant quantum computation," in *Introduction to Quantum Computation and Information*, World Scientific, 1998, pp. 213–269.

[82] A. Somoroff, Q. Ficheux, R. A. Mencia, H. Xiong, R. Kuzmin, and V. E. Manucharyan, "Millisecond coherence in a superconducting qubit," *Phys. Rev. Lett.*, vol. 130, p. 267 001, 26 Jun. 2023. DOI: 10.1103/PhysRevLett.130.267001. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.130.267001.

[83] C. D. Bruzewicz, J. Chiaverini, R. McConnell, and J. M. Sage, "Trapped-ion quantum computing: Progress and challenges," *Applied Physics Reviews*, vol. 6, no. 2, p. 021 314, May 2019, ISSN: 1931-9401. DOI: 10.1063/1.5088164. eprint: https://pubs.aip.org/aip/apr/article-pdf/doi/10.1063/1.5088164/19742554/021314_1_online.pdf. [Online]. Available: https://doi.org/10.1063/1.5088164.

[84] F. Filbet and E. Sonnendrücker, "Comparison of eulerian vlasov solvers," *Computer Physics Communications*, vol. 150, no. 3, pp. 247–266, 2003, ISSN: 0010-4655. DOI: https://doi.org/10.1016/S0010-4655(02)00694-X. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S001046550200694X.

[85] D. Yi and S. Bu, "A mass conservative scheme for solving the vlasov–poisson equation using characteristic curve," *Journal of Computational and Applied Mathematics*, vol. 324, pp. 1–16, 2017, ISSN: 0377-0427. DOI: https://doi.org/10.1016/j.cam.2017.04.019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377042717301899.

[86] A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy, "Uniformly high order accurate essentially non-oscillatory schemesiii," *Journal of Computational Physics*, vol. 71, no. 2, pp. 231–303, 1987. DOI: 10.1016/0021-9991(87)90031-3.

[87] A. Klimas and W. Farrell, "A splitting algorithm for vlasov simulation with filamentation filtration," *Journal of Computational Physics*, vol. 110, no. 1, pp. 150–163, 1994, ISSN: 0021-9991. DOI: https://doi.org/10.1006/jcph.1994.1011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999184710114.

[88] N. Schwersenz, V. Loaiza, T. Zimmermann, J. Madroñero, and S. Wimberger, "Comparison of two different integration methods for the (1+1)-dimensional schrödinger-poisson equation," *Computer Physics Communications*, vol. 300, p. 109 192, 2024. DOI: 10.1016/j.cpc.2024.109192. [Online]. Available: https://doi.org/10.1016/j.cpc.2024.109192.

[89] N. Developers, *Numba: Just-in-time compiler for python*, https://numba.pydata.org, 0.61.0.

[90] O. Hahn and R. E. Angulo, "An adaptively refined phase-space element method for cosmological simulations and collisionless dynamics," *arXiv e-prints*, 2015. eprint: 1501.01959.

[91] R. P. Feynman, "Simulating physics with computers," *International Journal of Theoretical Physics*, vol. 21, no. 6, pp. 467–488, Jun. 1982, ISSN: 1572-9575. DOI: 10.1007/BF02650179. [Online]. Available: https://doi.org/10.1007/BF02650179.

[92] F. Tacchino, A. Chiesa, S. Carretta, and D. Gerace, "Quantum computers as universal quantum simulators: State-of-the-art and perspectives," *Advanced Quantum Technologies*, vol. 3, no. 3, p. 1 900 052, 2020.

[93] A. Miessen, P. J. Ollitrault, F. Tacchino, and I. Tavernelli, "Quantum algorithms for quantum dynamics," *Nature Computational Science*, vol. 3, no. 1, pp. 25–37, 2023. [Online]. Available: https://doi.org/10.1038/s43588-022-00374-2.

[94] K. Mita, "Schrödinger's equation as a diffusion equation," *American Journal of Physics*, vol. 89, no. 5, pp. 500–510, 2021.

[95] I. Kassal, S. P. Jordan, P. J. Love, M. Mohseni, and A. Aspuru-Guzik, "Polynomial-time quantum algorithm for the simulation of chemical dynamics," *Proceedings of the National Academy of Sciences*, vol. 105, no. 48, pp. 18 681–18 686, Dec. 2008, arXiv:0801.2986 [quant-ph], ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.0808245105. [Online]. Available: http://arxiv.org/abs/0801.2986 (visited on 05/17/2022).

[96] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for linear systems of equations," *Phys. Rev. Lett.*, vol. 103, p. 150 502, 15 Oct. 2009. DOI: 10.1103/PhysRevLett.103.150502. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.103.150502.

[97] A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon, "Simulated Quantum Computation of Molecular Energies," *Science*, vol. 309, no. 5741, pp. 1704–1707, Sep. 2005, ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.1113479. [Online]. Available: https://www.science.org/doi/10.1126/science.1113479 (visited on 05/17/2022).

[98] R. D. Somma, "Quantum simulations of one dimensional quantum systems," *Quantum Info. Comput.*, vol. 16, no. 13–14, pp. 1125–1168, Oct. 2016, ISSN: 1533-7146.

[99] P. J. Ollitrault, G. Mazzola, and I. Tavernelli, "Non-adiabatic molecular quantum dynamics with quantum computers," *Physical Review Letters*, vol. 125, no. 26, p. 260 511, Dec. 2020, arXiv:2006.09405 [quant-ph], ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.125.260511. [Online]. Available: http://arxiv.org/abs/2006.09405 (visited on 05/16/2022).

[100] A. Tranter, P. J. Love, F. Mintert, N. Wiebe, and P. V. Coveney, "Ordering of trotterization: Impact on errors in quantum simulation of electronic structure," *arXiv preprint*, 2019. arXiv: 1912.07555 [quant-ph].

[101] R. Lorenz and J. Barrett, "Causal and compositional structure of unitary transformations," *Quantum*, vol. 5, p. 511, Jul. 2021, ISSN: 2521-327X. DOI: 10.22331/q-2021-07-28-511. [Online]. Available: https://doi.org/10.22331/q-2021-07-28-511.

[102] J. Gacon, J. Nys, R. Rossi, S. Woerner, and G. Carleo, *Variational quantum time evolution without the quantum geometric tensor*, 2023. arXiv: 2303.12839 [quant-ph].

[103] X. Yuan, S. Endo, Q. Zhao, Y. Li, and S. Benjamin, "Theory of variational quantum simulation," *Quantum*, vol. 3, p. 191, Oct. 2019, arXiv:1812.08767 [quant-ph], ISSN: 2521-327X. DOI: 10.22331/q-2019-10-07-191. [Online]. Available: http://arxiv.org/abs/1812.08767 (visited on 05/16/2022).

[104] One way of implementing a quantum spectral method for the solution of the SP equation would require a quantum Fourier transform (QFT) to move in the momentum space. Then a circuit able to reproduce $|\psi|^2 - 1$ would need to be followed by one able to divide for the squared momenta k^2. In the end a QFT^{-1} would return the exact potential on the quantum register.

[105] S. Chakrabarti, R. Krishnakumar, G. Mazzola, N. Stamatopoulos, S. Woerner, and W. J. Zeng, "A Threshold for Quantum Advantage in Derivative Pricing," *Quantum*, vol. 5, p. 463, Jun. 2021, arXiv:2012.03819 [quant-ph, q-fin], ISSN: 2521-327X. DOI: 10.22331/q-2021-06-01-463. [Online]. Available: http://arxiv.org/abs/2012.03819 (visited on 05/16/2022).

[106] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, "Evaluating analytic gradients on quantum hardware," *Physical Review A*, vol. 99, no. 3, p. 032 331, Mar. 2019, ISSN: 2469-9926, 2469-9934. DOI: 10.1103/PhysRevA.99.032331. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.99.032331 (visited on 05/16/2022).

[107] *Scipy least square solver v1.8.0*, https://docs.scipy.org/doc/scipy-1.8.0/reference/generated/scipy.linalg.lstsq.html.

[108] R. Jozsa, "Fidelity for mixed quantum states," *Journal of Modern Optics*, vol. 41, no. 12, pp. 2315–2323, 1994. DOI: 10.1080/09500349414552171. eprint: https://doi.org/10.1080/09500349414552171. [Online]. Available: https://doi.org/10.1080/09500349414552171.

[109] Qiskit contributors, *Qiskit: An open-source framework for quantum computing*, 2023. DOI: 10.5281/zenodo.2573505.

[110] T. Ishiyama, F. Prada, A. A. Klypin, *et al.*, "The Uchuu simulations: Data Release 1 and dark matter halo concentrations," *Monthly Notices of the Royal Astronomical Society*, vol. 506, no. 3, pp. 4210–4231, Sep. 2021. DOI: 10.1093/mnras/stab1755. arXiv: 2007.14720 [astro-ph.CO].

[111] Euclid Collaboration, T. Castro, and other, "Euclid preparation. XXIV. Calibration of the halo mass function in $\Lambda(\nu)$CDM cosmologies," *arXiv e-prints*, arXiv:2208.02174, arXiv:2208.02174, Aug. 2022. DOI: 10.48550/arXiv.2208.02174. arXiv: 2208.02174 [astro-ph.CO].

[112] R. E. Angulo, M. Zennaro, S. Contreras, G. Aricò, M. Pellejero-Ibañez, and J. Stücker, "The BACCO simulation project: exploiting the full power of large-scale structure for cosmology," *Monthly Notices of the Royal Astronomical Society*, vol. 507, no. 4, pp. 5869–5881, Nov. 2021. DOI: 10.1093/mnras/stab2018. arXiv: 2004.06245 [astro-ph.CO].

[113] A. D. Meglio and others (QC4HEP Working Group), "Quantum computing for high-energy physics: State of the art and challenges. summary of the qc4hep working group," 2023. arXiv: 2307.03236 [quant-ph].

[114] J. Gambetta, *Quantum-centric supercomputing: The next wave of computing*, https://research.ibm.com/blog/next-wave-quantum-centric-supercomputing, 2022.

[115] M. E. Hochstenbach, L. Reichel, and G. Rodriguez, "Regularization parameter determination for discrete ill-posed problems," *Journal of Computational and Applied Mathematics*, vol. 273, pp. 132–149, 2015.

[116] J. Lampe, L. Reichel, and H. Voss, "Large-scale tikhonov regularization via reduction by orthogonal projection," *Linear Algebra and its Applications*, vol. 436, no. 8, pp. 2845–2865, 2012, Special Issue dedicated to Danny Sorensen's 65th birthday, ISSN: 0024-3795. DOI: https://doi.org/10.1016/j.laa.2011.07.019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0024379511005313.

[117] M. Gavish and D. L. Donoho, "The optimal hard threshold for singular values is $4/\sqrt{3}$," *IEEE Transactions on Information Theory*, vol. 60, no. 8, pp. 5040–5053, 2014. DOI: 10.1109/TIT.2014.2323359.

[118] X. Li, X. Yin, N. Wiebe, *et al.*, "Potential quantum advantage for simulation of fluid dynamics," *Physical Review Research*, vol. 7, no. 1, p. 013 036, 2025.

[119] C. Sanavio, W. A. Simon, A. Ralli, P. Love, and S. Succi, "Carleman-lattice-Boltzmann quantum circuit with matrix access oracles," *Physics of Fluids*, vol. 37, no. 3, p. 037 123, Mar. 2025, ISSN: 1070-6631. DOI: 10.1063/5.0254588. [Online]. Available: https://doi.org/10.1063/5.0254588 (visited on 05/24/2025).

[120] A. Mizel, "Critically Damped Quantum Search," *Physical Review Letters*, vol. 102, no. 15, p. 150 501, Apr. 2009, ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.102.150501. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.102.150501 (visited on 10/22/2024).

[121] B. Yan, S. Wei, H. Jiang, *et al.*, "Fixed-point oblivious quantum amplitude-amplification algorithm," *Scientific Reports*, vol. 12, no. 1, p. 14 339, Aug. 2022, Publisher: Nature Publishing Group, ISSN: 2045-2322. DOI: 10.1038/s41598-022-15093-x. [Online]. Available: https://www.nature.com/articles/s41598-022-15093-x (visited on 10/22/2024).

[122] V. Turau, "Fixed-radius near neighbors search," *Information Processing Letters*, vol. 39, no. 4, pp. 201–203, Aug. 1991, ISSN: 0020-0190. DOI: 10.1016/0020-0190(91)90180-P. [Online]. Available: https://www.sciencedirect.com/science/article/pii/002001909190180P (visited on 06/20/2025).

[123] L. Verlet, "Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules," *Physical Review*, vol. 159, no. 1, pp. 98–103, Jul. 1967, ISSN: 0031-899X. DOI: 10.1103/PhysRev.159.98. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRev.159.98 (visited on 05/17/2025).

[124] R. A. Gingold and J. J. Monaghan, "Smoothed particle hydrodynamics: Theory and application to non-spherical stars," *Monthly Notices of the Royal Astronomical Society*, vol. 181, no. 3, pp. 375–389, Dec. 1977, ISSN: 0035-8711. DOI: 10.1093/mnras/181.3.375. eprint: https://academic.oup.com/mnras/article-pdf/181/3/375/3104055/mnras181-0375.pdf. [Online]. Available: https://doi.org/10.1093/mnras/181.3.375.

[125] V. Springel, N. Yoshida, and S. D. White, "Gadget: A code for collisionless and gasdynamical cosmological simulations," *New Astronomy*, vol. 6, no. 2, pp. 79–117, 2001, ISSN: 1384-1076. DOI: https://doi.org/10.1016/S1384-1076(01)00042-2. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1384107601000422.

[126] M. J. Abraham, T. Murtola, R. Schulz, *et al.*, "Gromacs: High performance molecular simulations through multi-level parallelism from laptops to supercomputers," *SoftwareX*, vol. 1-2, pp. 19–25, 2015, ISSN: 2352-7110. DOI: https://doi.org/10.1016/j.softx.2015.06.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352711015000059.

[127] G. Di Ilio, D. Chiappini, S. Ubertini, G. Bella, and S. Succi, "Fluid flow around naca 0012 airfoil at low-reynolds numbers with hybrid lattice boltzmann method," *Computers & Fluids*, vol. 166, pp. 200–208, 2018, ISSN: 0045-7930. DOI: https://doi.org/10.1016/j.compfluid.2018.02.014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045793018300677.

[128] Z. Yao, J.-S. Wang, G.-R. Liu, and M. Cheng, "Improved neighbor list algorithm in molecular simulations using cell decomposition and data sorting method," *Computer Physics Communications*, vol. 161, no. 1, pp. 27–35, 2004, ISSN: 0010-4655. DOI: https://doi.org/10.1016/j.cpc.2004.04.004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010465504002097.

[129] X. Chen and S. Güttel, *Fast and exact fixed-radius neighbor search based on sorting*, arXiv:2212.07679 [cs], Jan. 2024. DOI: 10.48550/arXiv.2212.07679. [Online]. Available: http://arxiv.org/abs/2212.07679 (visited on 06/18/2025).

[130] L. Wang, Y. Wu, Y. Deng, *et al.*, "Accurate and reliable prediction of relative ligand binding potency in prospective drug discovery by way of a modern free-energy calculation protocol and force field," *Journal of the American Chemical Society*, vol. 137, no. 7, pp. 2695–2703, 2015. DOI: 10.1021/ja512751q.

[131] Y. Shan, E. T. Kim, M. P. Eastwood, R. O. Dror, M. A. Seeliger, and D. E. Shaw, "How does a drug molecule find its target binding site?" *Journal of the American Chemical Society*, vol. 133, no. 24, pp. 9181–9183, 2011. DOI: 10.1021/ja202726y. [Online]. Available: https://doi.org/10.1021/ja202726y.

[132] P. J. Bond and M. S. Sansom, "Insertion and assembly of membrane proteins via simulation," *Journal of the American Chemical Society*, 2006. DOI: 10.1021/ja0569104.

[133] C. Liu, L. Xue, and C. Song, "Calcium binding and permeation in trpv channels: Insights from molecular dynamics simulations," *Journal of General Physiology*, vol. 155, no. 12, e202213261, 2023, Epub 2023 Sep 20. DOI: 10.1085/jgp.202213261.

[134] H. Zhao, P. Duan, Z. Li, *et al.*, "Unveiling the multiscale dynamics of polymer vitrimers via molecular dynamics simulations," *Macromolecules*, vol. 56, no. 23, pp. 9336–9349, 2023. DOI: 10.1021/acs.macromol.3c01893.

[135] M. P. Howard, J. A. Anderson, A. Nikoubashman, S. C. Glotzer, and A. Z. Panagiotopoulos, "Efficient neighbor list calculation for molecular simulation of colloidal systems using graphics processing units," *Computer Physics Communications*, vol. 203, pp. 45–52, 2016, ISSN: 0010-4655. DOI: https://doi.org/10.1016/j.cpc.2016.02.003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010465516300182.

[136] A. J. Proctor, T. J. Lipscomb, A. Zou, J. A. Anderson, and S. S. Cho, "Performance Analyses of a Parallel Verlet Neighbor List Algorithm for GPU-Optimized MD Simulations," in *2012 ASE/IEEE International Conference on BioMedical Computing (BioMedCom)*, 2012, pp. 14–19. DOI: 10.1109/BioMedCom.2012.9.

[137] F. Chiti and C. M. Dobson, "Protein misfolding, amyloid formation, and human disease: A summary of progress over the last decade," *Annual Review of Biochemistry*, vol. 86, pp. 27–68, 2017. DOI: 10.1146/annurev-biochem-061516-045115.

[138] K. Lindorff-Larsen, S. Piana, R. O. Dror, and D. E. Shaw, "How fast-folding proteins fold," *Science*, vol. 334, no. 6055, pp. 517–520, 2011. DOI: 10.1126/science.1208351.

[139] J. Abramson, J. Adler, J. Dunger, *et al.*, "Accurate structure prediction of biomolecular interactions with alphafold 3," *Nature*, vol. 630, no. 8016, pp. 493–500, 2024. DOI: 10.1038/s41586-024-07487-w.

[140] K. Schütze, M. Heinzinger, M. Steinegger, and B. Rost, "Nearest neighbor search on embeddings rapidly identifies distant protein relations," *Frontiers in Bioinformatics*, vol. 2, p. 1 033 775, 2022. DOI: 10.3389/fbinf.2022.1033775.

[141] M. van Kempen, S. S. Kim, C. Tumescheit, *et al.*, "Fast and accurate protein structure search with foldseek," *Nature Biotechnology*, vol. 42, no. 2, pp. 243–246, 2024. DOI: 10.1038/s41587-023-01773-0.

[142] P. Goloubinoff, M. Sulpizi, A. Cappellaro, J. Dutheil, H. A. Nguyen, and P. De Los Rios, "Chaperones convert the energy from atp into the nonequilibrium stabilization of native proteins," *Nature Chemical Biology*, vol. 14, no. 4, pp. 388–395, 2018. DOI: 10.1038/s41589-018-0013-8.

[143] S. Assenza, A. S. Sassi, R. Kellner, P. De Los Rios, and A. Barducci, "Efficient conversion of chemical energy into mechanical work by hsp70 chaperones," *Physical Review X*, vol. 9, no. 4, p. 041 033, 2019. DOI: 10.1103/PhysRevX.9.041033.

[144] P. De Los Rios and A. Barducci, "Non-equilibrium protein folding and activation by atp-driven chaperones," *Nature Reviews Molecular Cell Biology*, vol. 15, no. 10, pp. 611–622, 2014. DOI: 10.1038/nrm3868.

[145] E. M. Sontag, R. S. Samant, and J. Frydman, "Mechanisms and functions of spatial protein quality control," *Annual Review of Biochemistry*, vol. 86, pp. 97–122, 2017. DOI: 10.1146/annurev-biochem-060815-014616.

[146] C. L. Klaips, G. G. Jayaraj, and F. U. Hartl, "Pathways of cellular proteostasis in aging and disease," *Journal of Cell Biology*, vol. 217, no. 1, pp. 51–63, 2018. DOI: 10.1083/jcb.201709072.

[147] A. C. Crespo, J. M. Dominguez, A. Barreiro, M. Gómez-Gesteira, and B. D. Rogers, "Gpus, a new tool of acceleration in cfd: Efficiency and reliability on smoothed particle hydrodynamics methods," *PLOS ONE*, vol. 6, no. 6, pp. 1–13, Jun. 2011. DOI: 10.1371/journal.pone.0020685. [Online]. Available: https://doi.org/10.1371/journal.pone.0020685.

[148] V. Springel, "The cosmological simulation code gadget-2," *Monthly notices of the royal astronomical society*, vol. 364, no. 4, pp. 1105–1134, 2005.

[149] Q. Zhu, L. Hernquist, and Y. Li, "Numerical convergence in smoothed particle hydrodynamics," *The Astrophysical Journal*, vol. 800, no. 1, p. 6, 2015. DOI: 10.1088/0004-637X/800/1/6.

[150] T. L. Cassell, T. Deakin, A. Alpay, V. Heuveline, and G. B. Gadeschi, "Efficient tree-based parallel algorithms for n-body simulations using c++ standard parallelism," in *SC24-W: Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2024, pp. 708–717. DOI: 10.1109/SCW63240.2024.00099.

[151] *Supercomputer fugaku - fujitsu*. [Online]. Available: https://www.fujitsu.com/global/about/innovation/fugaku/specifications/.

[152] T. Papatheodore, *Summit architecture overview*. [Online]. Available: https://www.olcf.ornl.gov/wp-content/uploads/2019/05/Summit_System_Overview_20190520.pdf?utm_source=chatgpt.com.

[153] J. I. Read and T. Hayfield, "Sphs: Smoothed particle hydrodynamics with a higher order dissipation switch," *Monthly Notices of the Royal Astronomical Society*, vol. 422, no. 4, pp. 3037–3055, May 2012, ISSN: 0035-8711. DOI: 10.1111/j.1365-2966.2012.20819.x. eprint: https://academic.oup.com/mnras/article-pdf/422/4/3037/18598311/mnras0422-3037.pdf. [Online]. Available: https://doi.org/10.1111/j.1365-2966.2012.20819.x.

[154] M. Castelli, F. Marchetti, S. Osuna, *et al.*, "Decrypting Allostery in Membrane-Bound K-Ras4B Using Complementary In Silico Approaches Based on Unbiased Molecular Dynamics Simulations," *Journal of the American Chemical Society*, vol. 146, no. 1, pp. 901–919, Jan. 2024, Publisher: American Chemical Society, ISSN: 0002-7863. DOI: 10.1021/jacs.3c11396. [Online]. Available: https://doi.org/10.1021/jacs.3c11396 (visited on 06/20/2025).

[155] D. Frenkel and B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications*, 2nd. Academic Press, 2002, ch. 3. DOI: https://doi.org/10.1016/B978-0-12-267351-1.X5000-7.

[156] L. K. Grover, *A fast quantum mechanical algorithm for database search*, arXiv:quant-ph/9605043, Nov. 1996. DOI: 10.48550/arXiv.quant-ph/9605043. [Online]. Available: http://arxiv.org/abs/quant-ph/9605043 (visited on 05/13/2025).

[157] L. K. Grover, "Fixed-Point Quantum Search," *Physical Review Letters*, vol. 95, no. 15, p. 150 501, Oct. 2005, ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.95.150501. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.95.150501 (visited on 10/15/2024).

[158] T. J. Yoder, G. H. Low, and I. L. Chuang, "Fixed-Point Quantum Search with an Optimal Number of Queries," *Phys. Rev. Lett.*, vol. 113, no. 21, p. 210 501, 2014. DOI: 10.1103/PhysRevLett.113.210501.

[159] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, "Exponential improvement in precision for simulating sparse Hamiltonians," in *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, arXiv:1312.1414 [quant-ph], May 2014, pp. 283–292. DOI: 10.1145/2591796.2591854. [Online]. Available: http://arxiv.org/abs/1312.1414 (visited on 03/19/2025).

[160] A. A. Zecchi, C. Sanavio, S. Perotto, and S. Succi, *Improved amplitude amplification strategies for the quantum simulation of classical transport problems*, arXiv:2502.18283 [quant-ph], Feb. 2025. DOI: 10.48550/arXiv.2502.18283. [Online]. Available: http://arxiv.org/abs/2502.18283 (visited on 03/05/2025).

[161] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, "Quantum singular value transformation and beyond: Exponential improvements for quantum matrix arithmetics," in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, Phoenix AZ USA: ACM, Jun. 2019, pp. 193–204, ISBN: 978-1-4503-6705-9. DOI: 10.1145/3313276.3316366. [Online]. Available: https://dl.acm.org/doi/10.1145/3313276.3316366 (visited on 10/09/2024).

[162] J. M. Martyn, Z. M. Rossi, A. K. Tan, and I. L. Chuang, "Grand Unification of Quantum Algorithms," *PRX Quantum*, vol. 2, no. 4, p. 040 203, Dec. 2021, ISSN: 2691-3399. DOI: 10.1103/PRXQuantum.2.040203. [Online]. Available: https://link.aps.org/doi/10.1103/PRXQuantum.2.040203 (visited on 10/09/2024).

[163] D. Cesarini, F. Pitari, F. Barbari, *et al.*, "Eupex deliverable d3.2 applications optimised for sve and hbm," EuropeanUnion, Tech. Rep., 2023.

[164] S. Habib, V. Morozov, N. Frontiere, *et al.*, "HACC: Extreme scaling and performance across diverse architectures," *Commun. ACM*, vol. 60, no. 1, pp. 97–104, 2017. DOI: 10.1145/3015569. [Online]. Available: https://cacm.acm.org/research/hacc/.

[165] J. N. F. Alves, L. M. S. Russo, and A. Francisco, "Cache-oblivious hilbert curve-based blocking scheme for matrix transposition," *ACM Transactions on Mathematical Software*, vol. 48, no. 4, pp. 1–28, 2022. DOI: https://doi.org/10.1145/3555353.

[166] G. Brassard, P. Høyer, and A. Tapp, "Quantum counting," in *Automata, Languages and Programming: 25th International Colloquium, ICALP'98 Aalborg, Denmark, July 13–17, 1998 Proceedings 25*, Springer, 1998, pp. 820–831.

[167] S. Lee and S. Y. Nam, "Finding all solutions with grover's algorithm by integrating estimation and discovery," *Electronics*, vol. 13, no. 23, 2024, ISSN: 2079-9292. DOI: 10.3390/electronics13234830. [Online]. Available: https://www.mdpi.com/2079-9292/13/23/4830.

[168] E. Malvetti, R. Iten, and R. Colbeck, "Quantum Circuits for Sparse Isometries," en, *Quantum*, vol. 5, p. 412, Mar. 2021, arXiv:2006.00016 [quant-ph], ISSN: 2521-327X. DOI: 10.22331/q-2021-03-15-412. [Online]. Available: http://arxiv.org/abs/2006.00016 (visited on 06/05/2025).

[169] F. Mozafari, H. Riener, M. Soeken, and G. De Micheli, "Efficient Boolean Methods for Preparing Uniform Quantum States," *IEEE Transactions on Quantum Engineering*, vol. 2, pp. 1–12, 2021, ISSN: 2689-1808. DOI: 10.1109/TQE.2021.3101663. [Online]. Available: https://ieeexplore.ieee.org/document/9506863 (visited on 06/07/2025).

[170] X.-M. Zhang, M.-H. Yung, and X. Yuan, "Low-depth quantum state preparation," *Physical Review Research*, vol. 3, no. 4, p. 043 200, Dec. 2021, Publisher: American Physical Society. DOI: 10.1103/PhysRevResearch.3.043200. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevResearch.3.043200 (visited on 06/06/2025).

[171] X.-M. Zhang, T. Li, and X. Yuan, "Quantum State Preparation with Optimal Circuit Depth: Implementations and Applications," *Physical Review Letters*, vol. 129, no. 23, p. 230 504, Nov. 2022, Publisher: American Physical Society. DOI: 10.1103/PhysRevLett.129.230504. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.129.230504 (visited on 06/06/2025).

[172] X.-M. Zhang and X. Yuan, "Circuit complexity of quantum access models for encoding classical data," en, *npj Quantum Information*, vol. 10, no. 1, pp. 1–12, Apr. 2024, Publisher: Nature Publishing Group, ISSN: 2056-6387. DOI: 10.1038/s41534-024-00835-8. [Online]. Available: https://www.nature.com/articles/s41534-024-00835-8 (visited on 06/06/2025).

[173] D. Camps, L. Lin, R. Van Beeumen, and C. Yang, *Explicit Quantum Circuits for Block Encodings of Certain Sparse Matrices*, en, arXiv:2203.10236 [quant-ph], May 2023. [Online]. Available: http://arxiv.org/abs/2203.10236 (visited on 06/21/2024).

[174] S. A. Cuccaro, T. G. Draper, S. A. Kutin, and D. P. Moulton, *A new quantum ripple-carry addition circuit*, en, Oct. 2004. [Online]. Available: https://arxiv.org/abs/quant-ph/0410184v1 (visited on 06/10/2025).

[175] E. Thula, *Quantum incrementer*, https://egrettathula.wordpress.com/2024/07/28/quantum-incrementer/, July 28 2024.

[176] T. G. Draper, S. A. Kutin, E. M. Rains, and K. M. Svore, "A logarithmic-depth quantum carry-lookahead adder," *Quantum Info. Comput.*, vol. 6, no. 4, pp. 351–369, Jul. 2006, ISSN: 1533-7146.

[177] E. F. Combarro, I. F. Rúa, F. Orts, G. Ortega, A. M. Puertas, and E. M. Garzón, "Quantum algorithms to compute the neighbour list of N-body simulations," *Quantum Information Processing*, vol. 23, no. 2, 61, p. 61, Feb. 2024. DOI: 10.1007/s11128-023-04245-1.

[178] V. Shende, S. Bullock, and I. Markov, "Synthesis of quantum-logic circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 6, pp. 1000–1010, Jun. 2006, ISSN: 1937-4151. DOI: 10.1109/tcad.2005.855930. [Online]. Available: http://dx.doi.org/10.1109/TCAD.2005.855930.

[179] R. Vale, T. M. D. Azevedo, I. C. S. Araújo, I. F. Araujo, and A. J. da Silva, "Circuit Decomposition of Multicontrolled Special Unitary Single-Qubit Gates," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 43, no. 3, pp. 802–811, Mar. 2024, Conference Name: IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, ISSN: 1937-4151. DOI: 10.1109/TCAD.2023.3327102. [Online]. Available: https://ieeexplore.ieee.org/document/10293178 (visited on 01/30/2025).

[180] B. Zindorf and S. Bose, *Multi-controlled quantum gates in linear nearest neighbor*, 2025. arXiv: 2506.00695 [quant-ph]. [Online]. Available: https://arxiv.org/abs/2506.00695.

[181] B. Zindorf and S. Bose, *Efficient Implementation of Multi-Controlled Quantum Gates*, arXiv:2404.02279 [quant-ph], Mar. 2025. DOI: 10.48550/arXiv.2404.02279. [Online]. Available: http://arxiv.org/abs/2404.02279 (visited on 06/10/2025).

[182] C. Figgatt, D. Maslov, K. A. Landsman, N. M. Linke, S. Debnath, and C. Monroe, "Complete 3-Qubit Grover search on a programmable quantum computer," *Nature Communications*, vol. 8, no. 1, p. 1918, Dec. 2017, ISSN: 2041-1723. DOI: 10.1038/s41467-017-01904-7. [Online]. Available: https://doi.org/10.1038/s41467-017-01904-7.

[183] K. Zhang, K. Yu, and V. Korepin, "Quantum search on noisy intermediate-scale quantum devices," *Europhysics Letters*, vol. 140, no. 1, p. 18 002, Sep. 2022. DOI: 10.1209/0295-5075/ac90e6. [Online]. Available: https://dx.doi.org/10.1209/0295-5075/ac90e6.

[184] J. Leng, F. Yang, and X.-B. Wang, "Noise-tolerant Grover's algorithm via success-probability prediction," *Phys. Rev. Res.*, vol. 7, p. L012017, 1 Jan. 2025. DOI: 10.1103/PhysRevResearch.7.L012017. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevResearch.7.L012017.

[185] S. Jaques and A. G. Rattew, "Qram: A survey and critique," *Quantum*, vol. 9, p. 1922, 2025. DOI: https://doi.org/10.22331/q-2025-12-02-1922.

[186] A. D. McLachlan, "A variational solution of the time-dependent schrödinger equation," *Molecular Physics*, vol. 8, pp. 39–44, 1964. DOI: 10.1080/00268976400100021.

[187] T. Carleman, "Application de la théorie des équations intégrales linéaires aux systèmes d'équations différentielles non linéaires," *Acta Mathematica*, vol. 59, pp. 63–87, 1932. DOI: 10.1007/BF02546499.

UNIVERSITÀ
DEGLI STUDI
DI TRIESTE

ATTENZIONE! COMPILARE LA FRASE SEGUENTE COME INDICATO NELLA MAIL
PLEASE NOTE! FILL IN THE FOLLOWING SENTENCE AS WRITTEN ON THE EMAIL WE SENT YOU

La borsa di dottorato è cofinanziata con risorse dell'Unione europea, NextGeneration EU - Piano Nazionale di Ripresa e Resilienza, Missione 4 – Componente 2 – Investimento 1.4 CUP C53C22000350006