

Simone Montangero

Introduction to Tensor Network Methods

Numerical Simulations of Low-
Dimensional Many-Body Quantum
Systems

 Springer

Introduction to Tensor Network Methods

Simone Montangero

Introduction to Tensor Network Methods

Numerical Simulations of Low-Dimensional
Many-Body Quantum Systems

 Springer

Simone Montangero
Department of Physics and Astronomy
“G. Galilei”
University of Padova
Padova, Italy

ISBN 978-3-030-01408-7 ISBN 978-3-030-01409-4 (eBook)
<https://doi.org/10.1007/978-3-030-01409-4>

Library of Congress Control Number: 2018955916

© Springer Nature Switzerland AG 2018

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

*To my family,
a bundle of brave and amazing persons,
spread around but always united together.*

Preface

In the last years, a number of theoretical and numerical tools have been developed by a thriving community formed by people coming from different backgrounds—condensed matter, quantum optics, quantum information, high-energy, and high-performance computing—which are opening new paths towards the exploration of correlated quantum matter. The field I have the pleasure to work in is mostly based and supported by the numerical simulation performed via tensor network methods, which sprang from the density matrix renormalization group, introduced by S. White more than twenty years ago. White’s contribution, as the name itself suggests, was based on the powerful theoretical construction introduced in the seventies on the renormalization group and critical phenomena by K. Wilson. After a first decade where density matrix renormalization group has been applied mostly in condensed matter physics, starting from the new millennium, tensor network methods have been developed and adapted to a constantly increasing number of research fields, ranging from quantum information and quantum chemistry to lattice gauge theories.

This book contains the notes of the course I delivered at Ulm University from 2010 to 2016 on computational quantum physics. I planned the course trying to fulfil two very demanding requirements. From the one hand, it is structured in such a way that a student would be able to follow it, even without previous knowledge on programming, scientific calculations and only knowing the basics of the underlying theory, that is, a basic course in quantum mechanics. On the other hand, I aimed not only to introduce the students to the fascinating field of computational physics, but also to achieve that, at the end of the course, they could be able to attack one of the main open problems in modern physics—the quantum many-body problem. Thus, I designed a course that would bring the interested students from the very first steps into the world of computational physics to its cutting edge, at the point where they could—with a little additional effort—dive in into the world of research. Indeed, a student that would take such a challenge with a proper know-how will surely have plenty of occasions to enjoy this fast expanding field and will belong to an exciting and growing community. In the last ten years, I have

already witnessed many students that are succeeding in such process, which resulted in many fruitful scientific collaborations and interesting publications.

The book is structured in parts, each of them divided into chapters, according to the original plan of the course. The first part and the appendices introduce the basic concepts needed in any computational physics courses: software and hardware, programming guidelines, linear algebra and differential calculus. They are presented in a self-consistent way and accompanied by exercises that will make it easier to climb the steepest learning curve of the second part of the course. The central part presents the core of the course, focus on tensor network methods. Although in my course I briefly introduced also other successful numerical approaches (Monte Carlo, Hartree–Fock, density functional theory, etc.), they are not included here as many excellent books present them much better than what I could do here in a limited space. Next, I introduce elements of group theory and the consequences of symmetries in the correlated quantum world and on tensor networks. From the one hand, the appearance of different phases of matter, of spontaneous symmetry breaking and quantum phase transitions where correlations diverge, presents the highest difficulties for their numerical description. On the other hand, symmetries can be exploited to simplify the system description and to speed up the computation: symmetric tensor networks are fundamental to be able to perform state-of-the-art simulations. This is indeed true for global symmetries like the conservation of particle number in fermionic and bosonic systems or the magnetization in spin systems. However, recently it has been shown that also gauge symmetries can be embedded in the tensor network description, paving the way to very efficient simulation of lattice gauge theories, which are going to impact the modern research in condensed matter theory and high energy physics. Finally, the last part reviews the applications of the tools introduced here to the study of phases of quantum matter of their characterization by means of correlation functions and entanglement measures. Moreover, I present some results on out-of-equilibrium phenomena, such as adiabatic quantum computation, the Kibble–Zurek mechanism, and the application of quantum optimal control theory to many-body quantum dynamics.

In conclusion, this book can be used as a textbook for graduate computational quantum physics courses. Every chapter ends with exercises, most of which has been designed as weekly exercise for the students, to be evaluated in terms of programming, analysis and presentation of the results. At the end of the course, the students should be able to write a tensor network program to begin to explore the physics of many-body quantum systems. The course closed with each student choosing a final project to be performed in one month: the subject could also be related to other exams or laboratory activities, preparation for their master's thesis or as a part of their Ph.D. studies. Indeed, the scope of the course was to prepare them to these challenges, and in most cases, they demonstrated the success of such program.

The contents presented here are unavoidably based on my particular view of the field, and in particular, the focus is on the recent development of tensor network method I had the pleasure to participate in. Most of the contents, as acknowledge hereafter, are based on excellent books, review and research papers of collaborators and colleagues in the field. However, to the best of my knowledge, at the time I am

writing these lines, there is nowhere a self-contained book presenting all the elements needed to train students to work with tensor network method, despite the growing request of Ph.D. students with such skills. Finally, I think that this book could serve as a useful reference and guide to relevant literature for researchers working in the field, or as a starting point for colleagues that are entering it.

I would like to thank the many people that made this possible and that accompanied my life and scientific career until now. The colleagues at Ulm, J. Ankerhold, M. Freyberger, J. Hecker-Denschlag, S. Huelga, F. Jelezko, B. Nayedov, M. Plenio, F. Schmidt-Kaler, J. Stockburger and W. Schleich, made Ulm University a special and unique place to develop new scientific research. The support of the Center for Integrated Quantum Science and Technologies and of its director T. Calarco has been very precious in these last years. The scientific discussions with the other IQST fellows in Ulm and Stuttgart have been a very important stimulus.

The Institute for Quantum Optics and Quantum Information of the Austrian Academy of Science in Innsbruck played a unique role in my scientific life in the last years: especially since I became an IQOQI visiting fellow, I could fully profit from their hospitality and inspiration full environment. I am greatly thankful to Peter Zoller for his friendly guidance, his availability and the many enjoyable discussions.

A special thank to Giovanna Morigi that always supported me and to Saarland University that hosted my Heisenberg Fellowship. It has been a very fruitful and exciting time in a very pleasant environment: I am sure it will bring to a long-standing successful scientific collaboration.

The colleagues involved in the QuantERA project I have the pleasure to coordinate (QTFLAG) are working hard to develop the next generation of tensor network methods and experiments to investigate lattice gauge theories. I am sure that in the years to come they will inspire many of the readers of this book and myself with new exciting ideas and tools: many thanks to M. Dalmonte, I. Cirac, E. Rico, M. Lewenstein, F. Verstraete, U.-J. Wiese, L. Tagliacozzo, M.-C. Bañuls, K. Jansen, A. Celi, B. Reznik, R. Blatt, L. Fallani, J. Catani, C. Muschik, J. Zakrzewski, K. Van Acoleyen and M. Wingate.

This book could not have been written without the many persons that spent some years working with me, making my everyday life full of stimuli. I would like to thank P. Silvi, F. Tschirsich, D. Jaschke, M. Gerster, T. Felser, L. Kohn, Y. Sauer, H. Wegener, F. Schrodri, J. Zoller, M. Keck, T. Pichler, N. Rach, R. Said, J. Cui, V. Mukherjee, M. Müller, W. Weiss, A. Negretti, I. Brouzos, T. Caneva, D. Bleh, M. Murphy and P. Doria, for the uncountable hours of good physics.

In the last twenty years, I had the pleasure to collaborate with many other persons that, in different ways, have been present during my career development and showed me different interesting points of view: I am greatly thankful to G. Benenti, G. Casati, M. Rizzi, G. De Chiara, G. Santoro, D. Rossini, M. Palma, P. Falci, E. Paladino, F. Cataliotti, S. Pascazio, J. Prior, F. Illuminati, L. Carr, P. Zanardi, G. Pupillo, S. Lloyd, R. Onofrio, M. Lukin, L. Viola, J. Schmiedmayer, M. Tusk and to many others not included here. In addition to being always there, Saro Fazio also has the merit to be the first to point to me the original papers on the

density matrix renormalization group. From that discussion many years ago at Scuola Normale Superiore, a lot of exciting developments followed, and many others are still in sight.

Finally, it would not be possible for me to concentrate on my research without the full support of my family that every day gives me the strength of pursuing new challenges. Thank you, I owe you all. A huge thank to C. Montangelo for his help in improving the computer science part of the text, once more renovating the never-ending scientific confrontation between physicist and computer scientists, here enriched by a father–son debate. We did pretty well. With the advent of quantum technologies, I am sure there will be plenty of other occasions to keep on this enriching dispute between these two branches of science.

Despite the feedback I received on the draft from many colleagues, the remaining errors and typos in the text are my sole responsibility. I will appreciate any comment or feedback to improve the book in the future.

Padova, Italy

Simone Montangelo

Contents

1	Introduction	1
Part I The Single Body Problem		
2	Linear Algebra	9
2.1	System of Linear Equations	9
2.1.1	LU Reduction	11
2.2	Eigenvalue Problem	12
2.2.1	Power Methods	12
2.3	Tridiagonal Matrices	14
2.4	Lanczos Methods	15
2.5	Exercises	16
3	Numerical Calculus	19
3.1	Classical Quadrature	19
3.2	Gaussian Integration	22
3.3	Time-Independent Schrödinger Equation	24
3.3.1	Finite Difference Method	24
3.3.2	Variational Method	25
3.4	Time-Dependent Schrödinger Equation	27
3.4.1	Spectral Method	28
3.4.2	Split Operator Method	28
3.4.3	Partial Differential Equations Solvers	29
3.5	Exercises	32
Part II The Many-Body Problem		
4	Numerical Renormalization Group Methods	37
4.1	Mean Field Theory	37
4.1.1	Quantum Ising Model in Transverse Field	38
4.1.2	Cluster Mean Field	40

4.2	Real-Space Renormalization Group	41
4.3	Density Matrix Renormalization Group	43
4.4	Exercises	46
5	Tensor Network Methods	49
5.1	Tensor Definition	49
5.2	Tensor Manipulations	51
5.2.1	Index Fusion and Splitting	51
5.2.2	Compression	52
5.2.3	Tensor Network Differentiation	53
5.2.4	Gauging	54
5.2.5	Tensor Contraction Complexity	55
5.3	Ground States via Tensor Networks	55
5.3.1	Mean Field	56
5.3.2	Graphical Tensor Notation	58
5.3.3	Matrix Product States	60
5.3.4	Loop-Free Tensor Networks	64
5.3.5	Looped Tensor Networks	67
5.4	Time Evolution via Tensor Networks	69
5.4.1	Time-Dependent Density Matrix Renormalization Group	70
5.4.2	Fidelity-Driven Evolution	72
5.4.3	Time-Dependent Variational Principle	72
5.5	Measurements	74
5.6	Further Developments	76
5.7	Software	77
5.8	Exercises	77
6	Symmetric Tensor Networks	79
6.1	Elements of Group Theory	80
6.2	Global Pointlike Symmetries	81
6.3	Quantum Link Formulation of Gauge Symmetries	86
6.4	Lattice Gauge Invariant Tensor Networks	89
6.5	Exercises	93
Part III Applications		
7	Many-Body Quantum Systems at Equilibrium	97
7.1	Phase Transitions	97
7.2	Quantum-Classical Statistical Correspondence	100
7.3	Quantum Phase Transition	101
7.3.1	Critical Exponents	102
7.4	Entanglement Measures	105
7.4.1	Central Charge	106

- 7.4.2 Topological Entanglement Entropy 107
- 7.5 Exercises 108
- 8 Out-of-Equilibrium Processes 109**
 - 8.1 Adiabatic Quantum Computation 110
 - 8.1.1 Adiabatic Theorem 110
 - 8.1.2 Applications 112
 - 8.2 Kibble-Zurek Mechanism 114
 - 8.2.1 Crossover from Quantum to Classical Kibble-Zurek 115
 - 8.3 Optimal Control of Many-Body Quantum Systems 117
 - 8.4 Other Applications 122
 - 8.5 Exercises 123
- Appendix A: Hardware in a Nutshell for Physicists 125**
- Appendix B: Software in a Nutshell for Physicists 137**
- References 153**
- Index 171**

Symbols and Abbreviations

DMRG	Density matrix renormalization group
LGT	Lattice gauge theories
MF	Mean field
MPO	Matrix product operator
MPS	Matrix product state
RG	Renormalization group
SVD	Singular value decomposition
TN	Tensor networks
TTN	Tree tensor network
α_j	Index for local basis
β	Spontaneous magnetization
β_T	Inverse temperature $1/k_B T$
D	System dimensionality
d	Local Hilbert space dimension
η	Anomalous dimension
\mathcal{C}	Correlations
H	Hamiltonian
k_B	Boltzmann constant
λ	Matrix eigenvalue
m	Auxiliary bond dimension
N	Problem dimension or number of lattice sites
ν	Correlation length divergence exponent
$\mathcal{O}(\cdot)$	Order of approximation
ψ	Wave function
ρ	Density matrix
S	Entropy
\hat{S}	Spin operator
T	System temperature
z	Dynamical-scaling exponent

Chapter 1

Introduction



The many-body quantum problem is, in its different formulations, at the core of our understanding of nature and modern technologies [1]. On the one hand, it encompasses the modeling of the fundamental constituents of the universe [2] and quantum matter [3, 4]. On the other hand, the development of most future technology relies on our capability of describing and understanding many-body quantum systems: among others, the electronic structure problem with impact on chemistry and drugs research [5], the development of quantum technologies [6–8], and the engineering of new materials, e.g. more efficient light harvesting materials or displaying topological order [9–11].

The solution of the many-body quantum problem is thus an overarching goal of modern research in physics. Finding an approach to efficiently solve it would have an inestimable value and would impact directly or indirectly all fields in natural sciences – from medicine and chemistry to engineering and high-energy physics –, and it will pave the way to an Eldorado of technological developments that will enormously impact our everyday’s life. However, despite the enormous efforts spent and the impressive developments of analytical and numerical methods experienced in the last decades, the many-body quantum problem remains one of the most significant challenges we face.

The complexity of the many-body quantum problem immediately appears as the dimension of the space of possible configurations of N primary constituents of the system (number of quantum particles, lattice sites, etc.), increases exponentially with N . Indeed, if each elementary constituents is characterized by d possible different states $\{\alpha_i\}_{i=1}^d$ (spin configurations, electronic or molecular orbitals, harmonic oscillator or angular momentum levels, etc.), the number of possible configurations for the whole system is d^N . In quantum mechanics the fundamental object of interest is the wave function of the system, the amplitude of probability distribution $\psi(\alpha_1 \dots \alpha_N)$, whose modulus square gives the probability of the system to be in N -body configuration $\alpha_1 \dots \alpha_N$: that is, the object of interest is the vector containing the amplitude of probability for each possible system configuration. Typically, one has to update the

vector of probabilities to follow the system time evolution or to find the eigenvector corresponding to the ground state of the system. To date, only a small number of exactly solvable models have been identified [12] and exact diagonalization techniques [13] are restricted to few-body problems: thus the thermodynamical or the large N limit, which in many cases represents the interesting scenario, is often out of reach.

In conclusion, the exact treatment of the many-body quantum problem is almost always not available. However, many approximate analytical and numerical methods have been developed to attack it: a powerful class of methods is based on the semiclassical treatment of the quantum degrees of freedom [14], such as mean-field techniques that disregard or take into accounts only a small part of the quantum correlations of the system [15], e.g., Hartree-Fock methods and its generalizations [16]. It is well known, however, that these approaches typically perform optimally in high spatial dimensions, while they suffer in low dimensionality [17], especially in 1D where entanglement and quantum fluctuations play a dominant role. Another class of hugely successful techniques – mostly referred to as Monte Carlo methods – have been extensively used in the past decades to solve the many-body problem and in particular to study lattice gauge theories [18]. Monte Carlo methods capture the quantum and statistical content of the system by reconstructing the model partition functions by means of statistical samplings. However, Monte Carlo methods are severely limited whenever a negative or a complex action appears (sign problem) [19], e.g., while studying systems at finite chemical potential or out of equilibrium [20]. In many cases, careful, sophisticated choices of the statistical sampling algorithm can mitigate the sign problem. However, it has been proven that the sign problem is NP-hard, and thus a generic polynomial algorithm to study via Monte Carlo systems with complex action almost certainly does not exist [21].

An alternative approach to solve the many-body quantum problem is a class of numerical methods – at the center of this book – based on the renormalization group (RG) paradigm, that is, the idea of truncating the exponentially increasing Hilbert space based on energy considerations. The underlying hypothesis is that the low-energy physics of a system is mainly determined by the low-energy sectors of the system components [22]. In standard RG algorithms, the system size is then increased iteratively, and at each iteration, the high-energy sectors of the different parts of the system are discarded. This powerful idea has been further developed in the context of condensed matter systems where the density matrix renormalization group (DMRG) has been introduced [23]. Finally, it has been extensively further developed within the quantum information community with the development of tensor network (TN) methods [24–31]. As reported in details in this book, TN methods provide a recipe to describe efficiently many-body quantum systems with a controlled level of approximation as a function of the used resources, to optimally describe the relevant information in the system wave function [24]. Indeed, the TN approach is based on the fact that the tensor structure of the Hilbert space on which the wave function $\psi(\alpha_1\alpha_2\dots\alpha_N)$ is defined, introduces some structures. That is, typically ψ is not a fully general rank- N tensor and thus it might allow for an efficient representation. Moreover, in special but significant cases as the ground states of short-range interact-

ing Hamiltonians, one can show that the entanglement present in the system obeys some scaling laws – area laws of entanglement [32–37]. These scaling laws imply the possibility to introduce classes of states – TN ansätze, defined via their tensor structure – which can efficiently represent the system ground state [24, 25, 28, 30, 38–45]. Indeed, it is possible to decompose the most natural tensor structure defined by the rank- N tensor $\psi(\alpha_1\alpha_2\dots\alpha_N)$, in a network of lower rank tensors, which are equivalent unless approximations are introduced. Notice that while introducing a TN ansatz, one fixes the tensor structure, but not the dimension of the tensor indices nor the elements of each tensor: they are the variables to be optimized to build the most efficient representation of the system under study still capturing the system physical properties.

Starting from the reformulation of the DMRG algorithm in terms of the Matrix Product State (MPS) [46–48] – a one-dimensional tensor network ansatz – several classes of tensor networks, displaying various geometries and topologies, have been introduced over the years for different purposes, to include in the tensor structure the available physical information on the system or to allow a more efficient algorithmic optimization [24, 25, 49–54]. It is important to highlight that TN methods complement Monte Carlo methods as they do not suffer from the sign problem, allowing the simulation of out of equilibrium phenomena or at a finite chemical potential [55–77].

Indeed, a fascinating very recent field of application of tensor networks is lattice gauge theories [78, 79]. Since their introduction by Wilson [80], lattice gauge theories (LGTs) have found widespread application in the investigation of non-perturbative effects in gauge theories, with applications in different fields ranging from high-energy physics to condensed matter theory and quantum science [81–85]. LGTs are characterized by an extensive number of local symmetries and locally conserved quantities, as a consequence of the invariance of the theory under specific sets of transformations. The most common formulation of LGTs is the Lagrangian one, which is an ideal setup for Monte Carlo simulations [81–84]. Hereafter, we discuss instead the Hamiltonian formulation of LGT introduced by Kogut and Susskind [86, 87] as it is better suited for applications of tensor network methods and of quantum simulators [78, 79]. In particular, we focus on a compact formulation of the gauge fields, the so-called quantum link models [78, 88–90]. Quantum link models share many features with the Wilson formulation and converge to it in a particular limit [91], keep gauge invariance exact on the lattice, and are highly related to problems in frustrated quantum magnetism [3, 92].

Before concluding this introduction, it is also due mentioning that it exists another independent approach to solve the many-body problem, put forward by R. Feynman more than thirty years ago [93]. In modern words, he proposed to develop a dedicated quantum computer – a quantum simulator – to solve the quantum many-body problem, exploiting the fact that the exponential growth of the Hilbert space of the system under study is automatically matched by another many-body quantum system. Although hereafter we will not concentrate on this approach, we highlight that in the last decades quantum information science and the experimental setups have reached a level of control that allows different platforms (cold atoms, trapped ions, superconducting qubits, etc.) to perform quantum simulations of several many-body

phenomena [94, 95]. Paradigmatic classes of Hamiltonians of short-range interacting models such as the Ising and Heisenberg models for spin systems and the Hubbard or Bose-Hubbard model describing particles on a lattice have been quantum simulated [96–98], and also systems with long-range interactions have been engineered, exploiting Coulomb interaction of trapped ions and Rydberg atom simulators [99–104]. Different proposals for quantum simulations of lattice gauge theories have been put forward [77, 79, 99, 101, 105–118] and a first step in this direction has been very recently performed in a ion trap quantum simulator [119].

In conclusion, although tensor network methods and quantum simulators are relatively young and shall still reach full maturity, most probably future investigations on many-body quantum systems will be performed by means of three independent ways: either performing an experiment to directly interrogate the system of interest, or performing a simulation on a classical computer (either with Monte Carlo or tensor network methods), and finally performing a quantum simulation in a controlled environment. Given the complementarity of the approaches – not only in range of validity and efficiency but also in terms of complexity and resources needed for their realization, reproducibility, access to different direct or indirect information – most probably future research will be based on a combination of them, each exploiting its advantages and strengths and complementing and certifying the results of the others.

This book aims to introduce the reader to one of these pillars of future research – tensor network methods – from their basics to their application to lattice gauge theories, to bridge the gap between basics simulation methods for quantum physics and some of the most promising numerical methods that will most probably guide and support the research in fundamental quantum science and the development of quantum technologies.

The first part of the book reviews the most important tools available to solve numerically the single-body quantum problem, that will form the basis for the most advanced techniques presented in the second part of the book: Chap. 2 presents the linear algebra tools needed to diagonalize a Hamiltonian and some more advanced methods (Lancsoz) to diagonalize large matrices. In Chap. 3, we review the standard tools of numerical calculus, the numerical integration and the solution of partial differential equations – the Schrödinger equation – which can be used as standalone tools and will be part of tensor network algorithms. The second part presents the core of the book, the tensor network methods: We start in Chap. 4 reviewing the numerical real-space renormalization group and its generalization, the DMRG. In Chap. 5 we introduce the tools and methods to perform numerical studies (both at and out-of-equilibrium) of one-dimensional systems of hundreds of components, mainly based on Matrix Product States. Chapter 6 introduces more advanced concepts, in particular, the exploitation of symmetries to enhance the performances of the numerical simulations and to describe lattice gauge theories in the quantum link formulation. Finally, the last part of the book presents some applications of tensor network methods to physical situations: in Chap. 7, we report TN methods applications to the study of quantum phases of matter and the different quantities that can be used and computed efficiently to characterize them: correlation functions and entanglement measures. The last chapter presents some of the most successful

applications of TN methods to out-of-equilibrium phenomena: adiabatic quantum computation, the Kibble-Zurek mechanism, and the optimal control of many-body quantum processes. The appendices present some basic concepts that any physicist approaching the computational methods should know: the basics of classical computer hardware and architectures and good fundamental practices to produce reliable and useful software.

Part I
The Single Body Problem

Chapter 2

Linear Algebra



In this chapter, we attack one of the simplest but ubiquitous problems that a computational physicist typically faces, the solution of an eigenvalues problem. In quantum physics, the eigenvalues problem usually arise to solve a given Hamiltonian describing a single- or many-body quantum systems. Indeed, apart from a few classes of problems for which analytical solutions are known (from the particle in the box to the Bethe Ansatz solutions), the search for the ground and excited states of a quantum system shall be performed numerically. The numerical solution of the eigenvalues problem equals to the diagonalization of a matrix representing the Hamiltonian in a suitable basis, independently from the fact that the original system is discrete, e.g. a collection of qubits or atomic levels, or continuous, e.g. atoms or ions in free space or traps, or Bose-Einstein condensates. The latter scenarios shall be recast into discrete system either via space discretization (see Appendix B) or via second quantization. Finally, a plethora of problems in math, physics, and computer science can be recast in eigenvalues problems, thus hereafter we first introduce the problem and then present, in order of complexity, the state-of-art approaches for global and partial eigensolvers. As we will see, the eigensolver strategies are based on linear algebra operations that, for the sake of completeness, we briefly recall hereafter, also to set the notation: we will use throughout the book standard quantum mechanics notation and assume that the variables are complex. We base the quick overview presented here and in the next chapter (and appendixes) on many different excellent books and reviews, such as, e.g. [120–129].

2.1 System of Linear Equations

A system of linear equations can be represented in operator form

$$\hat{O}|x\rangle = |y\rangle; \tag{2.1}$$

where \hat{O} is a linear operator, and the vectors $|x\rangle$, $|y\rangle$ live in a Hilbert space $\mathcal{H} \equiv \mathbb{C}^N$. Given \hat{O} and $|y\rangle$, one can use the properties of equivalency of systems of linear equations to solve for the vector $|x\rangle$ which lead to the *Gaussian elimination*: (i) the multiplication of lines of the system by a factor $f_{i,j}$, and (ii) the replacement of one line with the line itself added term by term with another one does not change the system result. Thus, any system of linear equations of the form of (2.1) can be recast in an equivalent (with the same solution) system where the operator has an upper triangular form

$$\hat{U}|x\rangle = |y'\rangle; \quad \text{where} \quad \hat{U} = \begin{pmatrix} U_{11} & U_{12} & U_{13} & \dots & U_{1N} \\ 0 & U_{22} & U_{23} & \dots & U_{2N} \\ 0 & 0 & U_{33} & \dots & U_{3N} \\ & & & \ddots & \vdots \\ 0 & 0 & 0 & 0 & U_{NN} \end{pmatrix}. \quad (2.2)$$

Given the operator \hat{U} , the solution of the system of linear equations is easily obtained via back substitution as follows:

$$x_k = \left(y_k - \sum_{j=k+1}^N U_{kj}x_j \right) / U_{kk}. \quad (2.3)$$

The previous lines not only describe the Gaussian elimination method, but provide the description of the algorithm to solve any system of N linear equations that can be straightforwardly coded. A fundamental property of algorithms is their algorithmic complexity, defined as the number of operations needed to solve the problem as a function of the input size, here the number of equations N (see Appendix B for more on software complexity and an introduction to good software development). A simple estimate results in a total number of operations that scales as $\mathcal{O}(N^3)$: the reduction of each of the $N(N-1)/2$ elements of the lower part of the matrix in (2.2) to zero requires $\mathcal{O}(N)$ operations, that is, a total of $\mathcal{O}(N^3)$. The back-substitution described in (2.3) is of lower order, as it requires $\mathcal{O}(N^2)$ operations.

Notice that the algorithm for solving a system of linear equation provides also a method to compute the inverse of a matrix: indeed by definition the inverse of the matrix A is such that $\hat{O}^{-1}\hat{O} = \hat{O}\hat{O}^{-1} = \mathbb{1}$, then, we can identify

$$\hat{O}^{-1} \equiv \begin{pmatrix} | & | & \dots & | & | \\ x^1 & x^2 & \dots & x^{N-1} & x^N \\ | & | & \dots & | & | \end{pmatrix}, \quad \mathbb{1} \equiv \begin{pmatrix} | & | & \dots & | & | \\ y^1 & y^2 & \dots & y^{N-1} & y^N \\ | & | & \dots & | & | \end{pmatrix} \quad (2.4)$$

where $|x^j\rangle$ and $|y^j\rangle$ forms the column vectors of the two matrices respectively. In conclusion, the condition to find the inverse of the matrix becomes simply a set of linear systems of equations:

$$\hat{O}|x^j\rangle = |y^j\rangle; \quad (2.5)$$

which can be solved with the elementary construction of the augmented rectangular matrix, formed by the operator \hat{O} and the identity on its side. Gaussian eliminating the off-diagonal terms of the operator \hat{O} results in the computation of the operator \hat{O}^{-1} .

2.1.1 LU Reduction

The Gaussian elimination reviewed in the previous section is only one of a family of reduction methods to triangular matrices. Another reduction that plays a pivotal role in the numerical computation of ground state properties of quantum systems is the LU decomposition [122]. Indeed, one can show straightforwardly that any operator can be written as a left and right triangular operators

$$\hat{O} = \hat{L}\hat{U} \quad \text{such that} \quad \hat{L} = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ f_{2,1} & 1 & \dots & 0 & 0 \\ f_{3,1} & f_{3,2} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ f_{N,1} & f_{N,2} & \dots & f_{N,N-1} & 1 \end{pmatrix}, \quad (2.6)$$

where the coefficients $f_{i,j}$ are those used perform the gaussian elimination of the operator \hat{O} into the matrix \hat{U} given in (2.2).

As we will see later in the chapter, the usefulness of such reduction becomes clear whenever one needs to solve many different systems of linear equations which differs only by the vector $|y\rangle$. Indeed, in such a case, the original problem $\hat{O}|x\rangle = |y\rangle$ can be recast in two systems of linear equations

$$\begin{cases} \hat{L}|z\rangle = |y\rangle \\ \hat{U}|x\rangle = |z\rangle \end{cases}, \quad (2.7)$$

whose solution only requires $\mathcal{O}(N^2)$ operations as only the back substitution is needed, being both operators already triangular ones. Thus, the algorithm to solve different systems of linear equations with the same \hat{O} and different $|y\rangle$, is as follows:

1. LU-reduce the operator \hat{O} (scales as $\mathcal{O}(N^3)$);
2. Perform two back substitution S, as in (2.3), (scales as $\mathcal{O}(N^2)$).

This approach has no drawback, as the first step scales as the full Gaussian elimination. Thus, it does not introduce any overhead in computational time nor memory as the two triangular matrices can be stored with the same amount of memory as the original one. Moreover, if the number of systems of equations to be solved M (with the same \hat{O} operator and different $|y\rangle$) is such that $M \gg N$, it is much more convenient as $M \cdot \mathcal{O}(N^3) \gg 1 \cdot \mathcal{O}(N^3) + M \cdot \mathcal{O}(N^2)$.

2.2 Eigenvalue Problem

In this section, we introduce the most common methods to find the eigenvalues of an operator \hat{O} acting on the Hilbert space $\mathcal{H} \equiv \mathbb{C}^N$. Hereafter, we specialize for the case of Hermitian operators for the obvious prominent role they play in quantum mechanical problems. However, what follows can be generalized straightforwardly for more general cases. For example, the manipulation of non-Hermitian matrices naturally appears when studying open quantum systems.

We aim to solve the eigenvalue problem defined as

$$\hat{O}|x_j\rangle = \lambda_j|x_j\rangle \quad j = 1, \dots, N, \quad (2.8)$$

where hereafter we consider orthonormal eigenstates $\langle x_j|x_k\rangle = \delta_{j,k}$ and ordered (real) non degenerate eigenvalues $|\lambda_N| > |\lambda_{N-1}| > \dots > |\lambda_1|$.

2.2.1 Power Methods

The set of eigenstates $|x_j\rangle$ form a complete basis for the Hilbert space \mathcal{H} , thus any given (random) state $|y\rangle \in \mathcal{H}$ can be written as $|y\rangle = \sum y_j|x_j\rangle$, where $y_j = \langle x_j|y\rangle$. Thus, the application of the operator \hat{O} to an initial random state $|y\rangle$ results into

$$\hat{O}|y\rangle = \sum y_j\lambda_j|x_j\rangle. \quad (2.9)$$

That is, after k applications of the operator \hat{O} one obtains the vector state

$$\hat{O}^k|y\rangle = \sum y_j\lambda_j^k|x_j\rangle. \quad (2.10)$$

The net result of the application of the operator to the initial state is then to change the relative projections onto the eigenvectors, according to the ratio between the eigenvalues, that is, to rotate the state towards the eigenstate of maximal eigenvalue: the sequence of states for $k \rightarrow \infty$ tends to:

$$|y^k\rangle = \hat{O}^k|y\rangle \rightarrow |x_N\rangle, \quad (2.11)$$

and correspondingly the ratio $||y^k\rangle|/||y^{k-1}\rangle| \rightarrow \lambda_N^k/\lambda_N^{k-1} = \lambda_N$. In conclusion, one can find the most significant eigenstate and eigenvector with multiple applications of the operator to an initial random vector, and computing the ratio between two subsequent vector norms. We will show later how to generalize this procedure to find all the rest of the spectrum and the remaining eigenvalues.

This procedure is working properly, can be implemented easily, and works in most scenarios. However, its efficiency is drastically reduced whenever the system has

almost degenerate eigenvalues (which turns out to be a class of fundamental physical processes, e.g., as we will see in Chap. 7 systems that display critical behavior at the thermodynamical limit). Indeed, the speed of convergence of the sequence in (2.11) depends on the ratio λ_N/λ_{N-1} , which, if small, can practically prevent the algorithm to converge. To cure for this problem, and also speed up the algorithm convergence, the *inverse power method* has been introduced. The main idea stems from the fact that (2.8) is equivalent to

$$\hat{O}^{-1}|x_j\rangle = \lambda_j^{-1}|x_j\rangle, \quad (2.12)$$

thus, applying \hat{O}^{-1} one can find the smallest eigenvalue and eigenvector of \hat{O} . As before, the speed of convergence to the final result is given by the ratio λ_2/λ_1 . Apparently, there is no big difference between the two methods apart from the fact that they target the opposite extreme of the spectrum. However, the important difference is that in the latter the extreme eigenvalue is at the denominator of the speed of convergence ratio: this in many scenarios allows us to enhance the speed of convergence drastically. Indeed, in many cases, it is possible to estimate the lowest eigenvector of an operator, e.g., with some perturbative argument. Another typical scenario is, when looking for the ground state energy of a Hamiltonian as a function of a parameter g , allows exploiting the previously computed result for $g - \Delta g$. In conclusion, one can redefine the offset (e.g., setting a different zero value for the energy) of the eigenvalue problem as

$$(\hat{O} - q\mathbb{1})^{-1}|x_j\rangle = (\lambda_j - q)^{-1}|x_j\rangle, \quad (2.13)$$

where q is the estimate of the smallest eigenvalue λ_1 . The speed of convergence of the sequence

$$|y^k\rangle = (\hat{O} - q\mathbb{1})^{-1}|y^{k-1}\rangle \quad (2.14)$$

is now given by $(\lambda_2 - q)/(\lambda_1 - q)$ which can be highly enhanced if $\lambda_1 \sim q$. The parameter q is often referred as the *guess* value in the diagonalization subroutine and providing a good guess can make the difference between standard and professional results.

The careful reader has surely spotted that in the presentation above a crucial point has been not carefully addressed: indeed to compute (2.14), the operator $(\hat{O} - q\mathbb{1})^{-1}$ is needed. As we have seen in the previous section, this step can be done straightforwardly, however, once again there are scenarios where it is preferable to avoid it. In particular, writing the matrix representation of the operator \hat{O} shall be avoided if possible as it is highly inefficient. In particular, whenever the operator has a structure or representation by a sparse matrix, the implementation of a code computing the action of the operator on the state is more efficient than calculating the matrix representation and then performing the matrix-vector multiplication. In standard subroutines these subroutines are often called *amul* and are typically the only part of code left to be written [130]. Indeed, storing a matrix requires the square of the memory than storing a vector, reaching the memory limitation much faster: this

is particularly true for many-body quantum systems where the Hamiltonians contain typically only few-body interactions, and they are sparse matrices with well-defined tensor structure. Therefore, if we want to avoid to compute the inverse of the operator explicitly, we can proceed to rewrite (2.14) as

$$(\hat{O} - q\mathbb{1})|y^{k+1}\rangle = |y^k\rangle \quad (2.15)$$

and identify it as a system of linear equations: given the state at the k -th interaction, one shall find the state at the next one. To compute the whole series, we shall solve the same system of linear equation for different state vectors $|y_k\rangle$, which is exactly the scenario studied in the previous section, where we introduced the LU decomposition. Thus, one first shall compute the LU decomposition and then perform a Gaussian elimination for each iteration of the sequence in (2.15). Notice that despite the fact that both LU decomposition and the inversion scales as $\mathcal{O}(N^3)$, the former has a prefactor of one third with respect to the second one, thus it is still more convenient. Moreover, LU decomposition preserves sparseness of the matrix while the inversion does not allow exploiting the problem structure for efficiency.

Finally, once the first eigenvector and its correspondent eigenvalue has been computed, the procedure can be repeated to compute the next ones, provided that we eliminate the components periodically along $|x_0\rangle$: that is, after the initial random state is created, one shall redefine it as

$$|y_0'\rangle = |y_0\rangle - \langle x_0|y_0\rangle|x_0\rangle; \quad (2.16)$$

such that $|y_0'\rangle \perp |x_0\rangle$. The sequence in (2.15) will converge to the next smallest eigenvalue contained in the expansion $|y\rangle = \sum y_j|y_j\rangle$. Notice that this orthogonalization shall be repeated during the sequence as numerical errors might reintroduce components along $|x_0\rangle$ which would grow and prevent the algorithm from working properly.

2.3 Tridiagonal Matrices

Tridiagonal matrices, defined as the class of matrices that have elements different from zero only on the diagonal and the upper- and lower-diagonals, play a prominent role in physics as they naturally appear in wide class of problems such as the single body quantum problem (see Chap. 3) and in tight binding models. Moreover, as we will see later on in Sect. 2.4, most matrices can be written in such a form via a change of basis that can be efficiently computed. Finally, as we recall briefly hereafter, it is highly efficient to solve the eigenvalue problem for tridiagonal matrices. Indeed, given a tridiagonal matrix of the form

$$T_N = \begin{pmatrix} d_1 & b_1 & 0 & 0 & \dots & 0 & 0 \\ d_1 & d_2 & b_2 & 0 & \dots & 0 & 0 \\ 0 & b_2 & d_3 & 0 & \dots & 0 & 0 \\ & & & \ddots & & & \\ 0 & 0 & 0 & 0 & \dots & b_{N-1} & b_{N-1} \\ 0 & 0 & 0 & 0 & \dots & b_{N-1} & d_N \end{pmatrix} \quad (2.17)$$

the characteristic polynomial obeys the recurrence equation

$$\det(T_N - \lambda \mathbb{1}) = (d_N - \lambda) \det(T_{N-1} - \lambda \mathbb{1}) - d_{N-1}^2 \det(T_{N-2} - \lambda \mathbb{1}). \quad (2.18)$$

Thus, we can define the one-dimensional function $f_N(\lambda) = (d_N - \lambda)f_{N-1}(\lambda) - d_{N-1}^2 f_{N-2}(\lambda)$ with $f_0(\lambda) = 1$ and use very efficient standard methods to find its roots, that is, the eigenvalues of the original matrix in (2.2) [122].

2.4 Lanczos Methods

The Lanczos diagonalization algorithm aims to find the transformation from a general matrix to a tridiagonal form, so that the eigenproblem can be solved efficiently [131]. Moreover, it allows one to operate without writing explicitly the whole matrix, thus enormously enhancing its field of applicability. As for the power methods, it starts with a random vector in the relevant Hilbert space $|y_1\rangle$ and begins applying the operator to be diagonalized to it $|y_2\rangle = \hat{O}|y_1\rangle$. It then decomposes the new state $|y_2\rangle$ in its normalized perpendicular and parallel components with respect to the original state $|y_1\rangle$, i.e.

$$\hat{O}|y_1\rangle = d_1|y_1\rangle + b_1|y_2\rangle, \quad (2.19)$$

where by construction $d_1 = \langle y_1|\hat{O}|y_1\rangle$, $b_1|y_2\rangle \equiv \hat{O}|y_1\rangle - d_1|y_1\rangle$, and the coefficient b_1 can be found imposing $|\langle b_1 y_2|b_1 y_2\rangle|^2 = 1$. Obtained the first two vectors, the operation is iterated another time, with the additional care of decomposing the new state in the components orthogonal and parallel with respect to both previous states. We then obtain

$$b_2|y_3\rangle = \hat{O}|y_2\rangle - b_1|y_1\rangle - d_2|y_2\rangle \quad (2.20)$$

where, as before, $b_1 = \langle y_1|\hat{O}|y_2\rangle$, $d_2 = \langle y_2|\hat{O}|y_2\rangle$ and the coefficient b_2 can be calculated imposing the normalization of $b_2|y_3\rangle$.

After the first two iterations of the algorithm, we obtained the values of the coefficients d_1 , d_2 , b_1 , and b_2 as well as the states $|y_1\rangle$, $|y_2\rangle$, and $|y_3\rangle$. The main point of the algorithm appears in its beauty at the third iteration: applying once more the operator \hat{O} to the state $|y_3\rangle$ and decomposing it to the parallel and perpendicular components with respect to all previous computed states, we obtain

$$\hat{O}|y_3\rangle = r|y_1\rangle + b_2|y_2\rangle + a_3|y_3\rangle + b_3|y_4\rangle. \quad (2.21)$$

The first coefficient is defined as before as $r = \langle y_1 | \hat{O} y_3 \rangle$, which is equal to zero as $\langle y_1 | \hat{O} y_3 \rangle = \langle \hat{O} y_1 | y_3 \rangle = (d_1 \langle y_1 | + b_1^* \langle y_2 |) | y_3 \rangle = 0$, as the operator \hat{O} is Hermitian, and the orthogonality relation $|y_3\rangle \perp |y_2\rangle, |y_1\rangle$ holds by construction. In general, one can show that for any successive iteration, it holds

$$\hat{O}|y_k\rangle = b_{k-1}|y_{k-1}\rangle + a_k|y_k\rangle + b_{k+1}|y_{k+1}\rangle. \quad (2.22)$$

In conclusion, we define the matrix composed by the first N vectors $|y_k\rangle$ as

$$Y = \begin{pmatrix} | & | & & | \\ |y_1\rangle & |y_2\rangle & \dots & |y_N\rangle \\ | & | & & | \end{pmatrix}, \quad (2.23)$$

and the relation of (2.22) implies that $T_N = Y^\dagger \hat{O} Y$ where T_N is a tridiagonal matrix as defined in (2.17).

Notice that to compute the coefficients of the tridiagonal matrix T_N , one does not need to write explicitly the matrix Y , nor an explicit full matrix representation of the original operator \hat{O} if – as it is often the case – the matrix is sparse or has some structure to be exploited. Indeed, the coefficients in T_N can be obtained by means of (2.22): at every steps one need to store the vectors $|y_{k-1}\rangle, |y_k\rangle, |y_{k+1}\rangle$ and eventually two vectors of coefficients a_k and b_k . Once these coefficients are calculated, the approaches reviewed in Sect. 2.3 can be applied to find the eigenvalues and eigenvectors of the operator \hat{O} in its tridiagonal form T_N . Thus, with the Lancosz approach, huge matrices can be diagonalized, even those that cannot be stored in memory. We refer the interested reader to the extensive literature on Lancosz methods and the Krylov subspaces they are based on, see, e.g., [132].

2.5 Exercises

1. Consider a random Hermitian matrix A of size N .
 - a. Write a program that initializes A and performs an LU reduction. How does it scales with N ?
 - b. Diagonalize A and store the N eigenvalues λ_i in increasing order.
 - c. Compute the normalized spacings between eigenvalues $s_i = \Delta\lambda_i / \bar{\Delta\lambda}$ where

$$\Delta\lambda_i = \lambda_{i+1} - \lambda_i,$$

and $\bar{\Delta\lambda}$ is the average $\Delta\lambda_i$.

- d. Compute the average spacing $\bar{\Delta\lambda}$ over a different number of levels around λ_i (i.e. $N/100, N/50, N/10 \dots N$) and compare the results of next exercise for the different choices.

2. Study $P(s)$, the distribution of the s_i defined in the previous exercise, accumulating values of s_i from different random matrices of size.
- Compute $P(s)$ for a random hermitian matrix.
 - Compute $P(s)$ for a diagonal matrix with random real entries.
 - Fit the corresponding distributions with the function:

$$P(s) = as^\alpha \exp(-bs^\beta)$$

comparing them with the predictions of Random Matrix Theory [133].

3. Consider the Ising Hamiltonian defined in (4.4).
- Write a program that computes the $2^N \times 2^N$ matrix H for different N .
 - Diagonalize H for different $N = 1, \dots, N$ and $\lambda \in [0 : 3]$.
 - Plot the first k levels as a function of λ for different N . Comment the spectrum.

Chapter 3

Numerical Calculus



In this chapter, we review the standard numerical methods to perform differential and integral calculus using finite precision mathematics. Indeed, the price to pay to invoke the powerful help of numerical methods is to introduce a finite precision cutoff which shall be taken carefully into account as it enters as a numerical error. Nevertheless, whenever this error is kept under control and correctly bounded, e.g., to machine precision (see Appendix A), we say that the final result is numerically exact. Hereafter, we will review the most successful numerical approaches to solve integrals of functions and partial differential equations, and specify them to solve the Schrödinger equation for few-body systems, that is, whenever it is possible to write explicitly the system wave function and the operators acting on them. These methods form the basis for the numerical approaches based on tensor network methods that we will introduce in the next parts of the book.

3.1 Classical Quadrature

The most intuitive approach to compute numerically the integral of a one dimensional real function $f(x)$ follows straightforwardly the Riemann integration, that is, to define a grid $\{x_i\}_{i=1}^N$ in the support space of the function f and to approximate the function f with $N - 1$ trapezoids as

$$\int_a^b f(x)dx \simeq \frac{1}{2} \sum_{i=1}^{N-1} [f(x_i) + f(x_{i+1})](x_i - x_{i+1}). \tag{3.1}$$

If we assume the lattice spacing to be equidistant, $h = x_i - x_{i+1}$, we obtain the *trapezoidal rule*

$$\int_a^b f(x)dx \simeq h \left(\frac{1}{2}f_1 + f_2 + f_3 + \cdots + f_{N-2} + f_{N-1} + \frac{1}{2}f_N \right) \equiv h \sum_{i=1}^{N-1} w_i f_i \quad (3.2)$$

where $f(x_i) \equiv f_i$ and $w_i = 1, 1/2$. Thus, the integral following the trapezoidal rule is given as the sum of the function values f_i evaluated at points x_i , weighted with coefficients w_i ; before multiplying the result by h . By definition the result is exact in the limit of the lattice spacing h going to zero. However, it is fundamental to know the error introduced for every finite h . The order of the error can be evaluated by means of a Taylor expansion: the error is introduced by the approximation of each area of bases h under the function f with a trapezoid, while there is no additional error in the sum of each of them. It is then sufficient to compute the error for a single trapezoid, that without loss of generality we assume to be centered in zero. We then evaluate the area of the trapezoid with two functions evaluations, at $-h/2$ and $h/2$ and compare the result with the analytical integral obtained via Taylor expansion around zero, that is

$$\int_{-h/2}^{h/2} f(x)dx \simeq \int_{-h/2}^{h/2} f(0) + hf'(0)x + f''(0)\frac{x^2}{2} + \cdots dx = hf(0) + \frac{h^3}{3}f''(0) \dots \quad (3.3)$$

The value of $f(0)$ can be expressed as a function of the two extremal points $f(-h/2)$ and $f(h/2)$ again Taylor expanding them and summing the two expression: $f(0) = \frac{1}{2}(f(-h/2) + f(h/2)) - h^2/8f''(0) + \cdots$. Finally, one obtains the dependence of the value for the integral as a function of the extremal points

$$\int_{-h/2}^{h/2} f(x)dx \simeq \frac{h}{2}[(f(-h/2) + f(h/2)) + \frac{5h^2}{12}f''(0) + \cdots], \quad (3.4)$$

which compared with (3.2) shows that the trapezoidal rule is correct up to second order in h , as one disregard the term $\frac{5h^2}{12}f''(0)$ and higher orders ones. This is a not surprising result at all, as approximating the integral as a sum of trapezoids is equivalent to approximate linearly the function $f(x)$ within each interval.

It is clearly possible to improve the accuracy of the integration algorithm simply keeping higher orders in the Taylor expansion of $f(x)$. Repeating straightforwardly the previous steps and considering three function evaluation points, which are necessary to approximate at second order a function, one obtains

$$\int_{-h}^h f(x)dx \simeq \frac{1}{3}(f(-h) + 4f(0) + f(h)), \quad (3.5)$$

that is the *Simpson's rule* which prescribes $w_1 = w_N = 1/3$, $w_i = 4/3$ and $w_i = 2/3$ for even and odd terms respectively. The precision of such prescription can be double checked as before: the higher order expansion reads

$$\int_{-h}^h f(x) dx \simeq 2hf(0) + \frac{h^3}{3} f'(0) + \frac{h^3}{3} f''(0) + \dots \quad (3.6)$$

and inserting the relation $f''(0) = (f(h) - 2f(0) + f(-h))/h^2 + \mathcal{O}(h^2)$ one obtains the Simpson's rules neglecting $\mathcal{O}(h^4)$ terms. Notice that the approximation should be in principle at second order, however, the symmetry of the problem grants an additional order of precision.

This way of proceeding can be extended to higher orders. However, it quickly becomes quite lengthy and intricate. It is more convenient to reformulate the problem in terms of a system of linear equations, which can be solved with standard tools. Indeed, one can impose directly the condition introduced in (3.2), that is, the equivalence between the integral of the function and the sum of N weighted functions evaluations $w_i f_i$ up to some order $M + 1 \leq N$. In conclusion, one can impose the following conditions for functions of up to order $f(x) = x^M$

$$\int_a^b x^0 dx = b - a \equiv \sum w_i \quad (3.7)$$

$$\int_a^b x^1 dx = (b^2 - a^2)/2 \equiv \sum w_i x_i \quad (3.8)$$

$$\int_a^b x^2 dx = (b^3 - a^3)/3 \equiv \sum w_i x_i^2 \quad (3.9)$$

⋮

$$\int_a^b x^M dx = (b^{M+1} - a^{M+1})/(M + 1) \equiv \sum w_i x_i^M. \quad (3.10)$$

It can be easily checked that the first two equations give rise to a system of linear equations for the coefficients w_i whose solution (under the assumption $a = -h/2$ and $b = h/2$) is the trapezoidal rule. Similarly, the first three equations return Simpson's rule. This approach also highlights that the full potential of this numerical approximation is not fully exploited. Indeed, setting a priori the lattice spacing h is not convenient: to increase the numerical integration precision, we could include the positions x_i as variables. Thus, (3.9) can be solved not only for the weights w_i but also for the estimation points x_i : having doubled the number of variables, it is

possible to solve more equations with the same number of function evaluations. For example, imposing $x_1 = -x_2$ (but not $x_1 = -h/2$) and $|a| = |b|$, the system of four equations

$$\begin{cases} b - a = w_1 + w_2 \\ 0 = w_1 x_1 + w_2 x_2 \\ (b^3 - a^3)/3 = w_1 x_1^2 + w_2 x_2^2 \\ 0 = w_1 x_1^3 + w_2 x_2^3 \end{cases} \quad (3.11)$$

can be solved obtaining $w_1 = w_2 = (b - a)/2$ and $x_1 = -x_2 = \sqrt{(b^3 - a^3)/(3(b - a))}$. Thus, just changing the positions of the points sampling the function, the approximation can be improved from $\mathcal{O}(h^2)$ to $\mathcal{O}(h^4)$. The generalization of such procedure to any order M goes under the name of Gaussian integration as we will review in the next section.

3.2 Gaussian Integration

In the previous section, we have seen that it is possible to write and solve a system of N linear equations for the weights w_i of the N function sampling points f_i . In the case of $N = 2$, we also showed that by properly choosing the sampling points – thus abandon a standard equidistant lattice – it is possible to double the precision of the integral without increasing the computational cost. Hereafter, we will show that this is indeed always possible thanks to the properties of polynomials basis functions such as Legendre polynomials. Indeed, by sampling the function in N points and properly choosing the weights w_i , we have shown that it is possible to solve the first N (3.7)–(3.10). In general, the following equations define a set of polynomials Z_j of the variables x_i

$$Z_0(x_1, x_2, \dots, x_N) \equiv \sum w_i - \int_a^b x^0 dx = 0 \quad (3.12)$$

$$Z_1(x_1, x_2, \dots, x_N) \equiv \sum w_i x_i - \int_a^b x^0 x^1 dx = 0 \quad (3.13)$$

⋮

$$Z_{N-1}(x_1, x_2, \dots, x_N) \equiv \sum w_i x_i^{N-1} - \int_a^b x^{N-1} dx = 0 \quad (3.14)$$

$$Z_N(x_1, x_2, \dots, x_N) \equiv \sum w_i x_i^N - \int_a^b x^N dx \neq 0 \quad (3.15)$$

$$Z_{N+1}(x_1, x_2, \dots, x_N) \equiv \sum w_i x_i^{N+1} - \int_a^b x^{N+1} dx \neq \quad (3.16)$$

$$\vdots$$

$$Z_{N+K}(x_1, x_2, \dots, x_N) \equiv \sum w_i x_i^{N+K} - \int_a^b x^{N+K} dx \neq 0 \quad (3.17)$$

for which, in general, $Z_j(x_1, x_2, \dots, x_N) \neq 0 \forall j > N - 1$ and $Z_j(x_1, x_2, \dots, x_N) = 0 \forall j \leq N - 1$ for a proper choice of the weights w_i . Our task is then to find the set of positions $\{x_1, x_2, \dots, x_N\}$ that set to zero the maximal number of polynomials Z_j , effectively increasing the order of the approximation of the numerical integration. From now on, we rescale the integration bounds to $a = -1$ and $b = 1$ and we will exploit the Legendre polynomials $L_n(x) \equiv \sum_{m=0}^n p_{n,m} x^m$ and their properties, in particular,

1. L_n is a polynomial of order n with n zeros x_i such that $L_n(x_i) = 0$, with $L_0 \equiv 1$.
2. L_n forms an orthogonal basis for the space of functions in $[-1 : 1]$: any other polynomial can be expanded in this basis and in particular $x^m = \sum_{n=0}^m q_{m,n} L_n(x)$, where $q_{m,n}$ is the matrix inverse of $p_{n,m}$.
3. L_n is orthogonal to x^m with $m < n$ as $\langle x^m | L_n(x) \rangle = \sum_{j=0}^m q_{m,j} \langle L_j(x) | L_n(x) \rangle = 0 \forall j < n$.

We are now ready to look for a solution of the problem we have introduced at the beginning of this section: to find a set of $\{\bar{x}_i\}$ such that, for example, $Z_N(\bar{x}_i) = 0$. To do so, we multiply each of the first $N + 1$ (3.12)–(3.15) for the coefficient $p_{n,j}$ and sum them up, obtaining the relation

$$p_{n,n} Z_N(x_1, x_2, \dots, x_N) = \sum_j \left[p_{n,j} \sum w_i x_i^j - \int_{-1}^1 p_{n,j} x^j dx \right] \quad (3.18)$$

$$= \sum_i w_i L_n(x_i) - \int_{-1}^1 L_n(x) dx. \quad (3.19)$$

The second term of (3.19) is identically zero as $\int_{-1}^1 L_n(x) dx = \langle L_0(x) | L_n(x) \rangle = 0$. Finally, the polynomial $L_n(x)$ has exactly N zeros, thus if one choose them as $\{\bar{x}_i\}$, one obtains $Z_N(x_1, x_2, \dots, x_N) = 0$. In a similar way, one can show that also the next Z_{N+j} polynomials are identically zero for every $j \leq N - 1$. Indeed, to prove that Z_{N+j} is zero, it is sufficient to multiply the equations for Z_j to Z_{N+j} for the same coefficients as before to obtain the general relation

$$p_{n,n} Z_{N+j}(x_1, x_2, \dots, x_N) = \sum_i w_i x_i^j L_n(x_i) - \int_{-1}^1 x^j L_n(x) dx. \quad (3.20)$$

As before, the first term on the right hand side is identically zero for the same choice of x_i , while the second term is equal to $\langle L_j(x)|L_n(x)\rangle$ which is zero for any $j < n$. In conclusion, with $2N$ free parameters (the weights and the sampling positions) we have solved $2N$ equations granting a precision of the numerical integration of $\mathcal{O}(h^{2N})$.

3.3 Time-Independent Schrödinger Equation

In the rest of the chapter, we review the most common and well-established methods to solve partial differential equations numerically. In particular, we specialize our presentation on the exact numerical solution of both time-independent and time-dependent Schrödinger equation.

The fundamental problem of quantum mechanics is to find the eigenvalues and eigenvectors of the Hamiltonian describing the system. Once the Hamiltonian is diagonalized, all information about the system can be computed. A general approach to achieving such result is to write the Hamiltonian matrix representation and solve it with the numerical methods presented in Chap. 2. Hereafter, we present the standard approaches to recast a quantum mechanical problem defined in a continuous space into a discretized version, essential to write the Hamiltonian in a discrete and finite matrix representation. The main approaches are indeed of two classes: the discretization of the space, introducing a lattice as reviewed in Sect. 3.3.1 and the introduction of proper basis functions used to expand the Hamiltonian, as shown in Sect. 3.3.2.

3.3.1 Finite Difference Method

The time independent Schrödinger equation to solve a single particle quantum mechanical problem defined in a three dimensional space in presence of a potential $V(\mathbf{x})$, can be formally written

$$H|\psi(\mathbf{x})\rangle = \left[\frac{\mathbf{p}^2}{2m} + V(\mathbf{x}) \right] |\psi\rangle = \left[-\frac{\nabla^2}{2m} + V(\mathbf{x}) \right] |\psi\rangle = E|\psi\rangle, \quad (3.21)$$

where $\hbar = 1$ and \mathbf{x} , \mathbf{p} are three dimensional vectors for the position and momentum, respectively. To solve numerically this equation, one can introduce artificially a lattice in space $\{\Delta\mathbf{x}\}$ to define a discrete eigenproblem and then check a posteriori the convergence of the solution to the continuous one in the limit $\Delta\mathbf{x} \rightarrow 0$. As for the integration presented in the previous chapter, the estimate of the error as a function of $\Delta\mathbf{x}$ is fundamental for an accurate discretization of the problem (see Appendix B). Hereafter, we will briefly review the simplest possible choice, that is to introduce a lattice with constant spacing h in each spatial direction. The first necessary ingredient to discretize (3.21) is to approximate the second order derivative by means of finite differences, at some order in h . Similarly to what is shown in Sect. B.3 for the first

derivative, the discrete version of the second order derivative of a function $f(x)$ can be computed by means of the Taylor expansion of the function around a given point

$$f(x \pm h) = f(x) \pm hf'(x) + \frac{h^2}{2}f''(x) \pm \frac{h^3}{6}f^{(3)}(x) + \frac{h^4}{24}f^{(4)}(x) + \mathcal{O}(h^5). \quad (3.22)$$

Adding the two equations above and solving for the second derivative of the function we obtain

$$f''_n = \frac{f_{n+1} - 2f_n + f_{n-1}}{h^2} + \mathcal{O}(h^2). \quad (3.23)$$

In conclusion, the discrete version of the time-independent Schrödinger equation at the fourth order in a one dimensional interval $[a : b]$ such that $h = (b - a)/N$ is given by

$$\begin{pmatrix} 2/h^2 + V_1 & -1/h^2 & 0 & \dots & 0 & 0 & 0 \\ -1/h^2 & 2/h^2 + V_2 & -1/h^2 & \dots & 0 & 0 & 0 \\ & & & \vdots & & & \\ 0 & 0 & 0 & \dots & -1/h^2 & 2/h^2 + V_{N-1} & -1/h^2 \\ 0 & 0 & 0 & \dots & 0 & -1/h^2 & 2/h^2 + V_N \end{pmatrix} |\psi\rangle = E|\psi\rangle, \quad (3.24)$$

where $V_i = V(x_i)$ and we set $2m = 1$. The generalization of (3.24) to more dimensions and at higher orders is straightforward. As the reader has undoubtedly noticed, the matrix above is tridiagonal (and in general sparse); thus, it can be highly efficiently diagonalized, see Sect. 2.3. However, especially in dimensions larger than one, the size of the eigenvalue problem typically becomes intractable very fast, as it scales as the number of lattice points in a single dimension to the number of dimensions. For example, keeping one thousand points for each dimension, in three dimensions results into a matrix of the size of a billion, which is hard to attack even with the powerful methods reviewed in Chap. 2. For this reason, the introduction of more efficient discretization methods is highly recommended in such cases, such as the *finite elements methods* [129].

3.3.2 Variational Method

A second approach to treat high dimensional problems is to avoid the introduction of a lattice and perform the discretization using a finite set of basis functions. Indeed, it is always possible to choose a proper set of functions to expand the system wave function

$$|\psi\rangle = \sum_{i=1}^K a_i |\phi_i\rangle, \quad (3.25)$$

where $a_i = \langle \phi_i | \psi \rangle$, and $|\phi_i\rangle$ can be chosen by means of physical intuition or under the assumption that increasing their number K , any set of independent wave functions will eventually cover the full Hilbert space. As we will see later on, it is desirable that the functions $|\phi_i\rangle$ to be orthonormal, however, to provide the exact solution to the problem this is not a necessary condition. Having introduced the basis functions $|\phi_i\rangle$ it is straightforward to compute the matrix representation of the Hamiltonian in such basis, e.g. for a particle living in a one-dimensional external potential,

$$H_{i,j} = \langle \phi_i | \hat{H} | \phi_j \rangle = \int d\mathbf{x} \phi_i(\mathbf{x})^* \left(-\frac{\nabla^2}{2m} + V(\mathbf{x}) \right) \phi_j(\mathbf{x}); \quad (3.26)$$

and the overlap matrix

$$S_{i,j} = \langle \phi_i | \phi_j \rangle = \int d\mathbf{x} \phi_i(\mathbf{x})^* \phi_j(\mathbf{x}). \quad (3.27)$$

A choice of an orthonormal basis clearly results in $S = \mathbb{1}$, however, this is not true in general. Notice that this is the same approach exploited – together with other approximations – by the Hartree-Fock methods, one of the most powerful methods up to date used to compute molecular configurations [16, 134]; and to introduce models such as the Hubbard and Bose-Hubbard models to describe, e.g., electrons in solids and cold atoms in optical lattices respectively [96, 98].

It is now possible to compute the energy of any wavefunction $|\psi\rangle$ taking care of enforcing normalization, that is to compute

$$E = \frac{\langle \psi | \hat{H} | \psi \rangle}{\langle \psi | \psi \rangle}, \quad (3.28)$$

and to minimize such expression to find the ground state energy. If the wavefunctions $|\phi_i\rangle$ are the set of the Hamiltonian eigenfunctions, it is easy to show that the energy of any wavefunction is such that $E \geq E_0$, where E_0 is the system's ground state energy [135]. On the contrary, given a generic finite basis of K wavefunctions $|\phi_i\rangle$ it is possible to look for an approximation from above (the energy expectation value of any wave function is bigger or equal than E_0) of the ground state of the system within the spanned subspace. Once the minimization is performed, it is always possible to increase the number of basis functions to check for convergence.

The expression in (3.28) is minimized by any (normalized) wave function that satisfies the following equality (written in the chosen set of basis functions $|\phi_i\rangle$) with the minimal value of E :

$$\hat{H}|\psi\rangle = E\hat{S}|\psi\rangle. \quad (3.29)$$

If $S = \mathbb{1}$ this is once again an eigenvalue problem of dimension K that can be solved by means of standard methods presented in Chap. 2. On the contrary, it represents a *generalized eigenvalues problem* that can be solved recasting it in a standard eigen-

value problem as follows: the overlap matrix S is Hermitian by definition, and thus it can be diagonalized by means of a unitary matrix W such that $W^\dagger \hat{S} W = D$ where D is some diagonal matrix of real numbers. Rescaling properly each eigenvectors it is then possible to find the orthogonal transformation such that $V^\dagger S V = \mathbb{1}$ and define the wave function $|\psi\rangle = V|\xi\rangle$ for which the eigenproblem of (3.29) can be rewritten as

$$\hat{H} V|\xi\rangle = E \hat{S} V|\xi\rangle \Rightarrow V^\dagger \hat{H} V|\xi\rangle = E V^\dagger \hat{S} V|\xi\rangle = E|\xi\rangle. \quad (3.30)$$

Thus, a standard eigenvalue problem can be defined for the operator $V^\dagger \hat{H} V$ and the wave function $|\xi\rangle$: once solved the solution for the original problem can be found by means of the relation $|\psi\rangle = V|\xi\rangle$.

We conclude this section with an example of such approach, that is, how to compute the solutions of anharmonic oscillator: a one-dimensional quantum particle subject to a third order potential $V(x) = x^2 + x^3$. In this simple case, it is natural to choose the solutions of the harmonic oscillators as the set of basis functions $|n\rangle$ and expand the potential in such basis. Clearly, the result is a matrix with the harmonic oscillator eigenenergies on the diagonal and off diagonal terms given by

$$H_{n,m} = \langle n | \hat{x}^3 | m \rangle = \langle n | \left(\frac{a + a^\dagger}{2} \right)^3 | m \rangle, \quad (3.31)$$

where a^\dagger , a are the creation and destruction operator. The Hamiltonian is then a band matrix with bandwidth seven which shall be diagonalized.

3.4 Time-Dependent Schrödinger Equation

In this section, we review the most common approaches to solve the time-dependent Schrödinger equation building on the linear algebra methods that we have introduced so far: we aim to solve the problem of finding the evolved state $|\psi(t=T)\rangle$ of a quantum system according to

$$i\hbar \frac{\partial |\psi\rangle}{\partial t} = \hat{H} |\psi\rangle \quad (3.32)$$

given the boundary condition $|\psi(t=0)\rangle = |\psi_0\rangle$. The possible applications of such numerical method are ubiquitous and range, for example, from the evolution of a qubit or spin one-half system, to the transport properties into quantum wires or the description of chemical reactions or light-harvesting processes.

3.4.1 Spectral Method

For time-independent Hamiltonians the most straightforward approach one can exploit follows directly the standard quantum mechanics prescription: to decompose the initial state in the Hamiltonian eigenfunctions $|E_i\rangle$ and compute exactly the time evolved state

$$|\psi(T)\rangle = \sum_i c_i e^{-iE_i T/\hbar} |E_i\rangle, \quad (3.33)$$

where $H|E_i\rangle = E_i|E_i\rangle$ and $c_i = \langle E_i|\psi_0\rangle$. This is indeed the most precise method and allows to propagate the system for virtually infinitely large times T , as the error mainly comes from the computation of the eigenstates $|E_i\rangle$. However, in this nice property lays also the seed of its limitation: the computation of the whole spectrum of the system is the limiting factor in terms of scalability. Indeed, despite the powerful methods introduced in Chap. 2, this approach is mainly limited to systems with small Hilbert space size, that is few-body discrete systems or low-dimensional continuous ones.

3.4.2 Split Operator Method

Another efficient approach to solve time-independent Schrödinger equation in real space is the *split operator method*. The first step, as for most of numerical solutions of partial differential equations, is to discretize the time variable, such that $T = n \Delta t$ (see also Appendix B): the formal solution of the Schrödinger equation in (3.4) for the wave function after one time step Δt is given by

$$|\psi(t + \Delta t)\rangle = e^{-i\hat{H}\Delta t} |\psi(t)\rangle, \quad (3.34)$$

where we set $\hbar = 1$. Consider now a generic one-dimensional Hamiltonian defined over a continuous space and composed by a kinetic and a potential part

$$\hat{H} = \hat{T} + \hat{V} = \frac{\hat{p}^2}{2m} + \hat{V}(\hat{x}). \quad (3.35)$$

The split operator method stems from the observation that the two parts of the Hamiltonian being functions of a single variable – either \hat{x} or \hat{p} – are diagonal in the space and momentum representation respectively. Thus, even though the two parts do not commute, exploiting the Baker-Campell-Hausdorff relation [135] we can split the time-evolution operator that propagates the system from t to $t + \Delta t$ as

$$e^{-i\hat{H}\Delta t} \simeq e^{-i\hat{V}\Delta t/2} e^{-i\hat{T}\Delta t} e^{-i\hat{V}\Delta t/2} + \mathcal{O}(\Delta t^3). \quad (3.36)$$

A simple algorithm can now be written to compute the time evolution: each terms in the r.h.s of (3.36) is diagonal either in space or momentum representation. The change of base between position and momentum representation is the Fourier transform \mathcal{F} , for which it exists an highly efficient algorithm, the Fast Fourier Transform (FFT). The FFT scales as $\mathcal{O}(N \log N)$, thus clearly it outperforms a generic change of basis (matrix vector multiplication, $\mathcal{O}(N^2)$). Inserting the changes of bases in the (3.36) we obtain

$$e^{-i\hat{H}\Delta t}|\psi(x, t)\rangle \simeq e^{-i\hat{V}\Delta t/2}e^{-i\hat{T}\Delta t}e^{-i\hat{V}\Delta t/2}|\psi(x, t)\rangle \quad (3.37)$$

$$= U_V(x)\mathcal{F}^{-1}\mathcal{F}U_T(x)\mathcal{F}^{-1}\mathcal{F}U_V(x)|\psi(x, t)\rangle \quad (3.38)$$

$$= U_V(x)\mathcal{F}^{-1}\mathcal{F}U_T(x)\mathcal{F}^{-1}|\psi'(p, t)\rangle \quad (3.39)$$

$$= U_V(x)\mathcal{F}^{-1}U_T(p)|\psi'(p, t)\rangle \quad (3.40)$$

$$= U_V(x)|\psi''(x, t)\rangle = |\psi(x, t + \Delta t)\rangle. \quad (3.41)$$

In conclusion, provided that the proper change of basis is performed right before applying $U_T(p)$ or $U_V(x)$, all operators appear in their diagonal form. Thus, the split operator method requires to store only $\mathcal{O}(N)$ elements and its computational costs is given by the FFT, providing high speed-up when it is possible to apply it. Its precision can be enhanced considering higher terms in the expansion of (3.36) introducing an additional overhead to the algorithm computational cost. The split operator method can be straightforwardly generalized to dimensions larger than one, however, similarly to the spectral method, it can be hardly applied to systems living in more than two dimensions. Moreover, the split operator can also be applied to different Hamiltonians made of generic noncommuting terms. However, the transformation between different bases will differ from a Fourier transform, and thus the favorable scaling of the FFT algorithm cannot be exploited.

3.4.3 Partial Differential Equations Solvers

Whenever the two previous approaches cannot be exploited, one shall apply the direct integration of the Schrödinger equation. The main problem of the formal solution of the Schrödinger equation given by (3.34) is that to compute the exponential of the Hamiltonian operator one shall diagonalize it. However, as we have seen for the spectral method, diagonalizing the full Hamiltonian highly reduce the field of application of the method. Thus, the idea is to compute the relation at the right-hand side of (3.34) without computing explicitly the exponential of the matrix H . Exploiting the fact that Δt can be chosen small compared to any energy scale of the system, the ready solution to the problem is of course to expand the exponential in series of Δt : at first order one recovers the *Euler method*

$$|\psi(t + \Delta t)\rangle \simeq (\mathbb{1} - i\hat{H}\Delta t)|\psi(t)\rangle. \quad (3.42)$$

However, this rough approximation has the major problem that does not conserve the norm of the state. Indeed, the approximated time evolution operator is not unitary. This can be easily seen computing the norm of the time-evolved state after one time step at second order in Δt , $|\langle \psi(t + \Delta t) | \psi(t + \Delta t) \rangle|^2 = |1 - \Delta t^2 \langle \Delta \hat{H} \rangle^2|^2$, where $\langle \Delta \hat{H} \rangle^2 = \langle \hat{H}^2 \rangle - \langle \hat{H} \rangle^2$ is the (positive definite) variance of the Hamiltonian operator computed at time t .

To correct for such inconvenient artifact of the expansion, it is possible to impose unitarity of the truncated time evolution operator, simply expressing the time evolution operator as a function of the product of two adjoint operators. Indeed the following relation holds

$$e^{-i\hat{H}\Delta t} = (e^{i\hat{H}\Delta t/2})^{-1} e^{-i\hat{H}\Delta t/2} \simeq (\mathbb{1} - i\hat{H}\Delta t/2)^{-1} (\mathbb{1} - i\hat{H}\Delta t/2). \quad (3.43)$$

The right hand side of (3.43) is unitary by construction and thus solves the problem of the norm conservation, however it introduces the necessity of computing the inverse of a matrix. The solution of this problem, which goes under the name of the *Crank-Nicolson method* exploits our capability of solving systems of linear equations efficiently, as explained in Chap. 2. Indeed (3.34) and (3.43) implies the relation

$$(\mathbb{1} - i\hat{H}\Delta t/2)|\psi(t + \Delta t)\rangle = (\mathbb{1} - i\hat{H}\Delta t/2)|\psi(t)\rangle \equiv |\phi\rangle. \quad (3.44)$$

Thus, to compute the evolved state $|\psi(t + \Delta t)\rangle$ one has to solve the system of linear equations defined by the above expression.

Finally, we will show that the methods presented above are only a specific example of numerical integration of the partial differential equation: as we have seen in this chapter depending on the order of interpolation used, the precision of the method can increase drastically. Indeed, writing Schrödinger equation in its integral formulation

$$|\psi(t + \Delta t)\rangle = |\psi(t)\rangle + \int_t^{t+\Delta t} \mathcal{L}|\psi(t)\rangle, \quad (3.45)$$

where $\mathcal{L} = -i\hat{H}$, it is clear that solving the problem is equivalent to compute the integral on the right hand side of the equation. Thus, one obtains

$$|\psi(t + \Delta t)\rangle - |\psi(t)\rangle = \Delta t [w_1 \mathcal{L}|\psi(t)\rangle + w_2 \mathcal{L}|\psi(t + \Delta t)\rangle]. \quad (3.46)$$

where w_i are weights that can be chosen freely. The relation above can be rewritten as

$$\hat{A}|\psi(t + \Delta t)\rangle = \hat{B}|\psi(t)\rangle, \quad (3.47)$$

where $\hat{A} = \mathbb{1} - w_2 \mathcal{L} \Delta t$ and $\hat{B} = \mathbb{1} - w_1 \mathcal{L} \Delta t$, and the time propagator is defined as $\hat{A}^{-1} \hat{B}$. The trapezoidal rule prescribes $w_i = 1/2$ which indeed correspond to the Crank-Nicolson method. However, one could in principle make a different choice

(e.g. to approximate the integral with a rectangle of height equal to one extrema of the function) and choose $w_1 = 1$, $w_2 = 0$, obtaining the Euler method, which as we have seen, is less precise than the latter.

Having identified the solution of the time-dependent Schrödinger equation with an integration of the Liouvillian operator \mathcal{L} as in (3.46), it should be now clear how to improve the precision beyond the Crank-Nicolson using the Gaussian integration schemes presented in Sect. 3.2. Indeed, it is possible to introduce K intermediate function evaluations and compute the integral in (3.46) as

$$|\psi(t + \Delta t)\rangle - |\psi(t)\rangle = \Delta t \sum_{k=0}^K w_k \mathcal{L} |\psi(t + \tau_k)\rangle, \quad (3.48)$$

with $\tau_0 = 0$, $\tau_K = \Delta t$. We have seen in Sect. 3.2 that by means of Gaussian integration we can, in principle, achieve an accuracy of $\mathcal{O}(\Delta t^{2K})$. However, here an additional complication arise as in (3.48) $K - 1$ additional variables have been introduced: the intermediate wave functions $|\psi(t + \tau_k)\rangle$, $k = 1, K - 1$ are unknown (differently from previous scenario of the integration of a known function $f(x)$) and shall be computed together with $|\psi(t + \Delta t)\rangle$. The solution of this problem can be found exploiting the structure of the function to be integrated $\mathcal{L}_k \equiv \mathcal{L} |\psi(t + \tau_k)\rangle$, that is, exploiting the relation in (3.48) for each intermediate time steps, the (3.4) and the Taylor expansion in time of the wave function around the initial time t . Indeed, one can write the relation (3.48) for each time steps τ_k as

$$|\psi(t + \tau_1)\rangle - |\psi(t)\rangle = \Delta t T_{1,0} \mathcal{L}_0 \quad (3.49)$$

$$|\psi(t + \tau_2)\rangle - |\psi(t)\rangle = \Delta t (T_{2,0} \mathcal{L}_0 + T_{2,1} \mathcal{L}_1) \quad (3.50)$$

⋮

$$|\psi(t + \tau_k)\rangle - |\psi(t)\rangle = \Delta t \sum_{j=0}^k T_{j,k} \mathcal{L}_k, \quad (3.51)$$

where $T_{i,j}$ is a transfer matrix which contains the to-be-determined coefficients that relate the first $k - 1$ intermediate L_k to the k -th one. It is then possible to write a system of equations for the coefficients w_i (which relate the L_k with the final state), the transfer matrix coefficients $T_{i,j}$ and the intermediate time intervals τ_k such that the gaussian integration is performed. Hereafter we report the solution for $K = 2$, the *Runge-Kutta method* at fourth order. The (3.48) for $K = 2$ becomes

$$|\psi(t + \Delta t)\rangle - |\psi(t)\rangle = \Delta t (w_0 \mathcal{L}_0 + w_1 \mathcal{L}_1). \quad (3.52)$$

Recalling that $\mathcal{L} = \mathcal{L}(\psi, t)$ one can expand it at first order obtaining $\mathcal{L}_1 = \mathcal{L}_0 + \frac{\partial \mathcal{L}}{\partial \psi} \Delta \psi + \partial \psi \partial t \tau_1$. Inserting (3.49), recalling that $\tau_1 = c_1 \Delta t$ for some coefficient c_1 one can rewrite (3.52) as

$$|\psi(t + \Delta t)\rangle = |\psi(t)\rangle + \Delta t(w_0 + w_1)\mathcal{L}_0 + \Delta t^2 w_1(T_{1,0}\mathcal{L}_0 \frac{\partial \mathcal{L}}{\partial \psi} + c_1 \frac{\partial \mathcal{L}}{\partial t}). \quad (3.53)$$

Equating the above expression term by term with the second order Taylor expansion of $|\psi(t + \Delta t)\rangle$ one obtains the system of equations

$$\begin{cases} w_0 + w_1 = 1 \\ w_1 T_{1,0} = 1/2 \\ w_1 c_1 = 1/2 \end{cases},$$

whose solution provides the necessary coefficients to implement a Runge-Kutta solver at fourth order in Δt . Implementations of higher order approximations and adaptive Runge-Kutta methods can be found in most common platforms and programming languages.

3.5 Exercises

1. Consider a one-dimensional quantum harmonic oscillator defined by the Hamiltonian

$$H = \frac{\hat{p}^2}{2} + \frac{\hat{x}^2}{2}$$

and its eigenvalues E_k and eigenvectors $|\psi_k\rangle$ ($\hbar = m = \omega = 1$).

- a. Given the exact ground state $|\psi_0\rangle$, compute numerically the expectation value of the ground state energy $\langle \psi_0 | H | \psi_0 \rangle$ and compare it with the exact value E_0 .
 - b. Is the error due to the wave function discretization or to the approximation of the integral? *Hint: increase the integral approximation (Trapezoidal rule, Simpson's rule ...) for fixed wave function discretization and vice versa.*
2. Consider a one-dimensional quantum harmonic oscillator defined by the Hamiltonian defined in the previous exercise.
 - a. Write a Fortran program to compute the first k eigenvalues E_k and eigenvectors $|\psi_k\rangle$.
 - b. How would you rate your program in terms of the priorities for good scientific software development (Correctness, Stability, Accurate discretization, Flexibility, Efficiency), see Appendix A?

3. Given a time-dependent one-dimensional quantum harmonic oscillator defined by the Hamiltonian

$$H = \frac{\hat{p}^2}{2} + \frac{(\hat{q} - q_0(t))^2}{2};$$

with $q_0(t) = t/T$ and $t \in [0 : T]$. Given $|\psi_0\rangle = |n = 0\rangle$ (ground state of the Harmonic oscillator), compute $|\psi(t)\rangle$ for different values of T .

Part II
The Many-Body Problem

Chapter 4

Numerical Renormalization Group Methods



In this chapter, we introduce the renormalization group (RG) approach from a numerical perspective. While the analytical RG approach is a powerful tool routinely used to study different phenomena [22, 136, 137], hereafter, we concentrate on its application to numerically attack the many-body quantum problem. To set the stage, we start with the mean-field treatment of the many-body quantum problem and present its application to study the quantum Ising model in transverse field. We then introduce the numerical RG approach, and finally, we review the density matrix renormalization group (DMRG) method in its original formulation to highlight its connection with the standard RG.

4.1 Mean Field Theory

Many-body quantum systems are by definition composed by many quantum degrees of freedom. Hereafter, for simplicity, we assume that each quantum degree of freedom is local and lives on a lattice site. The paradigmatic example is a chain of spins, but the same construction can be derived starting from continuous degrees of freedom in many distinct settings, e.g., atoms in optical lattices [96]. The system wave function $|\psi\rangle$ lives in the N -body Hilbert space \mathcal{H}_N formed by the tensor product of all the local ones $\mathcal{H}_N = \mathcal{H}_1 \otimes \mathcal{H}_1 \otimes \dots \mathcal{H}_1$. By definition, the wave function gives the amplitude of probability of each possible system configurations $|\alpha_1 \alpha_2 \dots \alpha_N\rangle$, where the index α_i labels the possible single-body configurations of the i -th lattice site, $\{\alpha_i\}_1^d$ (e.g., if the system is composed by spin one-half systems, $|\alpha_1\rangle = |\uparrow\rangle$ and $|\alpha_2\rangle = |\downarrow\rangle$). Thus, a generic state is described by the N -rank tensor

$$|\psi\rangle = \sum_{\vec{\alpha}} \psi_{\alpha_1 \alpha_2 \dots \alpha_N} |\alpha_1 \alpha_2 \dots \alpha_N\rangle, \quad (4.1)$$

where $\vec{\alpha} = \alpha_1 \alpha_2 \dots \alpha_N$, and $\mathcal{P}_{\alpha_1 \alpha_2 \dots \alpha_N} = |\psi_{\alpha_1 \alpha_2 \dots \alpha_N}|^2$ gives the probability to find the system in the configuration $\alpha_1 \alpha_2 \dots \alpha_N$. Hereafter, the local Hilbert space dimension d can also be the result of a truncation of the local Hilbert space, e.g., while studying bosons in a lattice. Even though here for simplicity we consider the local Hilbert space dimension d uniform throughout the whole lattice, it could also be site dependent. Notice that the dimension of the N -rank tensor $\psi_{\alpha_1 \alpha_2 \dots \alpha_N}$ increases exponentially with the system size.

The mean field approach is a very powerful method to get insights into the physics of many-body quantum systems which relies on a strong approximation [14]. Within the mean field approximation, one assumes that quantum correlations shall be neglected, that is, the many-body wave function is constrained to be the product of N independent single body wave functions,

$$|\psi^{\text{MF}}\rangle = |\psi^1\rangle \otimes |\psi^2\rangle \otimes \dots \otimes |\psi^N\rangle = \quad (4.2)$$

$$= \sum_{\alpha_1=1}^d \psi_{\alpha_1}^1 |\alpha_1\rangle \otimes \sum_{\alpha_2=1}^d \psi_{\alpha_2}^2 |\alpha_2\rangle \otimes \dots \otimes \sum_{\alpha_N=1}^d \psi_{\alpha_N}^N |\alpha_N\rangle. \quad (4.3)$$

Thus, the object of study has been simplified from an exponential to a linear problem in the number of lattice sites N . An additional assumption (even though not strictly necessary) results in a further simplification: imposing translational invariance – all $|\psi^j\rangle$ are equal – recast the original problem into a problem independent of the system size N . Indeed, the number of degrees of freedom (free parameters) is d^N in the original problem, Nd in the mean field scenario and only d in the translationally invariant one. It is then not surprising that, within the mean field approximation, efficient algorithms or even analytical solutions exist to very complex problems. However, the drawback of this powerful approach is the fact that the introduction of (4.3) is, in general, unjustified and uncontrolled. Indeed, independent checks of the mean field approximation are always needed, either via comparison with experimental results or by other means. Nevertheless, whenever quantum correlations do not play a major role, the mean field approach can be very powerful and, as we will see in the later sections, it can give a powerful insight in the physics of the system and can serve as a guide and starting point for further, more refined, analysis.

4.1.1 Quantum Ising Model in Transverse Field

We present the mean field approach using an example of a paradigmatic model, namely the quantum Ising model in transverse field. The quantum Ising model represents one of the simplest nontrivial many-body quantum system, and has played and plays a fundamental role in our understanding of correlated states of matter [97, 137]. As we will show in more details in the last part of this book, this model has been instrumental to our understanding of critical phenomena in and out of equilibrium, and it is the first testbed for any numerical methods which aims to become a

competitive one. The system has an analytical solution – a mapping to free fermions via the Jordan-Wigner transformation – that allows for precise benchmarking of any software developed to study numerically correlated many-body quantum system [97, 138–140]. The Hamiltonian of the model reads

$$\mathcal{H}^I = \sum_{i=1}^{N-1} H_{i,i+1} = - \sum_{i=1}^{N-1} \sigma_i^x \sigma_{i+1}^x + \lambda \sum_{i=1}^N \sigma_i^z; \quad (4.4)$$

where σ s are Pauli matrices and represents a linear chain of N interacting spins 1/2 in presence of an external field of intensity λ . The mean field analysis of this model starts from the definition of (4.3), where

$$|\psi^{\text{MF}}\rangle = \otimes_{i=1}^N |\psi^1\rangle = \otimes_{i=1}^N \sum_{\alpha_i=1}^d A_{\alpha_i} |\alpha_i\rangle; \quad (4.5)$$

where for spin 1/2 one has clearly $\alpha_i = \uparrow, \downarrow = 1, 2$ and $d = 2$. Under the assumption of (4.5), the computation of the system's energy can be readily done as

$$E^{\text{MF}} = \langle \psi^{\text{MF}} | \mathcal{H}^I | \psi^{\text{MF}} \rangle = - \sum_{i=1}^{N-1} (\langle \psi^1 | \sigma_i^x | \psi^1 \rangle)^2 + \lambda \sum_{i=1}^N \langle \psi^1 | \sigma_i^z | \psi^1 \rangle. \quad (4.6)$$

The above quantity is extensive, thus it diverges at the thermodynamical limit, while the energy density $e = E/N$ is intensive and its limit for $N \rightarrow \infty$ can be calculated and is given by

$$e = - (\langle \psi^1 | \sigma_i^x | \psi^1 \rangle)^2 + \lambda \langle \psi^1 | \sigma_i^z | \psi^1 \rangle \equiv -r_x^2 + \lambda r_z, \quad (4.7)$$

where r_x and r_z are the two level parametrization in the Bloch sphere. The expression for the energy density above is then a function of two variables r_x, r_z , which shall be minimized to compute the ground state energy density of the Ising model at the thermodynamical limit within the mean field approximation. This minimization can be performed analytically (even though more complex models might require numerical minimizations which can be performed with your preferred method [122]): the wave function normalization imposes the additional constraint $r_x^2 + r_y^2 = 1$, resulting in an expression function of the external magnetic field λ

$$\begin{cases} e = -1 - \lambda^2/4 & \lambda \in [-2 : 2] \\ e = -|\lambda| & \lambda \notin [-2 : 2]. \end{cases} \quad (4.8)$$

The expression above is not exact; however, it captures the physics of the system in the sense that it signals that there are two special points, $\lambda = -2, 2$ where the second derivative of the energy density becomes a discontinuous function: as we will see in Chap. 7 this signals the presence of a quantum phase transition. The quantum phase

transition is indeed present; however, the mean field analysis is unable to compute the exact transition point and also to correctly describe the correlations occurring at the transition point. Indeed, by construction, the computation of any connected correlation functions $\mathcal{C}_{i,j}$ between two observables \hat{O}_i and \hat{O}_j acting on two different spins will result to exactly zero:

$$\mathcal{C}_{i,j} \equiv \langle \hat{O}_i - \langle \hat{O}_i \rangle \rangle \langle \hat{O}_j - \langle \hat{O}_j \rangle \rangle = \langle \hat{O}_i \hat{O}_j \rangle - \langle \hat{O}_i \rangle \langle \hat{O}_j \rangle \Rightarrow \quad (4.9)$$

$$\mathcal{C}_{i,j}^{\text{MF}} = \langle \psi^1 | \hat{O}_i | \psi^1 \rangle \langle \psi^1 | \hat{O}_j | \psi^1 \rangle - \langle \psi^1 | \hat{O}_i | \psi^1 \rangle \langle \psi^1 | \hat{O}_j | \psi^1 \rangle = 0. \quad (4.10)$$

In conclusion, mean field analyses are fundamental to get an insight of the physics of the system: whenever quantum correlations play a dominant role the mean field result fails to correctly describe the system, and thus calls for more advanced and sophisticated tools which we will introduce in the next sections.

4.1.2 Cluster Mean Field

A first possible correction to the mean field ansatz is cluster mean field. Indeed, whenever it is physical relevant to assume that the system ground state is characterized by short-range correlations, the mean field ansatz of (4.5) can be generalized to encompass them, thus greatly improving the result precision. A simple approach is to write a mean field ansatz of a cluster of M sites, i.e.,

$$|\psi_{\alpha_1 \alpha_2 \dots \alpha_N}^{\text{MF}}\rangle = \prod_{i=1}^{N/M} |\psi_i^M\rangle = \prod_{i=1}^{N/M} \sum_{\beta_i=1}^{d^M} A_{\beta_i} |\beta_i\rangle; \quad (4.11)$$

where the indexes β_i run on all possible states of each cluster. As in the previous section, starting from this ansatz, it is possible to find the cluster mean-field description of the ground state of the system under study. This analysis can also be useful to check the confidence of the results of the mean field analysis: increasing the cluster size and comparing the results, it is possible to estimate the errors and in some cases to extrapolate the results to the exact case $M = N$. Clearly, the exponential dependence of the number of states in (4.11) makes this approach inefficient as increasing the cluster size is exponentially expensive. However, a simple example might be helpful to clarify the scenarios where the cluster mean field will clearly outperform the mean field description. Let's consider the Hamiltonian

$$H_{XY}^D = \sum_{\langle i,j \rangle} h_{i,j} \left(\sigma_i^x \sigma_j^x + \sigma_i^y \sigma_j^y \right), \quad (4.12)$$

where $\langle i, j \rangle$ defines a disjoint cover of a graph (e.g., a two-dimensional square lattice). It is simply possible to solve exactly this problem, diagonalizing each two sites

hamiltonian independently: the result is that the two sites form a Bell state (maximally entangled state) among the interacting pairs of sites. The mean-field treatment of the problem will result in the description provided in the previous section, neglecting the correlations present in the system ground state completely. On the contrary, a cluster mean-field expansion with $M = 2$ will result in the exact solution. Beyond this simple scenario, it shall be now clear that cluster mean-field descriptions encompass exactly M -party entanglement while neglecting the entanglement beyond the cluster size M . In conclusion, whenever the M -reduced density matrix is pure M -cluster mean field provides the exact result. Vice versa, it will provide only an approximate description with non-decreasing precision with M while the computational cost increases exponentially with M . We will see in the next sections how this intimate relation between the purity of the reduced density matrix, i.e., the number of populations different from zero, can be made explicit and exploited to build powerful approximation methods beyond mean-field.

4.2 Real-Space Renormalization Group

Real-space renormalization group method is an approximation method based on a very powerful physical intuition [22]: the hypothesis that the ground state of a system is composed of low-energy states of the system's (non-interacting) bipartitions. Based on this assumption, it is indeed possible to introduce an algorithm that allows describing the ground state properties of many-body quantum systems with large sizes N , up to the thermodynamical limit corresponding to the fixed point of the renormalization flow [141].

The algorithm proceed as follows:

0. Consider a system composed of N sites that can be studied in an exact numerical way. Build the Hamiltonian $\mathcal{H}_N : \mathbb{C}^{d^N} \rightarrow \mathbb{C}^{d^N}$.
1. Diagonalize \mathcal{H}_N , finding its eigenvalues and eigenvectors $\mathcal{H}_N = \sum_{i=1}^{d^N} E_i |E_i\rangle \langle E_i|$, where the eigenvalues E_i are in increasing order. Consider the projector onto the lowest m eigenstates $P = \sum_{i=1}^m |E_i\rangle \langle E_i|$ which project the Hilbert space on the subspace spanned by the first m low-energy laying eigenstates. Compute the projected Hamiltonian $\tilde{\mathcal{H}}_N = P^\dagger \mathcal{H}_N P$ as well as any other needed operator representation in the projected space, i.e. $\tilde{O} = P^\dagger O P$.
2. Construct the Hamiltonian of a system of size $2N$ using the projected Hamiltonian $\tilde{\mathcal{H}}_N$ for each bipartition and the interaction among them, $\mathcal{H}_{2N} = \tilde{\mathcal{H}}_N \otimes \mathbb{1} + \mathbb{1} \otimes \tilde{\mathcal{H}}_N + \tilde{H}_{\text{int}}$. The interaction Hamiltonian can be obtained as $\tilde{H}_{\text{int}} = \tilde{A}_N \otimes \tilde{B}_N$ where \tilde{A} (\tilde{B}) are the projected operator acting on each system bipartition $\tilde{A} = P^\dagger A P$ ($\tilde{B} = P^\dagger B P$).
3. Repeat the steps 1–2 until the desired system size is reached or convergence to the renormalization group fixed point is achieved. Notice that, at each step of the algorithm, the dimension of the described system is doubled ($N \rightarrow 2N$) while the dimension of the Hamiltonian representation is kept constant to m .

In the last forty years, this elegant and powerful idea has triggered an enormous amount of theoretical and numerical work and allowed us to enhance enormously our understanding of many-body quantum systems [136, 137, 142]. However, the underlying assumption are not always true and indeed there are important classes of physical systems that simply violate it. Hereafter, we review one of such examples, to present in detail the algorithm and to highlight the intuition behind the development of yet another powerful algorithm which overcame such issue, the Density Matrix Renormalization Group method introduced in the next section [26].

The problem we consider is one of the simplest problem in quantum mechanics, the free particle in an infinite box potential given by (3.24): $V(x) = 0$ in an finite interval $[a:b]$, while $V(x)$ is infinite otherwise (we set $\hbar = 1$ to simplify the notation). In this example, the scaling quantity is not the number of particles, but the number of points we introduce to discretize the space. Thus, in the first step we set $N = 2$ (two discrete positions, $|-1\rangle, |1\rangle$ in arbitrary units) and write the corresponding Hamiltonian

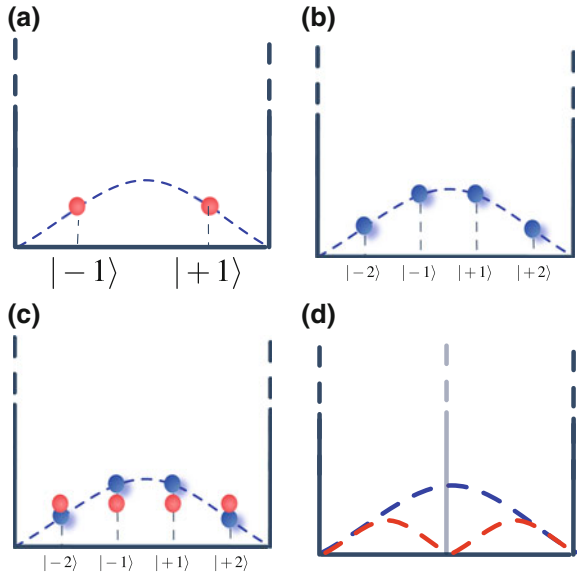
$$\mathcal{H}_2 = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}. \quad (4.13)$$

The Hamiltonian above can be easily diagonalized resulting in $E_{1,2} = 1, 3$, with groundstate $|E_1\rangle^{N=2} = (|-1\rangle + |1\rangle)/\sqrt{2}$. As sketched in Fig. 4.1a, this simple result is in agreement with the known solution of the infinite box potential under the assumption of the minimal discretization introduced above. Following step one of the algorithm above, we now project out high-energy levels of the system. However – given there are only two levels – for this very first step, we keep all of them setting $m = 2$ and, consequently, $P = \mathbb{1}$. We then proceed to step two of the algorithm and build the Hamiltonian of the system with $N = 4$, that is,

$$\mathcal{H}_4 = \begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix}. \quad (4.14)$$

This Hamiltonian is still exact due to the fact that previously all states have been kept ($m = 2$). The ground state of Hamiltonian (4.14) is $|E_1\rangle^{N=4} = (b|-2\rangle + a|-1\rangle + a|1\rangle + b|2\rangle)$ with $a \sim 0.3717$, $b \sim 0.6015$ (represented schematically in Fig. 4.1b), while the first excited state is $|E_2\rangle^{N=4} = (-a|-2\rangle - b|-1\rangle + b|1\rangle + a|2\rangle)$. We keep again $m = 2$ states, and build the effective Hamiltonian which describes the low-energy physics of the system with $N = 4$ with only $m = 2$ states, using the projector $P = |E_1\rangle\langle E_1| + |E_2\rangle\langle E_2|$. We shall also update all relevant operators in the new truncated basis, and repeat the procedure which results in the ground state for a system of double size, and eventually to the thermodynamical limit. Although keeping only two states at each step ($m = 2$) is an extreme approximation, the procedure is correct it will result in an approximate description of the system properties. Increasing the number of kept states m , the approximation can be improved. However, for this precise example, the assumption on which RG is based on is violated, and thus it

Fig. 4.1 Schematic representation of the infinite box potential (black lines) and its solution (dashed blue line). **a** Discretized amplitudes of probability in a minimal representation of a particle in the box, formed by two states (red circles) and **b** a box of double size, formed by four states (blue circles). **c** Comparison between $|E_1\rangle^{N=4}$ (blue circles) and $|E_1\rangle^{N=2} \otimes |E_1\rangle^{N=2}$ (red circles). **d** Large N solutions $|E_1\rangle^{2N}$ (blue dashed line) and $|E_1\rangle^N \otimes |E_1\rangle^N$ (red dashed lines) [143]



performs very badly. Indeed, the ground state in the case of $N = 4$ is composed almost only by the tensor product of two ground states of size $N = 2$ as shown in Fig. 4.1c and as can be verified computing the overlap between the two states $\langle E_1^{N=4} | E_1^{N=2} \otimes E_1^{N=2} \rangle \sim 0.97$. However, the overlap decreases rapidly with N : in the large N limit, the ground state of a system of size $2N$ and the tensor product of two ground states of size N differ drastically (see Fig. 4.1d): in particular, the latter has a node where the former has a maximum. Even worst, all possible combinations of the eigenvalues of systems of size N have a node in the middle, thus, to obtain $|E_1\rangle^{2N}$, a large combination of high-energy eigenstates $|E_i\rangle^N$ is needed, falsifying the assumptions on which RG is based on. However, this counterexample has the merit that pointed out that hard boundary conditions can be highly problematic and eventually resulted in the development of the DMRG presented in the next section [26].

4.3 Density Matrix Renormalization Group

The DMRG is a powerful modification of the original RG algorithm, where the truncation rule is improved resulting in a higher precision description of the final state at the price of slowing down the growth of the system size. Indeed, in the DMRG algorithm, the system size increases linearly instead than exponentially with the number of iterations. The original algorithm, developed to describe systems

with nearest neighbor interaction at the thermodynamical limit (infinite DMRG) is composed of the following steps:

1. Consider the biggest system size N that can be diagonalized exactly with reasonable resources and regroup the N sites in two single sites in the middle and others in two groups of M sites as in Fig. 4.2a (the reason of this grouping will become clear later on). The system's Hamiltonian can be written as $\mathcal{H}_N = \mathcal{H}_{L+1} + \mathcal{H}_{int} + \mathcal{H}_{R+1}$, where \mathcal{H}_{L+1} and \mathcal{H}_{R+1} in the literature are referred at as the left and right enlarged blocks, and \mathcal{H}_{int} is the the interaction among them. The dimension of the Hilbert space of the system is $(dm)^2$, where d is the single site physical dimension and $m = d^M$ the dimension of the M grouped sites. The diagonalization of \mathcal{H}_N returns the ground state expressed in the basis

$$|E_1^N\rangle = \sum \psi_{\beta_1, \beta_2} |\beta_1\rangle |\beta_2\rangle$$

where $|\beta_i\rangle$ spans the basis of the left and right half sites $\beta_i = 1, \dots, d^{M+1}$.

2. Compute the density matrix of the ground state

$$\rho_1^N = |E_1^N\rangle \langle E_1^N| = \sum_{\beta_1, \beta_2} \sum_{\beta'_1, \beta'_2} \psi_{\beta_1, \beta_2} \psi_{\beta'_1, \beta'_2}^* |\beta_1\rangle |\beta_2\rangle \langle \beta'_1| \langle \beta'_2|, \quad (4.15)$$

and the reduced density matrix of one half of the system

$$\rho_L = \text{Tr}_R \rho_1^N = \left(\sum_{\beta_2} \psi_{\beta_1, \beta_2} \psi_{\beta'_1, \beta_2}^* \right) |\beta_1\rangle \langle \beta'_1|. \quad (4.16)$$

3. Diagonalize $\rho_L = \sum_{i=1}^{md} w_i |w_i\rangle \langle w_i|$ and order the eigenvalues w_i (being ρ_L a density matrix they are non negative real numbers such that $\sum w_i = 1$) in descending order. If we assume the system to be left-right symmetric we also have that $\rho_L = \rho_R$. Define the projector $P = \sum_{i=1}^m |w_i\rangle \langle w_i|$ composed by only the first m eigenvectors of ρ_L .
4. The projector P defines a truncation of the Hilbert space from md to m states that can be used to compute the effective Hamiltonian of the system and all necessary operators in the reduced space, $\tilde{\mathcal{H}}_{L+1} = P^\dagger \mathcal{H}_{L+1} P$ and given that $\mathcal{H}_{int} = \sum_k c_k A_L^k \otimes B_R^k$ (where A_L^k and B_R^k acts on the left and right enlarged block respectively) the interaction term in the projected space can be computed applying the projector P on the left and right separately. We thus obtain an effective matrix describing the Hamiltonian for the system of N sites of dimension m instead than md .
5. The algorithm is iterated starting again from Step 1 provided that the Hamiltonian of the left and right block are replaced with the effective Hamiltonians computed in the previous step. The net effect is that one can describe a system of $N + 2$ sites with a Hamiltonian of size $(md)^2$ as depicted in Fig. 4.2b. At every step the size of the described system is incremented by two sites while keeping the computational resources constant.

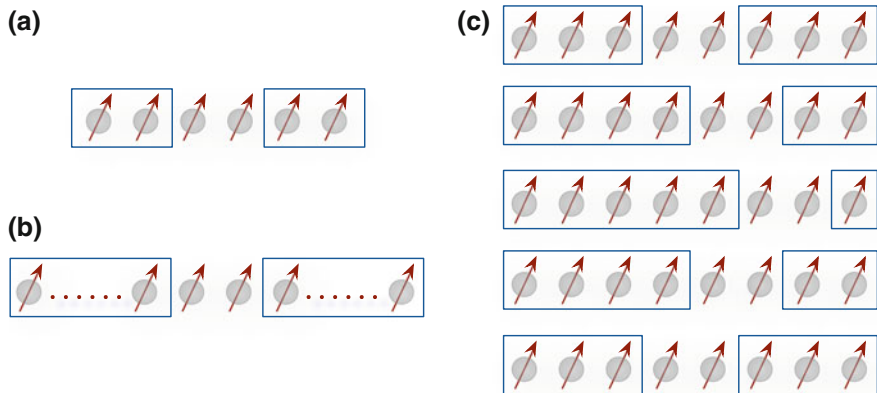


Fig. 4.2 **a** Sketch of step 1 of the INFINITE DMRG algorithm: **a** construction of the system Hamiltonian as left (right) block and two free sites. **b** a system of $2M + 2$ sites is represented by an effective Hamiltonian of dimension md . **c** Graphical representation of half-sweep of the FINITE DMRG algorithm: at each iteration, the number of sites included in one block is increased by one, while the other block decreases by one

The algorithm described above forms the core the DMRG approaches and differs from the RG approaches mainly from the truncation rule: indeed, the choice of keeping the states of the reduced density matrix with higher population can be shown to be equivalent to have the best possible approximated description of the system (in terms of observables and correlations) with the available resources. Indeed, the DMRG truncation rule optimize the overlap between the truncated and the exact system wave functions [144, 145]. For example, the computation of local observables acting on site i (belonging to the left enlarged block, and in general any observables with support only on the left block) for the exact wave function of a system of size N is

$$\langle \hat{O}_i \rangle = \langle \psi | \hat{O}_i | \psi \rangle = \text{Tr}_{\rho_L}(\hat{O}_i) = \sum_{i=1}^{md} w_i \langle w_i | \hat{O}_i | w_i \rangle. \quad (4.17)$$

The expectation value after the truncation step is $\langle \hat{O}_i \rangle_{AP} = \sum_{i=1}^m w_i \langle w_i | \hat{O}_i | w_i \rangle$, thus the difference is $|\langle \hat{O}_i \rangle - \langle \hat{O}_i \rangle_{AP}| \leq \varepsilon \cdot \mathcal{C}$, where \mathcal{C} is an observable-dependent constant and $\varepsilon = (1 - \sum_{i=1}^m w_i)$ is a non-increasing function with m . It is clear that in this basis, given the descending order of the reduced density populations, any other truncation choice would result in a higher error. In particular, if the matrix populations become exactly equal to zero for some $i > K$ (i.e., K is the Schmidt rank of the density matrix), no error is introduced by the truncation in the computation of the expectation value if $m \geq K$. A particular scenario when this can occur is once more whenever the exact system wave function is mean field, that is, $K = 1$ and thus $m = 1$ is sufficient to describe exactly the system properties. However, it should be clear by now that the DMRG can go beyond that, and exactly describe also systems beyond mean field provided that $m \geq K$. One can show that requiring to

optimize the description of the entanglement present in the system (as quantified by the von Neumann entropy $S = -\sum_i w_i \log w_i$) results exactly in the DMRG truncation prescription [144].

The considerations above are equivalent to state that the infinite DMRG algorithm results in the best possible approximation of the reduced density matrix of half of the system given the available resources (m states). However, the optimization concern only the bipartition of the system in two equal subsystems: it might be possible to have a better approximation of the global many-body states if we could optimize the reduced density matrix of all possible bipartitions of the system. This refinement procedure is provided by the finite DMRG algorithm, which aims to refine the representation of a system composed of N sites. The algorithm is defined as follows:

1. Build the representation of a system composed of $N = 2M + 2$ sites using the Infinite DMRG algorithm. At each step store the Hamiltonian (and operators) truncated representations \mathcal{H}_j , with $j = 1, \dots, M$.
2. Start a new DMRG iteration taking care when building the new system Hamiltonian to keep the system size fixed to N . That is, $\mathcal{H}_L = \mathcal{H}_{M+2} + \mathcal{H}_{int} + \mathcal{H}_M$. Notice that the left and right enlarged block now are different and represent a different number of sites. Obtain the truncated representation of the Hamiltonian $\tilde{\mathcal{H}}_{M+2}$.
3. Keep on iterating, increasing the size of the left block and decreasing that of the right block, keeping N constant. When the boundary is reached, reverse the process and keep iterating inverting the role of the left and right block as sketched in Fig. 4.2c. At each iteration, the energy of the system shall decrease while the precision of the wave function representation increases. Once the free sites are in the middle for the second time, a DMRG *sweep* has been completed: additional sweeps will result in higher precision and improved convergence.

The described algorithms can be straightforwardly be generalized to more general scenarios, such that of Hamiltonians with space-dependent parameters or beyond the nearest neighbor interaction (regrouping physical sites in logical ones of larger size, even though this procedure is inefficient and thus always limited to short-range interactions). Finally, notice that Step 1 of the infinite DMRG algorithm can be modified in the sense that one can start with four sites ($M = 1$) and increase the system size without truncating until $d^M \leq m$. A step by step very clear and instructive example of the DMRG algorithm can be found in [146].

4.4 Exercises

1. Given an Hamiltonian on a one-dimensional lattice with nearest neighbor interaction of the general form

$$H = \sum_{i,\alpha} h_i^\alpha + \sum_{i,i+1,\alpha} \lambda_\alpha h_i^\alpha h_{i+1}^\alpha$$

where $i = 1, \dots, N$ runs on the lattice site and α on different operators; consider the mean field translational invariant ansatz

$$|\psi_{MF}\rangle = |\phi\rangle|\phi\rangle|\phi\rangle \dots |\phi\rangle$$

Compute the general mean field approximation of the ground state energy per site and specialize it for the quantum Ising Hamiltonian in transverse field in (4.4) for different values of λ .

2. Compute the mean field approximation of the ground state energy for the antiferromagnetic Heisenberg model

$$H = \sum_i \sigma_i^x \sigma_{i+1}^x + \sigma_i^y \sigma_{i+1}^y + \sigma_i^z \sigma_{i+1}^z$$

using $|\psi_{MF}\rangle$ and the staggered mean field ansatz

$$|\psi_{MFD}\rangle = |\phi_1\rangle|\phi_2\rangle \dots |\phi_1\rangle|\phi_2\rangle.$$

3. Given the quantum Ising Hamiltonian in transverse field of (4.4) on a one-dimensional lattice with nearest neighbor interaction:
 - a. Compute the ground state energy as a function of the transverse field λ by means of the real-space RG.
 - b. Compute the ground state energy as a function of λ by means of the infinite DMRG algorithm.
 - c. Compare the resulting ground state energy density between them and with the mean field solution.

Chapter 5

Tensor Network Methods



In this chapter, we introduce the core concepts of tensor network methods: we first define the tensors themselves and the most common operations for their manipulations. Then, we present some of the most successful algorithms developed exploiting tensor networks to study the many-body problem. We first reformulate the mean-field approach using the tensor notation. Then, we introduce the Matrix Product State (MPS) and the reformulation of the DMRG we presented in the previous chapter in this new formalism. This shift of point of view has been accompanied by an explosion of new classes of variational tensor network states and algorithms, to accommodate different needs and describe more efficiently several physical phenomena. Hereafter, we present some algorithms to simulate the ground state properties and the time evolution of one-dimensional many-body quantum systems; and review the generalization of the MPS to the most straightforward hierarchical structure, the tree tensor network (TTN) and the algorithms for its optimization. The presented material, despite being far from covering the whole fast growing literature dedicated to the field, forms a solid starting point to enter in the field of tensor network methods. We conclude the chapter with a quick overview and pointers to the literature on further tensor network methods recently developed to push beyond our capability of simulating many-body quantum systems.

5.1 Tensor Definition

We first introduce the definition of the mathematical objects we will use hereafter, and of the standard operations to act on them. As already seen in (4.1), an N -rank tensor is an object with N indexes, which, in general, has the form

$$T_{\alpha_1, \dots, \alpha_N}, \quad (5.1)$$

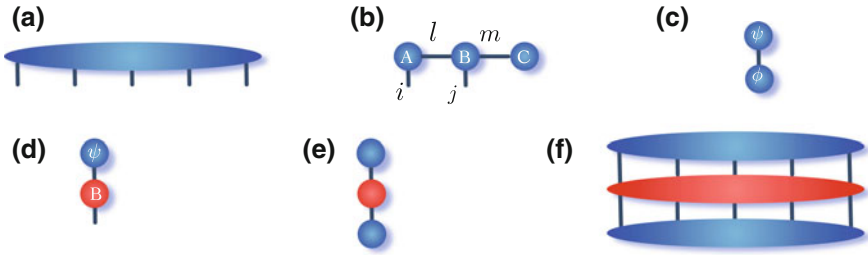


Fig. 5.1 Tensor graphical representations: **a** A n -rank tensor. **b** Tensor contractions equivalent to (5.3). **c** Wave functions scalar product $\langle \phi | \psi \rangle$. **d** Operator B acting on a wave function ψ . **e** Expectation value of an observable (red tensor) of the state represented by the wave function encoded in the blue tensor. **f** As **e** but for a N body wave function

Table 5.1 Definitions of different rank- N tensors

Name	Example	Rank	Notation
Scalar	Constant	0	λ
Vector	Wave function	1	ψ_i
Matrix	Operator	2	$O_{i,j}$
\vdots	\vdots	\vdots	\vdots
Rank- N tensor	N -body wave function	N	$\psi_{\alpha_1, \dots, \alpha_N}$

and can be represented as shown in Fig. 5.1a: a ball with n outgoing links, one for each index. Table 5.1 report the most common names of low rank tensors and typical examples,

A tensor network is a network composed by different tensors, connected via the contraction of different indexes, according to the natural generalization of the standard matrix-vector multiplication rule

$$\hat{O}|\psi\rangle = \sum_j O_{i,j} \psi_j \equiv O_{i,j} \psi_j, \quad (5.2)$$

where in the rightmost equality we introduced Einstein's notation: repeated indexes are implicitly summed. Thus, the contraction depicted in Fig. 5.1b is equivalent to

$$\sum_{l,m} A_{i,l} B_{l,j,m} C_m \equiv A_{i,l} B_{l,j,m} C_m. \quad (5.3)$$

The result is a tensor with two indexes (a matrix) as all others have been contracted. Similarly, the computation of the scalar product between two wave functions, that is,

$$\langle \phi | \psi \rangle = \sum_i \phi_i^* \psi_i = \phi_i^* \psi_i, \quad (5.4)$$

can be depicted as shown in Fig. 5.1c, where we introduce the convention that tensors with indexes pointing upwards are the adjoint of the same tensor with indexes pointing downwards. Notice that the resulting object has no indexes (all indexes are contracted) and thus it is a scalar, as it shall be. The application of an operator to a wave function and the computation of an expectation value are similarly depicted in Fig. 5.1d, e respectively. Again, a check of the correctness of the results can be obtained by counting the final number of free indexes: it should match the expected one for the computed quantity. Notice that computing the scalar product of two N -body wave functions written in a product basis ($|\alpha_1\rangle \otimes |\alpha_2\rangle \otimes \cdots \otimes |\alpha_N\rangle$) does not differ from what presented in (5.4). Indeed, in this case we have

$$\langle \phi_N | \psi_N \rangle = \phi_{\alpha_1, \dots, \alpha_N}^* \psi_{\alpha_1, \dots, \alpha_N} = \phi_i^* \psi_i, \quad (5.5)$$

where we have introduced the notion of global or fused index, that is, the operation of grouping and splitting indexes (introduced in the next section). Finally, one can recast operator operations in similar forms, e.g. the computation of the energy expectation value can be clearly written as

$$\langle \psi_N | \hat{H} | \psi_N \rangle = \psi^{*\beta_1, \dots, \beta_N} H_{\beta_1, \dots, \beta_N}^{\alpha_1, \dots, \alpha_N} \psi_{\alpha_1, \dots, \alpha_N} = \psi^{*i} H_i^j \psi_j. \quad (5.6)$$

5.2 Tensor Manipulations

In this section, we introduce the basics operations on tensors that will be needed throughout the rest of the book to describe faithfully and efficiently the physical system of interests: they allow us to manipulate, compress and transform the entries of the tensors within the tensor networks.

5.2.1 Index Fusion and Splitting

Tensors are nothing more than a structured way to organize information, that is, an ordered collection of numbers or variables. For example, to describe three dimensional relativists systems, the four space-time variables x , y , z , and t , typically are organized in a single vector to improve our representation and manipulation capabilities. However, if it turns out to be more convenient, one could arrange them in a 2×2 matrix. As we will see, the reshuffling of vectors in matrices and the generalizations of this operation, are indeed very useful in different settings. In general, a rank- n tensor can be rearranged in a tensor of a different rank following some given (invertible) rule: in the example given above one has

$$X_i = \begin{pmatrix} x \\ y \\ z \\ t \end{pmatrix} \Leftrightarrow O_{j,k} = \begin{pmatrix} x & y \\ y & t \end{pmatrix}; \quad (5.7)$$

where we have changed the way we organize the variables of interest, and no information is lost. Clearly, also the operations acting on the objects shall be changed accordingly. The transformation performed in (5.7), if applied to wave functions, is known as Liouville representation and, more formally, as Choi-Jamiolkowski isomorphism [147–152]. Notice that in transforming a matrix in a vector, we fuse two indexes in a single one. This index fusion operation can be generalized to any number of indexes simply as

$$\alpha_1 \dots \alpha_n \equiv \text{d-coding}(\mathbf{i}); \quad (5.8)$$

where $\alpha_i = 1, 2, \dots, d$ and the transformation is given by the d -nary coding of the global index \mathbf{i} . From now on, we will indicate the fusion operation as $\alpha_1 \dots \alpha_n \succ \mathbf{i}$. The inverse operation is the splitting of indexes, $\mathbf{i} \prec \alpha_1 \dots \alpha_n$. Finally, note that with the above definition, the last equality in (5.5) automatically holds. Similarly, the relation between the graphical representations in Fig. 5.1e, f is the operation of fusing all indexes into a single one.

5.2.2 Compression

The fusion operation allows one to recast any n -rank tensor in a matrix form. Indeed, it is always possible to choose a bipartition of n indexes in two disjoint groups and fuse together the indexes belonging to each group. That is,

$$T_{\alpha_1, \dots, \alpha_N} \equiv T_{\mathbf{i}, \mathbf{j}}, \quad (5.9)$$

where $\{\alpha_k, \alpha_j, \dots, \alpha_l\} \succ \mathbf{i}$, $\{\alpha_m, \alpha_n, \dots, \alpha_p\} \succ \mathbf{j}$, and $\{\alpha_k, \alpha_j, \dots, \alpha_l\} \cup \{\alpha_m, \alpha_n, \dots, \alpha_p\} = \{\alpha_1, \alpha_2, \dots, \alpha_N\}$. Given that the tensor is now recast in matrix form, one can exploit the whole potential of linear algebra operations acting on matrices. In particular, the Singular Value Decomposition (SVD) – being the generalization of matrix diagonalization – will play a fundamental role in what follows. Indeed, given the tensor $T_{\mathbf{i}, \mathbf{j}}$ it is always possible to decompose it in three tensors via the SVD, such that,

$$T_{\mathbf{i}, \mathbf{j}} = \sum_{\mathbf{k}} S_{\mathbf{i}, \mathbf{k}} V_{\mathbf{k}, \mathbf{k}} D_{\mathbf{k}, \mathbf{j}}, \quad (5.10)$$

where V is a diagonal and positive real matrix, while S and D are unitary matrices. The diagonal elements of the tensor V are real and positive, and thus they are bounded from below by zero and can be ordered. From now on, we assume them to be ordered

in descending order, that is, $V_{1,1} \equiv \lambda_1 \geq V_{2,2} \equiv \lambda_2 \geq V_{3,3} \equiv \lambda_3 \cdots \geq 0$. A central role in tensor network algorithms is played by the fact that, if $\lambda_j < \epsilon$, $\forall j > z$, it is possible to disregard some of the singular values λ_j . That is, whenever the singular values decay fast enough (or are exactly zero after some given value $\lambda_z = 0$) we can neglect them. Correspondingly, the matrices S , V can be truncated. This reduction introduces an error in the representation of the original tensor $T_{i,j}$ which can be estimated in

$$\|T_{i,j} - \sum_{k=1}^m S_{i,k} \lambda_k D_{k,j}\| = \left\| \sum_{k=m+1}^{d^N} S_{i,k} \lambda_k D_{k,j} \right\| < \sum_{k=m+1}^{d^N} \|S_{i,k} \epsilon D_{k,j}\| < \epsilon C \quad (5.11)$$

where C is a finite constant that can be estimated in terms of some matrix norm $\|\cdot\|$ and their dimensions. In particular, if ϵ drops to zero (or equivalently to its numerical representation, typically of the order of 10^{-15} when working in double precision) truncating the tensor dimension to m introduces no error, drastically reducing the computational resources needed to describe the system of interest.

5.2.3 Tensor Network Differentiation

In many cases, we shall perform an optimization with respect to a given figure of merit of the entries of the tensors within a tensor network, for example when minimizing the expectation value of the energy to find a tensor network representation of the system ground state. Thus, the ability to extremize functions of the tensor network's coefficients is fundamental. The computation of the mean energy expressed in (5.6) is a quadratic function of two N -rank tensors:

$$\langle \psi_N | \hat{H} | \psi_N \rangle = \mathcal{H}(\psi_{\beta_1, \dots, \beta_N}^*, \psi_{\alpha_1, \dots, \alpha_N}) \equiv \mathcal{H}(\psi_i^*, \psi_j), \quad (5.12)$$

where we recall that being complex valued, ψ_i^* and ψ_j are independent variables as in complex analysis [153]. Thus, to find the extrema of the function \mathcal{H} with respect to the entries of the tensors, the differential of the function shall be set to zero,

$$\frac{\partial \mathcal{H}}{\partial \psi_j} \stackrel{!}{=} 0; \quad (5.13)$$

which given the expression in (5.6), is equivalent to the condition $\psi_i^* H_j^i = 0$ as the variable appears linearly in the function.

In conclusion, not surprisingly, the derivative of any linear function of tensors (any function defined via a tensor network where each tensor appears only once) with respect to a particular tensor A , is given by the tensor network where the tensor A has been removed. The graphical representation of an example of such computation is reported in Fig. 5.2b.

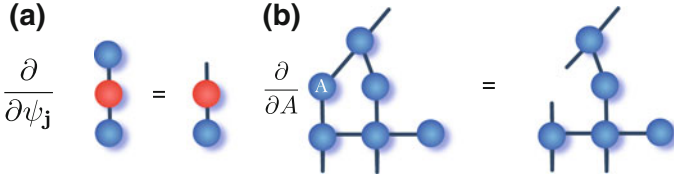


Fig. 5.2 Graphical representation of **a** the derivation of (5.13); **b** a generic partial derivation with respect to a tensor A embedded in a tensor network

5.2.4 Gauging

An operation that plays a prominent role in tensor network algorithms is the gauging, that is, to change the tensors entries by means of local transformations, without changing the global state to exploit some desired properties. Indeed, whenever two tensors are contracted, it is possible to insert an identity operator between them without changing the overall tensor network properties. Using the equivalence $\mathbb{1} = U^\dagger U$, that holds by definition for any unitary operator U , the tensor entries can be changed as

$$A_i B_j = A_i \mathbb{1}_{i,j} B_j = A_i U_{i,k}^\dagger U_{k,j} B_j = \tilde{A}_k \tilde{B}_k; \quad (5.14)$$

with the net result that the tensor structure remains unchanged, but the tensors entries have changed and thus can satisfy some desired properties. Clearly, this freedom can be exploited at the level of the fused indices \mathbf{i}, \mathbf{j} as well as at each single indexes α_j independently.

To avoid a possible confusion, we shall warn the reader that in the following (see Sect. 6.4) we will introduce gauge invariant tensor networks: despite the similar name and the clear connection (in both cases we exploit invariant properties of the tensor network under the action of local transformations), the two constructions are in general different and have different physical origins. Here, we are exploiting the invariance under transformations on auxiliary indexes, in the latter the invariance arises from the symmetries of the theory one is studying, and the gauge transformations act on the physical indexes.

Another powerful manipulation tool is the polar decomposition, that is, the property that every matrix can be decomposed in a unitary and nonunitary part (geometrically, any linear operation can be decomposed in a rotation and a scaling), that is:

$$A_{i,j} = U_{i,k} P_{k,j}, \quad (5.15)$$

where U is a unitary matrix, and P a positive-semidefinite matrix.

As we will see, these operations allow to drastically reducing the computation time of some global contractions. For example, we will show that the computation of the norm of large tensor networks can be reduced to a simple small vector norm computation, exploiting the properties of unitary matrices.

5.2.5 Tensor Contraction Complexity

Hereafter, we show in more details the computational costs of tensor contractions and why it is always important to work with tensors with minimal rank. Indeed, starting from the wave function scalar product depicted in Fig. 5.1c and defined in (5.4), it is straightforward to see that its computation requires $\mathcal{O}(m)$ operations, where m is the dimension of the Hilbert space the wave function lives in, that is, the dimension of the index j . A slightly more complex contraction is represented in Fig. 5.1b and expressed in (5.3). A simple inspection returns the number of operations needed to compute it: for each element of the resulting matrix (two free indices) two nested sums shall be computed, resulting in an overall $\mathcal{O}(m^4)$ operations (assuming all indices have dimension m). From these examples, it is easier to extract a general rule that can be used to compute the number of operations needed to perform a tensor network contraction: *the contraction complexity is given by the product of the dimensions of the free indices and the contracted ones*. Indeed, the product of the free indexes gives the number of coefficients of the resulting tensor to be computed, while the product of the dimensions of the contracted indices gives the order of the number of operations that are necessary to compute each of them. We shall mention that this estimate, although it is very useful to guide algorithm developments, in practice it gives only an estimation of the contraction scaling. Indeed, it is possible to decrease it by using advanced linear algebra manipulation algorithms, e.g., divide-and-conquer schemes for matrix-matrix manipulations [124]: For example, the Strassen algorithm for matrix-matrix multiplication provides a scaling of approximatively $\mathcal{O}(n^{2.8})$ instead of $\mathcal{O}(n^3)$ [154].

In conclusion, the rule above results in an exponential algorithmic complexity of any operation performed on an N -body wave function. Indeed, the computation of the scalar product in (5.5) is $\mathcal{O}(m^N)$ either if performed as the contraction of two rank- N tensors or as viewed as a contraction of two rank-1 tensor (as in the latter case the dimension of the contracted index is m^N). As we will see in the following sections and in particular in Sect. 5.3.3, the introduction of tensor networks reduces in many cases the contraction complexity from exponential to a polynomial function, practically allowing the manipulation N -body wave functions for systems composed of hundreds of sites.

5.3 Ground States via Tensor Networks

In this section, after reformulating the mean field approach in terms of tensor networks, we introduce the most common and successful tensor network state ansatz, the *matrix product state*. Then, we present some algorithms to perform the variational optimization on this class of states to search for ground state properties of many-body Hamiltonians. Finally, we present the generalization of such approaches to general

loop-free tensor networks, concluding the section with an overview on more general and complex looped tensor networks which are the focus of the modern research in TN methods.

5.3.1 Mean Field

Being equipped with the tensor network representation introduced in the last sections, we can now reinterpret the powerful mean field approach introduced in Chap. 4 as a tensor network algorithm. Indeed, the (single-body) mean field approximation introduced in (4.3) can be represented by a tensor network composed by the product of N single rank tensors, as depicted in Fig. 5.3. The task is then to find the entries of the tensors that minimize the expectation value of the energy computed within the mean-field ansatz. Notice that we can comfortably relax the translational invariant condition, i.e., the condition on all tensors ψ^1 of (4.5) to be equal, to accommodate for, e.g., boundary effects and/or non-translational invariant Hamiltonians. Thus, under the assumption that the system of interest is described by a nearest-neighbor Hamiltonian (as, for example, in the case of the Ising model in transverse field defined in (4.4)) the energy expectation value is given by the sum of $N - 1$ terms as depicted in Fig. 5.3d.

Notice that each term of the Hamiltonian can be represented as a tensor network itself, as depicted in Fig. 5.3d (red tensors): indeed, in the case of the Ising Model and many other Hamiltonians, the explicit form of each tensor can be analytically found. The simplest example, in the case of zero transverse field, each red tensor in figure is exactly the correspondent Pauli matrix, and the dimension of the auxiliary index k is one. Alternatively, any two-site operator $H_{\alpha_i, \alpha_j}^{\alpha'_i, \alpha'_j}$ can be decomposed in two smaller tensors via a singular value decomposition as show in (5.10), where the indexes have been fused as $\alpha_i, \alpha'_i > \mathbf{i}$ and $\alpha_j, \alpha'_j > \mathbf{j}$. As we will see later on, this is one of the simplest examples of a Matrix Product Operators defined over two sites

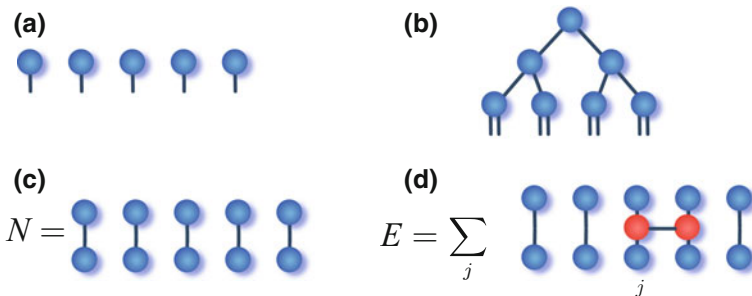


Fig. 5.3 **a** Tensor network representation of the mean field ansatz, **b** of the real-space renormalization group, aka tree tensor networks. **c** Computation of the norm of the state and **d** Energy expectation value E within mean-field approximation

(MPO) which can be used to described and manipulate efficiently operators acting on many-body quantum systems. The computation of each contraction of the tensors trivially results in the norm of each local wave function $|\psi_j^1\rangle$, except for the case where the Hamiltonian term is present. In the later sections, we will see how this simplification can also be introduced in more complex tensor networks, taking care of adequately gauging the tensors.

Finally, the minimization on the tensors entries shall be performed, taking into account that the normalization condition should hold. A formal way of doing such procedure, is to introduce the constraint using a Lagrange multiplier [125], that is, to build build the Lagrangian

$$\mathcal{L}(\psi_1^1, \dots, \psi_N^1, \psi_1^{1*}, \dots, \psi_N^{1*}) = \langle \psi | \mathcal{H} | \psi \rangle - \lambda (\langle \psi | \psi \rangle - 1) \equiv E - \lambda (\mathcal{N} - 1). \quad (5.16)$$

We can search for the minima of the Lagrangian, by iterating over the different variables ψ_j^1 and ψ_j^{1*} . Each of them is then defined by the differential condition (hereafter for easy of notation we assume nearest neighbour interacting Hamiltonians and normalized single-site wave functions $|\psi^1\rangle$, and omit the superscript 1):

$$\frac{\partial \mathcal{L}}{\partial \psi_j^*} = H_{\alpha_{j-1}, \alpha_j}^{\alpha'_{j-1}, \alpha'_j} \psi_{\alpha_{j-1}}^* \psi_{\alpha'_{j-1}} \psi_{\alpha'_j} + H_{\alpha_j, \alpha_{j+1}}^{\alpha'_j, \alpha'_{j+1}} \psi_{\alpha_{j+1}}^* \psi_{\alpha'_{j+1}} \psi_{\alpha'_j} - \lambda \psi_{\alpha'_j} = 0 \quad (5.17)$$

Finally, after computing the *effective Hamiltonian* $\tilde{H}_{\alpha_j}^{\alpha'_j} = H_{\alpha_{j-1}, \alpha_j}^{\alpha'_{j-1}, \alpha'_j} \psi_{\alpha_{j-1}}^* \psi_{\alpha'_{j-1}} + H_{\alpha_j, \alpha_{j+1}}^{\alpha'_j, \alpha'_{j+1}} \psi_{\alpha_{j+1}}^* \psi_{\alpha'_{j+1}}$, the condition above can be expressed as

$$\tilde{H}_{\alpha_j}^{\alpha'_j} \psi_{\alpha'_j} = \lambda \psi_{\alpha'_j}, \quad (5.18)$$

that is, an eigenvalue problem for the effective Hamiltonian \tilde{H} . Solving it by means of the numerical methods introduced in Chap. 2, provides the new entries for the tensor ψ_j : by inspection, the ground state of the effective Hamiltonian provides also the minimal energy for the overall Hamiltonian. Moreover, due to the Hermitian property of the Hamiltonian, the adjoint tensor ψ_j^* solves automatically the problem $\frac{\partial \mathcal{L}}{\partial \psi_j^1} = 0$. Thus, there is no need to iterate on the adjoint part of the tensor network.

In conclusion, to solve a mean field problem within the tensor network paradigm corresponds to the following algorithm:

1. Define a tensor network composed by N independent tensors, each of them represented by a vector of dimension equal to the problem local dimension d . Fill the tensors with random entries and/or with a guess for the system ground state. If necessary, enforce normalization.
2. Iterate over each tensor ψ_j computing the effective Hamiltonian \tilde{H} and solving the correspondent eigenproblem. Update the tensor ψ_j .

3. Sweep over the whole tensor network until convergence or the desired precision is reached.

We stress that despite the fact that the above-sketched algorithm results in a mean-field description of the systems of interest, thus limited by construction, it plays a key role in the development of tensor network algorithms as it provides an operative way to look for ground states of Hamiltonians. Indeed, it contains most of the concepts employed in a large class of tensor network algorithms.

5.3.2 Graphical Tensor Notation

From the equations presented in the previous sections, it is clear that the expressions needed for the tensor network manipulations became quickly overcrowded by indexes which are somehow cumbersome and prone to typos. This is indeed the reason why we introduced the graphical tensor notation in Sect. 5.1. However, the explicit formulas remain hard to read and risk to hide their physical and operational content. Moreover, the standard notation does not convey explicitly some important information, as the information on the gauging. Thus, hereafter we introduce a compressed notation that tries to mitigate these issues, possibly at the cost of being unconventional with respect to standard notation in mathematics and other fields where tensors are used.

The first assumption we make is that tensor networks are oriented, that is, a direction of reference \mathcal{D} has to be chosen. Hereafter, we choose the down-up direction with respect to the page. The second assumption is that no links of the graph representing indexes are allowed to be orthogonal to such direction. This is always possible as single tensors edges can be moved freely. Moreover, in case there are two tensors contracted with an edge orthogonal to \mathcal{D} , a virtual tensor between them can always be introduced, effectively introducing two links which have a different direction without moving the original tensors, see Fig. 5.4a. As a consequence, each link of every tensor in the graph defines a direction (from the tensor) which has a non-zero projection, either parallel or anti-parallel, with respect to \mathcal{D} : we represent the formers as superscripts, while the latter as subscripts in the tensor notation. In general, we adopt the convention that the direction \mathcal{D} corresponds to ket states labeled via subscripts and states defined in the dual space (bra states) are labeled via superscripts. Thus, superscripts contract only with subscripts and vice-versa. Moreover, going in the Liouville representation is equivalent to lowering superscripts and similarly for the reverse transformation (see Fig. 5.4b):

$$L(\rho_i^j) = L(\rho_i^j |i\rangle\langle j|) \equiv \rho_{i,j} |i, j\rangle = \rho_{i,j} |i\rangle |j\rangle = \rho_{i,j}. \quad (5.19)$$

Hereafter, we number the links of the tensor network starting from bottom to top (with respect to \mathcal{D}) of the wave function and left to right, and we write explicitly only the index number, see Fig. 5.4c. According to our graphical notation, performing the adjoint of a tensor, corresponds to swapping super- with subscripts and performing

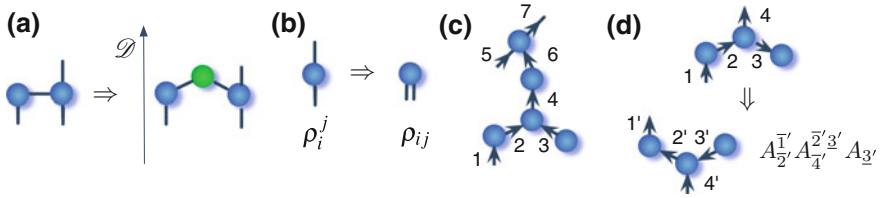


Fig. 5.4 Tensor network notation: **a** virtual tensor (green) introducing a direction in the link. An auxiliary tensor containing the singular values of the left-right bipartition of the system has been explicitly introduced (green rank-2 tensor). **b** Liouville representation of a density matrix. **c** Directed and labeled tensor networks with respect to a reference direction \mathcal{D} . **d** Adjoint operation acting on a directed tensor network with the correspondent tensor notation. Notice that the subscripts lower than the superscripts implicitly implies that the tensor entries shall be conjugated

the complex conjugate of the tensor entries. Notice, that all tensors belonging to a tensor network representing a wave function in the adjoint space will have the superscript smaller than the subscript, thus we can uniquely identify adjoint tensors.

In conclusion, the expression in (5.17) can be rewritten as

$$\frac{\partial \mathcal{L}}{\partial \psi_j} = H_{j-1',j}^{j-1,j} \psi^{j-1'} \psi_{j-1} \psi_j + H_{j',j+1'}^{j,j+1} \psi^{j+1'} \psi_{j+1} \psi_j - \lambda \psi_j \quad (5.20)$$

which despite being still a complex formula, it is more compact than the original one. Notice also that the Hamiltonian, as well as any operator acting on physical indexes, can be recognized by the fact that super- and sub-scripts have the same number.

Importantly, the introduced notation might generate some confusion as ψ_j is typically the j -th element of the vector ψ . Hereafter, whenever we want to specify the single entries of the tensor and some confusion might arise, we will write

$$[\psi]_j. \quad (5.21)$$

Alternatively, it is always possible to go back to the more complete notation reintroducing the greek name of the index ψ_{α_j} . Finally, it will be necessary to introduce a directed tensor network graph, for example, to depict gauged or symmetric tensor networks. Whenever we specify the internal direction of the link, we underline (overline) depending on which direction the arrow points, that is either \bar{j} or \underline{j} . Thus, the adjoint operation of an oriented tensors equals to swapping lower and upper indexes, see Fig. 5.4d.

From now on, we will use the notation introduced here, starting from the MPS ansatz we will introduce in the next section.

5.3.3 Matrix Product States

The Matrix Product State (MPS) is one of the most successful tensor networks introduced so far. It is by now used routinely to study one-dimensional quantum systems in- and out-of-equilibrium. The MPS has been introduced by Ostlund and Rommer [155] and has been recognized to be completely equivalent to a DMRG description [24]. The MPS is defined as

$$|\psi^{\text{MPS}}\rangle = A_1^{N+1} A_{N+1,2}^{N+2} \cdots A_{N+j-1,j}^{N+J} \cdots A_{N-1,2N-2}^{2N-1} A_{N,2N-1} |1, 2 \dots N\rangle \quad (5.22)$$

as depicted in Fig. 5.5b, where all indexes below and equal to N are referred as *physical indexes* while the others *auxiliary ones*. The state is a clear generalization of the mean-field ansatz, as an additional link is present among each different local wave functions $|\psi^i\rangle$. As for the case of cluster mean field introduced in Sect. 4.1.2, these additional links allow representing the correlations between different sites. However, contrary to the cluster mean field ansatz, the structure is translationally invariant, and it allows one to interpolate between the mean field representation and the exact representation of the many-body state, depending on the *auxiliary dimension* of the newly introduced index. Indeed, disregarding momentarily, the problem of efficiency and memory constraints, starting from the exact many-body wave function $\psi_{1,2,3,\dots,N}$, its MPS representation can be built with a subsequent series of singular value decomposition as follows:

1. Group all indexes in two groups splitting the tensor in the middle: $1, \dots, N/2 \succ \mathbf{i}$ and $N/2 + 1, \dots, N \succ \mathbf{j}$, and singular value decompose the matrix $M_i^j = L^{-1}(\psi_{i,j})$. The resulting matrices $M_i^j = S_i^k D_k^j$ (we absorbed the singular values by contracting the matrix V with one of the other matrices) are square matrices of dimension $d^{N/2}$.
2. Repeat the above operation on each matrix separately, splitting the remain physical index in two at each iteration and grouping the auxiliary indexes according to the sketch in Fig. 5.5c.
3. Iterate until each matrix contains only one single physical index and two auxiliary ones. The resulting matrices form the MPS ansatz and define, by construction, the $A_{N+j-1,j}^{N+J}$ matrices.

As can be seen from point 1 above, the described algorithm allows us to map a many-body wave function in an MPS, but it scales exponentially with the number of constituents in the system N . However, suppose that in the process the resulting singular values are all precisely zero apart from the first one, that is, that the dimension of the auxiliary indexes is one. What one unveils is that the original wave function can be expressed exactly in a MPS with auxiliary dimensions one: it is an exact mean-field state. On the contrary, if the state cannot be represented as a mean-field state (i.e., it contains correlations), the auxiliary dimension has to be bigger than one.

The MPS approach, as for all other tensor network algorithms, stems from the assumption that the system of interest can be described with an auxiliary dimension

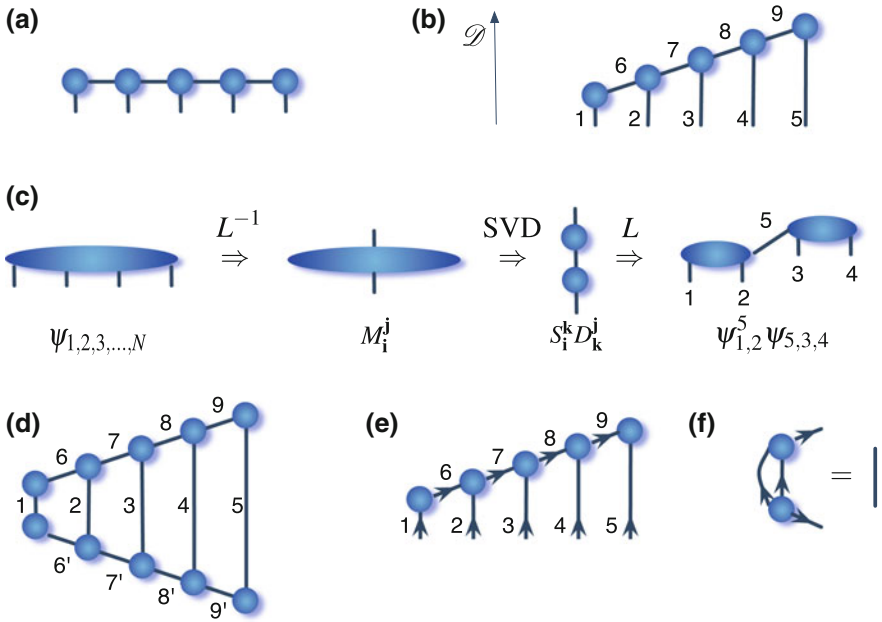


Fig. 5.5 **a** Matrix product state ansatz for $N = 5$ and **b** its ordering. **c** The first iteration of the recasting of a many-body wave function in MPS form. **d** The computation of the norm of an MPS state. **e** A gauged MPS state with respect to the rightmost tensor. **f** Isometric condition: the contraction in (5.24) of the two tensor is equal to an identity

bigger than one (i.e., beyond mean field) but not exponential in N , and thus it is possible to perform an efficient computation of the system’s properties. Thus, $\mathbf{k} = 1, \dots, m$ with m which plays the role of the cut dimension in RG methods (see Chap. 4), and all results shall be checked for convergence in m . Extrapolation in $1/m$ allows extending the results virtually to the exact case, even though the arising errors shall be treated with due care.

The variational algorithm for ground state search within the MPS ansatz follows step by step the algorithm presented in the previous section, with an additional care that shall be taken with respect to the state gauging. We start introducing such a concept and then present the whole algorithm. The main idea stems from the fact that the MPS ansatz has some redundant degrees of freedom that can be used to simplify the calculations, gaining in efficiency and precision. The very first example of such possibility arise whenever the norm of the MPS state is computed

$$\begin{aligned} \mathcal{N} &= \langle \psi^{\text{MPS}} | \psi^{\text{MPS}} \rangle \\ &= A_1^{N+1} \dots A_{N+j-1,j}^{N+J} \dots A_{N,2N-1} A_{N+1'}^1 \dots A_{N+J'}^{N+j-1',j} \dots A^{N,2N-1'} \end{aligned} \quad (5.23)$$

as depicted in Fig. 5.5d. This contraction can be done efficiently: indeed, starting from one extreme (e.g., the leftmost tensors), upper and lower tensor are contracted

($\mathcal{O}(dm^2)$ operations). Then, the resulting tensor is contracted with the other two, one after the other with $\mathcal{O}(dm^3)$ operations. The tensor structure is now effectively shorter by one physical site, and thus repeating the same procedure with the tensors on the right still to be contracted, the norm can be computed with $\mathcal{O}(Ndm^3)$ operations. Despite the fact that the cost needed is linear in N , avoiding these operations could be highly desirable, also because similar computations appear in every step of the minimization algorithm. The solution comes from the gauging as depicted in Fig. 5.5f: indeed, if we gauge the MPS in such a way that each tensor obeys the isometric condition

$$A_{N+j-1,j}^{N+J} A_{N+j-1',j}^{N+J'} = \mathbb{1}, \quad (5.24)$$

the contraction in (5.23) is analytically equivalent to the contraction of the last tensor on the right with itself. This is what is typically referred as an MPS in the right gauge and is represented in Fig. 5.5e. Equivalently, there exists the left gauge when the arrows in the auxiliary dimension in the figure are reversed (and correspondingly the condition in (5.24)). The preparation of a gauged MPS starting from a not gauged one is possible using the tensor manipulation operation introduced previously: by means of a singular value or a QR decomposition, any tensor can be transformed into a product of a unitary and a generic tensor. The unitary tensor redefines the original one while the latter can be contracted into the next tensor in the chain. Repeating this procedure through all the chain, it is possible to enforce the isometric condition to all but the last tensors. In particular,

$$A_{N+j-1,j}^{N+J} \rightarrow A_k^l \rightarrow S_k^j V_j^j D_j^l \rightarrow \tilde{A}_{N+j-1,j}^l K_l^{N+J} \quad (5.25)$$

where now the tensor \tilde{A} obeys the isometric condition and the tensor K shall be contracted into $A_{N+j,j}^{N+J+1}$ before performing its isometrization. A similar procedure can be performed using the QR decomposition [24, 31]. Moreover, by slightly varying the MPS tensor structure, i.e., adding a rank two tensor for each auxiliary link (as shown in Fig. 5.4a), it is possible to always keep explicitly stored the singular values: this gauge allows for an easy control of the entanglement present in the system and parallelization of the variational algorithms [24].

Finally, the algorithm to find the mean-field description of the ground state of a Hamiltonian can straightforwardly be generalized to the MPS tensor structure. Indeed, the algorithm is defined by the following steps:

1. Initialize the MPS ansatz at given auxiliary bond dimension m following the preferred strategy, among which: (a) with random entries, (b) via a guess obtained via analytical formulas, e.g., via perturbation theory or simpler system parameters, (c) grow an MPS (performing an infinite DMRG) until system size N is reached.
2. Define the Lagrangian in (5.16) and find its extremum with respect to all tensors appearing in the network iterating sequentially on each of them, keeping the MPS gauged with respect to the optimized tensor at each step. For each tensor, impose the condition $\partial \mathcal{L} / \partial A_{N+j-1',j}^{N+J+1} = 0$ computing the corresponding effective Hamiltonian and solving the eigenvalue problem for $A_{N+j-1,j}^{N+J}$. Here, the importance of

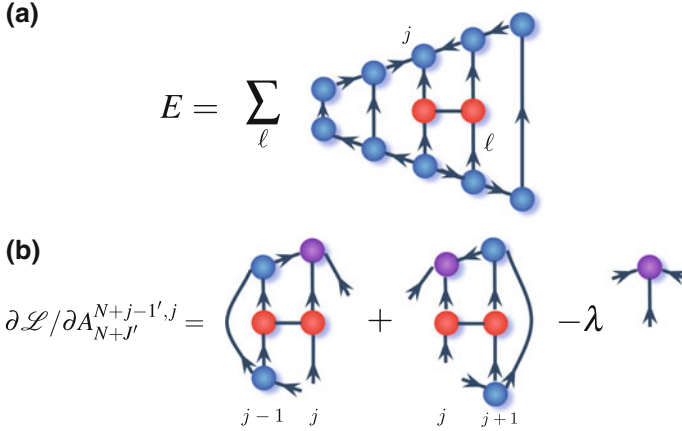


Fig. 5.6 **a** Energy expectation value of an MPS state gauged with respect to the tensor in position j . **b** Computation of the derivative of the Lagrangian in (5.16) with respect to $A_{N+J'}^{N+j-1',j}$ for an MPS gauged with respect to the j -th tensor. The condition $\partial \mathcal{L} / \partial A_{N+J'}^{N+j-1',j} = 0$ results in an eigenvalue problem for the tensor $A_{N+J-1,j}^{N+J}$ (highlighted in purple)

gauging become evident as, as shown in Fig. 5.6, the computation is highly simplified, and it reduces a generalized eigenvalue problem into a standard one, for which numerical methods to solve them are faster and more stable [24, 31].

The computational cost of all operations included in this algorithm is upper bounded by the eigenvalue problem, whose solution can be computed via Lanczos algorithm. Thus, the algorithm complexity scales with the computation of the application of the matrix to the vector, i.e., the tensor contraction represented in Fig. 5.6b, that is, $\mathcal{O}(m^3)$.

The algorithm above optimizes each tensor singularly. Thus it is typically referred as a single tensor update. This is not the only possible choice, as one could solve the eigenvalue problem which arises first contracting two tensors and eventually separating the solution again with an SVD. This second choice is the two tensors update strategy, which strictly resembles the DMRG approach, and plays a prominent role whenever symmetries and thus different charge sectors are considered: it allows one a straightforward strategy to optimize the open charge sectors, see Sect. 6.2, even though it is not the only possibility [31].

Finally, we mention that the algorithms above can be generalized to study also the excited states of the system. Indeed, the Lagrangian (5.16) can be generalized to include also the condition that the state which extremize it, is orthogonal to one or more given states $|\phi_k\rangle$:

$$\mathcal{L}_{ex} = \mathcal{L} - \sum_k \mu_k (\langle \phi_k | \psi \rangle). \quad (5.26)$$

Defining $|\phi_0\rangle$ as the ground state found extremizing the Lagrangian \mathcal{L} , thus allows to find the first excited state of the system (or in case of degenerate ground state, an orthogonal ground state with respect to $|\phi_0\rangle$). Repeating the procedure including an increasing number of states in principle allows computing the whole spectra. However, this approach remains limited to the first low energy levels due to the growing difficulty of convergence and high request of computational resources.

The MPS ansatz is not the unique choice of tensor networks which might be used to extend the mean-field state. Hereafter, we briefly review other possibilities which have been introduced, classifying them as loop-free or looped networks as the different topology introduces a fundamental difference between them: the formers can be used together with algorithms equivalent to that introduced in this section, while the latter requires special attention and much more careful treatment due to higher computational costs and additional technical difficulties.

5.3.4 Loop-Free Tensor Networks

A straightforward generalization of the MPS ansatz is the Matrix Product Operator (MPO) ansatz [156]: it represents the tensor network arising from the application to operators of the concepts and methods introduced above for states. Indeed, to represent an operator the number of physical indexes has to be doubled, one-half for the space it is acting on, and the second half for the dual space. Apart from that, the analysis presented in the previous paragraphs can be straightforwardly extended. In particular, it is easy to show that a MPO is equivalent to a MPS provided that the local dimension is squared: they are linked by a Liouville transformation and a fusion of the two indexes as follows

$$B_1^{3N-1, N+1} \dots \lambda_{N+J, N+J+1} B_J^{N+J+1, 3N+J-2, N+J+2} \dots B_N^{3N-2, 4N-2} \quad (5.27)$$

$$B_{1, 3N-1}^{N+1} \dots \lambda_{N+J, N+J+1} B_{J, 3N+J-2}^{N+J+1, N+J+2} \dots B_{N, 4N-2}^{3N-2} \quad (5.28)$$

as depicted in Fig. 5.7a, b, where the B tensors are represented in red and the λ tensors containing the singular values in green. Finally, the original operator can be represented via a wave function

$$|\psi_{MPO}\rangle = B_1^{N+1} \dots \lambda_{N+J, N+J+1} B_J^{N+J+1, N+J+2} \dots B_N^{3N-2} |\mathbf{1}, \mathbf{2} \dots \mathbf{N}\rangle, \quad (5.29)$$

where $J, 3N + J - 2 > \mathbf{J}$.

A very useful application of the MPO ansatz is its capability of representing Hamiltonians in a compact form. Indeed, having the MPO expression of a Hamiltonian allows computing the energy expectation value by a very simple-to-code and handle tensor network, as represented in Fig. 5.7c. The minimization procedure introduced in Sect. 5.3.3 can be readily applied starting from that tensor network. This approach is particularly relevant to treat long-range interacting Hamiltonians, for

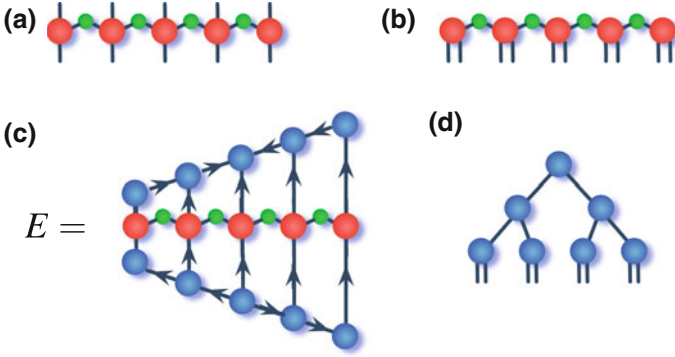


Fig. 5.7 **a** Matrix product operator graphical representation with the tensor $B(\lambda)$ of (5.27) represented in red (green). **b** Graphical representation of the Liouville representation given in (5.28) of the MPO. **c** Computation of the energy expectation value via MPO and MPS tensor networks. **d** Tree tensor network graphical representation

which the algorithm presented in the previous section becomes pretty fast highly involuted, practically preventing its application in most interesting cases.

In the following, we briefly introduce how to recast Hamiltonians in compact MPO form, while we refer the interested reader to relevant literature for the complete mapping into MPO form of long-range Hamiltonians [39, 157]. We consider nearest neighbor Hamiltonians, $\hat{H} = \sum_{j=1}^{N-1} \hat{h}_j \otimes \hat{k}_{j+1}$, where \hat{h}, \hat{k} are operators acting on the local Hilbert space of dimension d . Under this assumption, it is easy to verify that the MPO expression given in (5.27) corresponds to the product of matrices whose elements are matrix themselves. To show how this works, it is instructive to start with a simple example, where $N = 2$. In this case, the Hamiltonian becomes $\hat{H} = \hat{h} \otimes \hat{k} = h_1^4 k_2^5 = H_{1,2}^{4,5}$ which we aim to write it in the MPO form of (5.27), that is for two sites, $B_1^{4,3} B_2^{3,5}$. Equating the two previous relations, it is simple to identify the conditions for the equality to hold: the index 3, i.e. the auxiliary index between the two MPO matrices, shall play no role as it can have only one value, and $B_1^4 = h_1^4$ and $B_2^5 = k_2^5$. To generalize the relation for $N > 2$ we define the products of two operator-valued vectors

$$[B_1^4]^3 = (\mathbb{1}, \hat{h}, 0) [B_3^5]_3 = \begin{pmatrix} 0 \\ \hat{k} \\ \mathbb{1} \end{pmatrix}; \tag{5.30}$$

whose elements are matrices and such that their scalar product is $H_{1,2}^{4,5}$. It can be now easily checked that the general MPO expression for a nearest neighbour Hamiltonian $\hat{H} = \sum_{i=1}^{N-1} \hat{h}_i \otimes \hat{k}_{i+1}$ is formed by the product of $N - 2$ matrices of the form

$$[B_J^{3N+J-2}]_{N+J+1}^{N+J+2} = \begin{pmatrix} \mathbb{1} & h_i & 0 \\ 0 & 0 & k_{i+i} \\ 0 & 0 & \mathbb{1} \end{pmatrix}, \quad (5.31)$$

and in first and last position the vectors of (5.30). Thus, single-operator nearest neighbour Hamiltonians can be represented with an MPO of auxiliary dimension (the dimension of the vectors in (5.30)) $m = 3$. The generalization to Hamiltonians of the form $\sum_{i=1}^{N-1} \sum_{k=1}^p \hat{h}_i^k \otimes \hat{k}_{i+1}^k$ (e.g., the Heisenberg model) is straightforward, increasing the dimension of the MPO bond dimension $m = p + 2$. Finally, as any two-body operator can be rewritten as a sum of at most d^2 terms via a SVD, the maximal needed dimension of a MPO r -range interacting Hamiltonians is $m = d^2 r + 2$ [157].

A second very useful application of MPO is their possible use to describe efficiently many-body quantum systems at finite temperature and, in general, the density matrix of open many-body quantum systems. While with this approach it is possible to study the out-of-equilibrium dynamics, as we will show in the next section [39], it is also possible to search directly for the steady state of a Liouvillian evolution via a variational algorithm [158, 159]. Indeed, by writing the density matrix of the system in an MPS form and the Liouvillian \mathcal{L} in its superoperator MPO form, it is possible to search for the lowest eigenvector of \mathcal{L} or of $\mathcal{L}^\dagger \mathcal{L}$ to find the steady state of the system by adapting the algorithm sketched in the previous section. While the latter choice (minimize $\mathcal{L}^\dagger \mathcal{L}$) ensures the operator to be Hermitian and semi-positive, the former provides an improved numerical performance at the price of a necessary more careful fine-tuning of the convergence [158, 159].

Finally, as said before, MPO can encode general operators which can be exploited to develop different tasks and more specialized algorithms. We will review in some details one of such application in Sect. 6.4, where we present an explicit MPO form for the projectors onto the gauge-invariant subspace of an abelian gauge symmetry.

The MPS are not the only possible tensor network that can be used as a variational ansatz to describe equilibrium properties of many-body quantum systems. Indeed, any loop-less graph can be used to define the correspondent tensor network: the variational algorithm presented in this section for MPS can be applied almost straightforwardly. However, general graphs will eventually result in computational costs which scale at least as the higher rank tensor present in the network (see Sect. 5.2.5). Thus, at constant auxiliary dimension, a convenient balance has to be found: higher rank tensors encodes more information, lower rank ones result in more favourable computational costs. On one edge of this spectrum there are binary trees as depicted in Fig. 5.7d and commonly referred at as Tree Tensor Networks, where each tensor has rank three, the minimal rank necessary to have a non-trivial structure [41, 43, 160, 161]. The network is composed starting from a top rank-two tensor, and then additional layers are created adding a rank-three tensor for each free link. This construction results in a doubling of the number of tensors and of physical indices at each layer. Eventually, after $\log_2 N$ levels, the tensor network can readily accommodate the description of N physical sites. The network is then defined as

$$|\psi^{TTN}\rangle = \prod_{k=1}^{\log_2(N/2)} \prod_{j=1}^{N/2^k} \Lambda_{2^{j-1}+f_1, 2^j+f_1}^{j+f_2}, \quad (5.32)$$

where $f_1 = \sum_{l=1}^{k-1} N/2^{l-1}$ and $f_2 = \sum_{l=1}^k N/2^{l-1}$. Due to its hierarchical structure and to the fact that each couple of tensors is connected via a link-path of length $2 \log N$ at most, this structure is an excellent candidate to describe critical systems (see more on that in Part III), at least in one dimension [43, 162]. Its generalization to higher dimensional systems or general loop-free graphs is straightforward [41, 161, 163].

Notice that under the assumption of translationally invariant tensors (all tensors at each level are equal) and that each tensor is constrained to be an isometry, i.e. $\Lambda_{2^{j-1}+f_1, 2^j+f_1}^{j+f_2} \Lambda_{(j+f_2)}^{2^{j-1}+f_1, 2^j+f_1} = \mathbb{1}$, this tensor structure is equivalent to the real space renormalization group described in Chap. 4. However, it can be shown that relaxing the isometry constraint the precision of the resulting ground state energy remains constant with N at constant bond dimension [31, 43].

Finally, any loop-free tensor network can be represented as rank-three tensor networks using a sequence of SVD, which decompose a rank- n tensor in two rank- $n-1$ tensors. The price to pay is that the auxiliary dimension connecting the two new tensors scales as the maximum of the products of the remaining dimensions of each tensor. However, one can truncate the smallest singular values of each SVD, thus introducing also an additional compression.

5.3.5 Looped Tensor Networks

In the previous sections, we have shown how to optimize loopless tensor network ansatz to represent equilibrium properties of given many-body Hamiltonians. However, there are some scenarios where it is more natural, or it appears to be more convenient, to choose tensor networks with loops. Due to their different topology, such structures present additional difficulties when optimization algorithms are implemented, either due to a less favorable scaling or because they are prone to more numerical instabilities. Thus, for most of them, (if one wants to go beyond a proof-of-principle numerical experiment) the implementation requires additional care and higher expertise. A detailed presentation of such tools go beyond the scope of this book. Hereafter, we briefly recall some of the most common and successful looped tensor networks (also sketched in Fig. 5.8) and refer the reader interested to take this challenge to the relevant literature.

One of the most straightforward scenarios where looped tensor networks appear quite naturally is the infinite MPS (iMPS). Assuming that the system of interest is translationally invariant, one can start from the hypothesis that the entries of all tensors of the MPS ansatz of (5.22) are equal (i.e., a structure as AAAAA) and thus it is possible to work directly at the thermodynamical limit. In such case, the problem can be recast into the minimization of the tensor structure depicted in Fig. 5.8a and

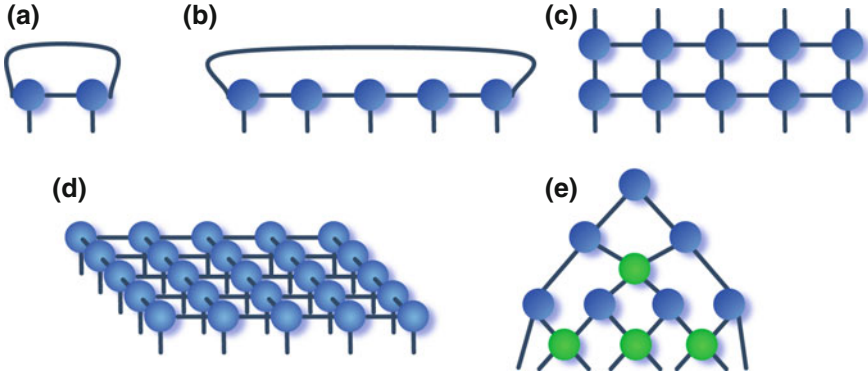


Fig. 5.8 **a** Infinite MPS. **b** MPS with periodic boundary conditions. **c** Locally purified tensor network. **d** Projected entangled pairs state. **e** Multi-scale entanglement renormalization ansatz composed by unitaries (green tensors) and isometries (blue tensors)

standard minimization techniques or imaginary-time evolution can be applied [24, 29]. Given the simplicity of the ansatz, the algorithm can be efficiently and easily implemented obtaining accurate results. However, the drawback of such method is that it cannot be applied to finite systems and thus it hardly compares with experimental results, and it cannot be applied to cases where translational invariance is broken, e.g., in the presence of imperfections, disorder or presence of trapping potential or boundaries. Moreover, being defined only at the thermodynamical limit, finite-size scaling (see Chap. 7) cannot be applied, preventing the use of a very powerful method to characterize the properties of critical systems. The iMPS ansatz can also be naturally extended and improved using a hybrid assumption, that is, to assume that the tensors are equal only with a periodicity bigger than one, i.e., structures as *ABABAB* or *ABCABCABC*, similarly to the spirit of cluster mean-field presented in Sect. 4.1.2.

A second scenario where a looped tensor network has been introduced is to describe one-dimensional systems with periodic boundary conditions. Indeed, in such a case, it is natural to introduce an ansatz of the form

$$|\psi_{\text{PBC}}^{\text{MPS}}\rangle = A_{2N,1}^{N+1} A_{N+1,2}^{N+2} \dots A_{N+j-1,j}^{N+j} \dots A_{N-1,2N-2}^{2N-1} A_{N,2N-1}^{2N} |1, 2 \dots N\rangle, \quad (5.33)$$

as depicted in Fig. 5.8b. Once again, it is possible to apply straightforwardly the ideas presented in the previous sections to this ansatz. However, the additional link that has been introduced, has the disadvantage to increase the scaling of the algorithm and hindering the gauging, practically discouraging its use unless no other solution is at hand and more sophisticated strategies are applied [164]. An alternative approach is based on the re-ordering of the lattice sites from $1, 2, 3, \dots, N$ to $1, N, 2, N-1, \dots$. Thus, it is possible to map the system with periodic boundary conditions to one with open boundaries. Finally, one can use standard approaches either mapping two neighbour physical sites into a logical one at the price of increasing the local

dimension from d to d^2 or, preferably, writing the resulting next nearest neighbour Hamiltonian in a convenient MPO.

Another looped tensor network has been introduced to describe one-dimensional open many-body quantum systems and to cope with the problem of the positivity of the density matrix: the Locally Purified Tensor Network (LPTN) stems from imposing the positivity of the density matrix, writing it as $\rho = XX^\dagger$ [165]. Writing the operator X as an MPO results in a tensor network as depicted in Fig. 5.8c. On such structure, it is possible to apply the strategies presented before, and a generalization to the complete Liouville operator of the time-dependent DMRG presented in the next section [166]. Moreover, being almost one-dimensional, LPTNs profit from the gauging unlike most other looped tensor network.

Finally, other remarkable TN classes, specialized for different scenarios, have been introduced such as Projected Entangled Pair States (PEPS) to simulate two-dimensional MBQS (Fig. 5.8d) [167, 168], the Multiscale Entanglement Renormalization Ansatz (MERA) to study hierarchical scale-invariant systems (Fig. 5.8e) [169–171], the branching MERA [172], the weighted graph states [173], the entangled-plaquette states [174], the string-bond states [175], and the hyperinvariant tensor networks [176]. Covering all these possibilities goes beyond the scope of this book, however, we refer the interested reader to the relevant literature.

5.4 Time Evolution via Tensor Networks

In the previous sections, we have seen how it is possible to find the best tensor network approximations of equilibrium states. In this section, we show how to investigate out of equilibrium properties of many-body quantum systems by means of solving time-dependent Schrödinger equation via tensor networks. The first step in this direction is typically the decomposition of the many-body time-evolution operator via the *Suzuki-Trotter decomposition* in such a way that the exponentially large operator is decomposed and approximated as the product of few-body operators [177, 178]. Hereafter we focus on the case of the dynamics generated by time-independent Hamiltonians. However, the following presentation can be straightforwardly extended to the case of time-dependent Hamiltonians [179, 180].

As for most numerical solutions of time dependent problems, the starting point is to discretize the time axis and thus to recast the time-evolution operator in the form of products of propagators for a small time step Δt as in (3.34). Thus, the task to be solved is to efficiently apply multiple times the operator $\hat{U} = e^{-i\hat{H}\Delta t/\hbar}$ to the tensor network. This challenge is in general double-faced: on the one hand, the tensor structure of the state shall remain invariant at every step, otherwise the algorithm cannot be iterated. On the other hand, the dimensions of each tensor index shall remain constant (or increase up to a given threshold) to maintain the algorithm efficiency. We will see later on how this is possible in many interesting scenarios, starting from the decomposition of the operator \hat{U} in terms of two sets of commuting operators having

disjoint support. For example, for one dimensional nearest neighbour interacting systems, the Hamiltonian operator can be divided in odd and even terms

$$\hat{\mathcal{H}} = \sum_{i=1}^{N-1} \hat{H}_{i,i+1} = \sum_{j=1}^{N/2} \hat{F}_{2j-1,2j} + \sum_{j=1}^{N/2-1} \hat{G}_{2j,2j+1} \quad (5.34)$$

such that

$$[F_i, F_j] = [G_i, G_j] = 0 \quad \text{and} \quad [F_i, G_j] \propto \delta_{i,j}. \quad (5.35)$$

Finally, making use of Baker-Campbell-Hausdorff formula, the time evolution operator can be approximated as

$$\hat{U} = \exp\left(\frac{-i \sum_i \hat{F}_i \Delta t}{2\hbar}\right) \exp\left(\frac{-i \sum_i \hat{G}_i \Delta t}{\hbar}\right) \exp\left(\frac{-i \sum_i \hat{F}_i \Delta t}{2\hbar}\right) + \mathcal{O}(\Delta t^3) \approx \prod \hat{W}_i \quad (5.36)$$

In conclusion, if it is possible to apply a two-body gate to the tensor network representing the state $|\psi\rangle$ preserving its tensor structure and keeping the index dimensions constant, by subsequent iterations the system time evolution can be reproduced. In the following sections we describe multiple strategies available to achieve this goal.

5.4.1 Time-Dependent Density Matrix Renormalization Group

The first algorithm that unveiled the fascinating possibility of direct simulation of the real-time evolution of a many-body quantum system is the time-dependent DRMG (t-DMRG) or Time Evolving Block Decimation (TEBD) algorithm [24, 181, 182]. They provide the sequence of operations that are necessary to time-propagate a state in MPS form: the former in DMRG language, the latter in the equivalent tensor network formulation. Hereafter, we follow the second one for easy of presentation. Having at hand the complete theoretical description and technical tools we have nowadays, this step might appear somehow straightforward, however, its development has been a breakthrough.

As depicted in Fig. 5.9a, the contraction of a single gate acting on sites i and $i + 1$ is a local operation which involves only three tensors: the gate itself and the two MPS tensors containing the physical indices i and $i + 1$. As a consequence, it can be performed efficiently, i.e., with an N -independent scaling, in particular as $\mathcal{O}(m^2)$. Finally, the contracted tensor – which contains two physical indexes – can be split again in two rank-3 tensors using an SVD, which is an operator of the order $\mathcal{O}(m^3)$. Notice that, after the contraction and the SVD the auxiliary dimension connecting the two tensors has increased. To resume the initial structure of the MPS, it shall be truncated keeping only the m highest singular values. These steps can be

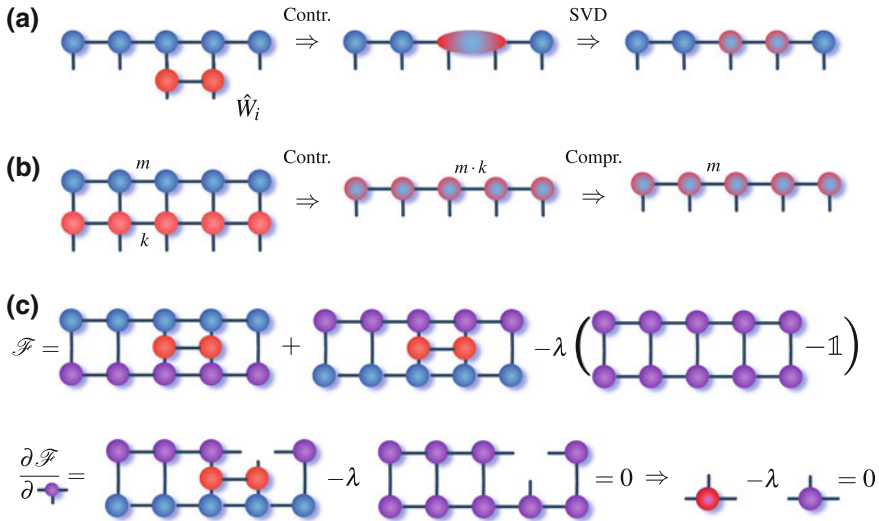


Fig. 5.9 Time-evolution via tensor network: **a** the contraction with the MPS (blue tensors) and subsequent compression via SVD of a single time evolution gate \hat{W}_i (red tensors). **b** The contraction with the MPS to be evolved (blue tensor) of the MPO representation of the time evolution operator \hat{U} (red tensors) followed by an MPS compression via subsequent SVDs. Here the links dimensions are explicitly reported for clarity of presentation. **c** Graphical representation of the figure of merit in (5.37), and of its derivative with respect to a specific tensor $A_{N+j-1,j}^{N+j}$ (in the example $j = 4$). The condition $\partial \mathcal{F} / \partial A_{N+j-1,j}^{N+j} = 0$ results in an update rule for $A_{N+j'}^{N+j'-1,j}$

iterated, contracting the following gates of the expansion in (5.36). Eventually, after contracting all gates, a single Δt time evolution has been performed: concatenating all necessary Δt time evolutions, the state at the end of the evolution is obtained in the form of a MPS with a fixed bond dimension m . Once more, the fixed value of the auxiliary bond dimension m introduces an approximation at every step of the algorithm, which shall be kept under control. It is possible to give bounds on the final error [24, 183, 184], which anyway depends strongly on the kind of time evolutions one is simulating.

Finally, we mention that whenever it is convenient to write the time evolution operator in terms of an MPO (for example in case of long-range interactions), the time evolution can be performed contracting the MPO directly to the MPS representing the state as depicted in Fig. 5.9b, and then performing a compression of the whole MPS [157, 186].

As a final remark, the TEBD scheme can also be used to look for ground states of many-body Hamiltonians merely performing an imaginary-time evolution ($t \rightarrow it$) starting from an initial random state. However, despite the simplicity of implementation, for high performing solutions, a DMRG algorithm is typically preferred due to its faster convergence rate. Indeed, the exponential tails of the imaginary time evolutions drastically slow down its convergence.

5.4.2 Fidelity-Driven Evolution

Another practical way to evolve a tensor network exists which exploits the computation of the fidelity between the evolved state and another tensor network with constant bond dimension: the idea is to search for the tensor network at fixed dimension which best approximates the evolved state. It can be used for complex tensor networks, and in particular for looped ones [171]. The algorithm is again based on the Trotter decomposition and follows the following steps to apply every gate:

1. Apply the gate to the state to be evolved $|\psi\rangle$ and compute the overlap of the evolved state with another general state described by the same tensor network $|\phi\rangle$, $\text{Re}(\langle\phi|\hat{W}_i\psi\rangle)$.
2. Minimize the distance between the two tensors with the additional constraint that the new state has to be normalized

$$\min_{\phi} \mathcal{F} = \min_{\phi} \text{Re}(\langle\phi|\hat{W}_i\psi\rangle) - \lambda(\langle\phi|\phi\rangle - 1). \quad (5.37)$$

The minimization can be achieved, once more, by means of iterative minimization with respect to each tensor $A_{N+j-1,j}^{N+J}$ in the MPS representation of the state $|\phi\rangle$. The gradient $\partial\mathcal{F}/\partial A_{N+j-1,j}^{N+J}$ is depicted in Fig. 5.9c: its extremum can be computed efficiently by contracting the remaining tensor network, resulting in a simple condition for the update of the tensor. If the MPS is gauged with respect to the j -th tensor, the new tensor is proportional to the tensor resulting from the contraction of the term $\langle\hat{W}_i\psi|\phi\rangle$ apart from the tensor $A_{N+j-1,j}^{N+J}$ itself (see Fig. 5.9c).

3. Set $|\phi\rangle = |\psi\rangle$ and iterate the previous step for each tensor until convergence is reached (for small Δt practically few iterations are needed). The resulting $|\phi\rangle$ tensor represents the tensor network approximation at a fixed bound dimension of the time-evolved state.

The algorithm above can be easily adapted to the case where the infinitesimal time-evolution operator \hat{U} is given in an MPO representation.

5.4.3 Time-Dependent Variational Principle

An alternative elegant approach to the simulation of time-evolutions via tensor networks is provided by the *Time-dependent variational principle* [188, 189]: the idea is to project the time evolution operator to the manifold tangent to the MPS space with given bond dimension in such a way that the tensor network structure is preserved by construction. Hereafter, we report the main points of the derivation of the algorithm and refer the reader to the original publications for the explicit mathematical derivation [188]. The TDVP algorithm exploits the gauging of the MPS and updates every tensor of the MPS iteratively, solving the time-dependent Schrödinger equation for an infinitesimal time-step Δt , where the generator of the time-evolution is explicitly

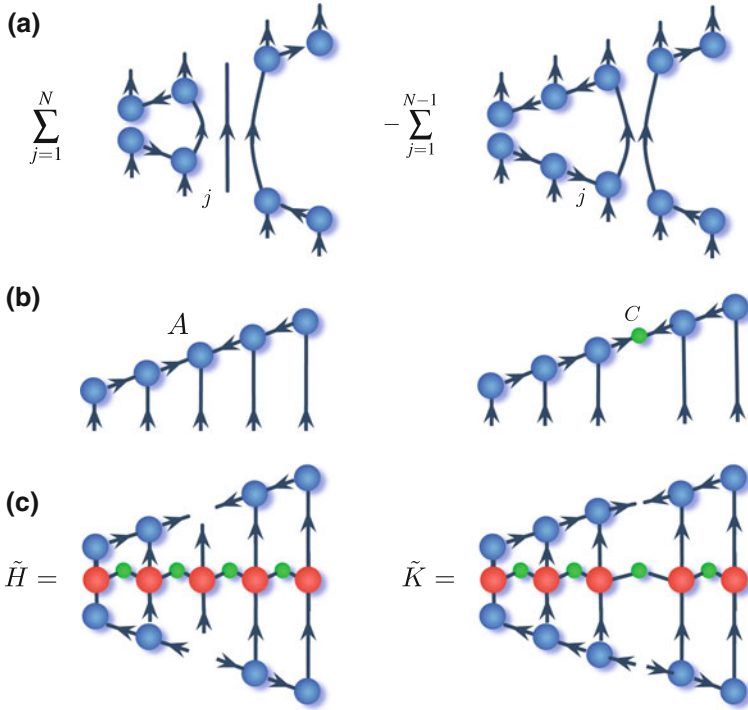


Fig. 5.10 **a** Graphical definition of the tangent space projector $\hat{P}_{\mathcal{T}}$. **b** Definition of the tensor A (left) and C (right) of (5.39) and (5.40). **c** Definition of the effective Hamiltonians \tilde{H} (left) and \tilde{K} from the system Hamiltonian in MPO form (red and green tensors)

projected in the tangent manifold of the MPS:

$$\frac{d|\psi^{\text{MPS}}\rangle}{dt} = -i\hat{P}_{\mathcal{T}}\hat{H}|\psi^{\text{MPS}}\rangle. \quad (5.38)$$

It can be shown that the the projector $\hat{P}_{\mathcal{T}}$ can be expressed in the form depicted in Fig. 5.10a. It then follows that the i -th tensor $A_{N+j-1,j}^{N+j}$ and the vector of singular values $C_{N+J,N+J+1}$ – obtained after a SVD of the tensor $A_{N+j-1,j}^{N+j}$ to change gauge (see Fig. 5.10b) – evolve in time according to the equations:

$$A_{\mathbf{k}}(t + \Delta t/2) = \exp(-i\tilde{H}_{\mathbf{k}}^{\mathbf{j}}\Delta t/2)A_{\mathbf{j}}(t) \quad (5.39)$$

$$C_{\mathbf{k}}(t + \Delta t/2) = \exp(+i\tilde{K}_{\mathbf{k}}^{\mathbf{j}}\Delta t/2)C_{\mathbf{j}}(t), \quad (5.40)$$

where the indexes \mathbf{k}, \mathbf{j} are the indexes resulting by the fusion of the three (two) indexes of the tensor $A(C)$. The effective Hamiltonians \tilde{H} and \tilde{K} are defined graphically in Fig. 5.10c starting from the system Hamiltonian in the MPO representation. In

conclusion, the single-site TDVP algorithm can be implemented as follows: starting from a left gauged MPS, the first tensor A_j^{N+j} is evolved according to (5.39), followed by an SVD of the updated tensor which defines the C tensor to be evolved according to (5.40). The evolved C tensor is then contracted in the next tensor on the right, and the procedure is iterated throughout the whole MPS. Completing a whole sweep (left to right and backward) correspond to evolving the system wavefunction $|\psi^{\text{MPS}}\rangle$ from t to $t + \Delta t$ with an error of $\mathcal{O}(\Delta t^2)$ [188]. This scheme can be straightforwardly implemented also on a general loop-free network [31].

5.5 Measurements

Finally, many interesting physical quantities can be measured efficiently once the entries of the tensor network of choice have been given or optimized to represent the state of interest (e.g., the ground state of a many-body Hamiltonian or the time-evolved state at some final time). We refer the reader to Part III for an explicit introduction and the physical interpretation of such quantities. Hereafter, we define some of the most interesting quantities and present how it is possible to compute them practically in a tensor network language. We explicitly present the calculation for MPS states, however, a straightforward generalization of such procedures is possible for most tensor network structures.

The first class of quantities that can be directly computed are *local observables*, thus, any expectation value of operators with support only on a single local Hilbert space define over a lattice site,

$$\langle \hat{M}_j \rangle = \langle \psi^{\text{MPS}} | \hat{M}_j | \psi^{\text{MPS}} \rangle. \quad (5.41)$$

The local observables, as depicted in Fig. 5.11a, can be readily computed similarly to the computation of the norm of the state. Moreover, the computation is reduced to a system size independent contraction (three tensors) if the MPS is gauged with respect to the j -th site. The second class of efficient measurements is the evaluation of non-local observables such as k -points correlations

$$\langle \hat{M}_{i_1} \hat{M}_{i_2} \dots \hat{M}_{i_k} \rangle = \langle \psi^{\text{MPS}} | \hat{M}_{i_1} \hat{M}_{i_2} \dots \hat{M}_{i_k} | \psi^{\text{MPS}} \rangle \quad (5.42)$$

which (assuming $i_1 < i_2 < \dots < i_k$), in a properly gauged MPS (i.e. with respect to a site $i_1 \leq j \leq i_k$) scales as the distance between the lowest and the highest index $i_k - i_1$, see Fig. 5.11b.

Finally, it is possible to have easy access to the reduce density matrix of any system bipartition, containing k and $N - k$ neighbouring lattice sites respectively,

$$\rho_k = \text{Tr}_{j>k}(|\psi^{\text{MPS}}\rangle\langle\psi^{\text{MPS}}|). \quad (5.43)$$

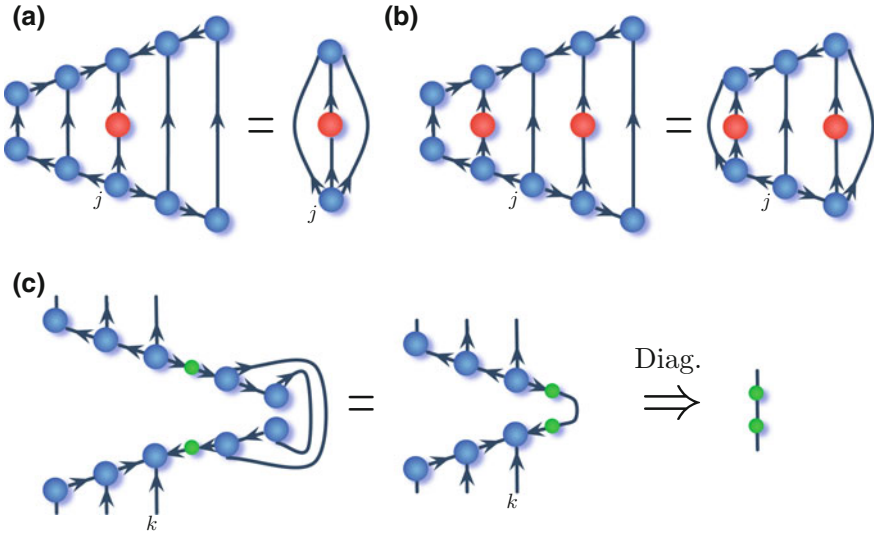


Fig. 5.11 **a** Computation of an expectation value of a single-site operator with support on site j on a MPS gauged with respect to the j -th site. **b** Computation of a two-point correlator on a MPS state properly gauged (see text). **c** Computation of the reduced density matrix $\rho_3 = \text{Tr}_{j>3}(\rho)$ from the density matrix of a system of five lattice sites expressed via MPS state $\rho = |\psi^{\text{MPS}}\rangle\langle\psi^{\text{MPS}}|$. The green tensor contains the singular values λ_1 obtained via the compression of the third and fourth A tensor. The diagonalization of the reduced density matrix can be obtained by means of the unitary operator $U = A_1^6 A_{6,2}^7 A_{3,7}^8$, resulting in the diagonal matrix with diagonal form λ_i^2 (rightmost diagram)

The trace operation is by definition equivalent to the contraction of the indexes $j > k$. If the MPS is in the correct gauge, it is equivalent to eliminate the contracted tensors. The computation of ρ_3 in a system composed of five lattice sites is depicted in Fig. 5.11c, where for clarity and later use we have explicitly represented the diagonal matrix of the singular values λ_i resulting from the compression of the j -th and $j + 1$ -th tensors of the MPS (green rank-two tensor in the figure). While this operation is possible and efficient for any number of sites k , the explicit construction of the reduced density matrix requires an exponential increase memory with k . Thus the explicit computation of ρ_k is still limited to few lattice sites. However, it is possible to compute the first m populations of the reduced density matrix efficiently and thus, an approximation of different entropy measures between the two subsystems bipartitions, as for example all Rényi entropies and the von Neumann entropy. Indeed, the diagonalization of the reduced density matrix can be readily performed applying the remaining MPS tensors themselves which, again due to the gauge condition simplify and result in a diagonal ρ_k (see Fig. 5.11c). The populations of the reduced density matrix are given by

$$p_i = \lambda_i^2. \tag{5.44}$$

As we will see in the next chapters, this capability will play a major role in the characterization of equilibrium and out of equilibrium properties of many systems, from the characterization of quantum phase transition to the estimation of the efficiency of quantum computations.

5.6 Further Developments

The tensor network ansatzes and algorithms introduced so far form the basic tools to study mainly one-dimensional finite closed many-body quantum systems. However, starting from the tools introduced in this chapter, tensor network methods have been extended – with different levels of developments and success – to encompass many other scenarios. Indeed, for example, TN formulations to address directly the thermodynamical limit [51, 52] or to work in the momentum or hybrid space [190–192] have been introduced. TN methods to compute thermodynamical properties of many-body quantum systems have also been presented [183, 193]. Moreover, as already mentioned, the extension to finite temperature and to study open Lindbladian dynamics has been introduced by means of MPDO or LPTN ansatzes and by means of quantum trajectories [50, 166, 194–198].

Other extensions of TN methods go in the promising direction of describing systems defined in the continuum, e.g., quantum field theories. The first attempts to combine second quantization and TN methods have been followed by a more rigorous formalization in the so-called continuous MPS [199–201]. Alongside, condensed matter systems such as Wigner crystals has been efficiently described using an adaptive local basis [202].

Finally, worth mentioning are the increasing number of applications of TN methods to different fields beyond condensed matter and quantum science. On top of the already mentioned applications to the study of lattice gauge theories for high-energy physics (see Sect. 6.4); the application of TN methods to quantum chemistry problems is fast increasing: investigations are ongoing to individuate the most promising approach and to benchmark them against standard approaches [203–206]. The theoretical construction underlying the idea of the MERA tensor network has been exploited in some quantum gravity theory. Indeed, it is now conjectured that the MERA realizes some features of the CFT/AdS correspondence [207–210] and some numerical calculations – which could support the thriving theoretical activities along these lines – have been already performed, see e.g. [211, 212].

Last but not least, there are many possible applications of tensor network methods to classical computer science problems. Indeed, being a class of tools to efficiently perform manipulation, compression, and optimization of a large amount of data, different applications are being explored: from image compression to machine learning and data analytics [213, 214].

5.7 Software

To date, there are plenty software platforms available which allow entering the world of tensor networks at different levels. Some of them are ready-to-go suites which allow starting investigating the model of interest without almost any knowledge of the details of the program running under the hood. The itensor and the TNT libraries, allow one to conveniently manipulate tensors and write TN algorithms without the need of low-level programming [215, 216]. The TNT libraries also includes a DMRG/MPS code, similarly to other implementations freely available such as – among others – the ALPS project and the openMPS code [217, 218]. It is also possible to freely download specific implementations of TN codes tailored to attack the electronic structure problems for quantum chemistry [205, 219, 220].

Finally, a ready-to-go, simple but reliable and flexible (with an arbitrary number of Abelian symmetries embedded) DMRG and t-DMRG code has been distributed since 2006 by the author of this book and his collaborators [178].

5.8 Exercises

1. Define a class of objects to describe general n -rank tensors with the basics operations acting on them: initialization and contraction.
2. Include in the previous exercise the functions acting on a general tensor of indexes fusion, reshaping, SVD and compression.
3. Exploiting the code developed in the previous exercises, define the object MPS and the basic operations on it: computation of the norm and evaluation of expectation values of local and nearest neighbor operators.
4. With the tools developed above, write a t-DMRG code for the Ising model in transverse field. Perform the imaginary time evolution starting from a random MPS and compute the resulting ground state energy with those computed using other methods (see exercises in the previous chapter).

Chapter 6

Symmetric Tensor Networks



As well known, symmetries play a fundamental role in physics: their mathematical description has been used heavily in many branches of physics to understand the main properties of the systems of interest, simplify their description, and improve the numerical performances of numerical codes employed to describe them. The most straightforward scenario in quantum mechanism – familiar to any physicist – where symmetries can be exploited, is that of a system described by a Hamiltonian invariant according to a given transformation g . Elementary quantum mechanics shows that the Hamiltonian is degenerate (i.e., different eigenstates with the same energy exist that behave differently under the action of the transformation), and the Hamiltonian and the operator g generating such transformation commute and can be simultaneously diagonalized [135]. That is, the system's eigenstates can be labeled according to a multiplet of labels which uniquely identify each system's eigenstate: the typical example being the eigenstates of any system with spherical symmetry (e.g., the Hydrogen atom), whose eigenstates can be labelled via three indexes n, l, m ; the first one identifying the energy level, the others the system angular momentum and its projection on the quantization axis and thus the corresponding spherical harmonic [135]. Finally, Noether's theorem states that a system invariant with respect to a given continuous transformation has a conserved continuous quantity [221, 222].

In many-body physics, symmetries are combined with the additional degree of freedom introduced by the spatial extension of the system. Indeed, they might appear as global symmetries (treating the whole system as a single one, e.g., reflection symmetry, translational invariance, etc.), as global point-like symmetry (a global symmetry which results as invariance under the action of a combination of local observables, as the global magnetization, the total charge or the rotation of spins around their own axes) and gauge symmetries (independent local symmetries as Gauss' law). Hereafter, we show how symmetries can be exploited to boost the performance of tensor network algorithm and to address specific symmetry sectors.

In this chapter, we first recap the most important elements of group theory, the mathematical description of symmetries and the theoretical ground of the formulation of symmetric tensor networks. Then, we introduce the reader to symmetric global point-like and gauge invariant tensor networks. This chapter is meant to be an introduction to the subject where technicalities are hidden as much as possible: the readers interested on the details of the technical aspects and of the implementation of symmetric TN, are invited to read [31, 59, 79] on which this chapter is mostly based on.

6.1 Elements of Group Theory

The statements reported in the introduction of this chapter can be introduced starting from the mathematical definition of a group \mathcal{G} : a set of elements g_i equipped with a multiplication operation (any composition rule between two elements of the set) which satisfies the conditions: (i) of being closed under multiplication, (ii) the associative property holds, (iii) the identity exists in the set and (iv) the inverse of each element in the group exists in the set. If the multiplication is commutative the group is said to be *Abelian*, non-Abelian otherwise. Groups can be as simple as the identity group (composed only by one element, the identity) or composed of infinite elements. The number of groups elements is the *order* of the group. From the previous very abstract definition, it is possible to build a group-multiplication table which reports the result of all possible multiplication of the group elements. The multiplication table uniquely characterizes the group: two groups with the same multiplication table are said to be *isomorphic*, that is, there is a unique one-to-one correspondence between the elements of the group. A slightly more general property between groups we will need soon is that of homomorphism: two groups are *Homomorphic (group)* if it exists a correspondence one-to-many among them that is, for each element of the first group A_i^1 we can associate a set of elements in the second group $\{A_j^2\}_{j=1, \dots, m_i}$ such that if $A_i^1 A_j^1 = A_k^1$ then the product of any elements in the correspondent sets $\{A_i^2\}, \{A_j^2\}$ belongs to $\{A_k^2\}$.

Group theory is related with the symmetries introduced before as, defining some operations on the system (translation, rotations, etc.), it is possible to combine (multiply) them. If they form a group, it is possible to build the correspondent multiplication table. However, the previous abstract construction would be of hard use in quantum mechanics if not connected with the standard description that a physicist has of symmetry operations. The bridge is given by the theory of group representation: a representation of a group is any group of concrete mathematical entities (hereafter square matrices) homeomorphic to the original group. If every matrix is different, then the two groups are isomorphic, and the representation is said to be faithful or true. The dimension of such matrices is called *dimensionality* of the representation. Finally, from a representation it is possible to build an equivalent one by means of a similarity transformation S , i.e., $A' = S^{-1}AS$. Indeed, the group multiplication table

is preserved under the action of a similarity transformation, and all representations related by a similarity are said to be *equivalent*.

Given any representation, it is always possible to build another representation by doubling the matrix dimension and defining its elements as a block-matrix with each block composed by the original matrix, i.e., $A^2 = A^1 \oplus A^1$: this is indeed a valid representation (preserve the multiplication table) but is *reducible*. A similarity transformation might conceal the block structure, and thus, to check for reducibility of a representation, one shall look for the existence of a similarity transformation that brings all the elements of the group to the same block structure. If this is not possible, the representation is said to be the *irreducible* representation (irrep), as it cannot be reduced to representations of smaller dimensionality.

In conclusion, it is possible to associate a matrix to any element in a group g , in such a way that according to the standard matrix multiplication, the group matrix multiplication table is reproduced. Moreover, it can be proved that any group representation by matrices (with non-zero determinant) is equivalent to a representation by unitary matrices $U(g)$ [223]. That is, a symmetry is identified by a unitary representation $U(g)$ of a group \mathcal{G} which commutes with the Hamiltonian: $[U(g), H] = 0 \forall g \in \mathcal{G}$ [223]. As a consequence, simultaneously diagonalizing the Hamiltonian and the $U(g)$, it is possible to find an eigenbasis of the Hamiltonian such that each energy eigenstate can be labeled also via the eigenvalues of the group unitary representations (commonly referred as quantum numbers). Abelian groups, on which we will concentrate on hereafter for the sake of simplicity, have one-dimensional unitary (complex) irreps. Specifically, as the group is Abelian (i.e. the elements commute and thus they can all be simultaneously diagonalized) such irreps in their diagonal form are phase factors $W^{[\ell]}(g) = e^{i\varphi_\ell(g)}$, where the phase φ depends on the quantum number ℓ and group element g , and $U(g)|\psi_\ell\rangle = W^{[\ell]}(g)|\psi_\ell\rangle$.

6.2 Global Pointlike Symmetries

In this section, we introduce one of the most common scenarios where a global symmetry is present and that can be exploited to build symmetry-invariant tensor networks: a system with a constant number of particles, such as atoms (either bosons or fermions) hopping and interacting on a lattice. To boil down the presentation to one of the simplest non-trivial scenarios where this construction can be exploited, we consider the one-dimensional Hamiltonian:

$$H = -t \sum_i (c_i^\dagger c_{i+1} + c_{i+1}^\dagger c_i) \quad (6.1)$$

where c_j^\dagger (c_j) are creation (annihilation) operators at the site j of the lattice, t is the tunneling matrix element, m the mass of the particles, and the sum runs over the lattice sites. The Hamiltonian of (6.1) clearly conserves the total number of particles in the lattice. Indeed, it can be easily checked that it commutes with the total particle

number operator $\mathcal{N} = \sum_i n_i$, where $n_i = c_i^\dagger c_i$, that is, $[H, \mathcal{N}] = 0$. However, the Hamiltonian of (6.1) has also another important property: it is invariant under the transformation

$$c_i \rightarrow e^{i\varphi} c_i, \quad \varphi \in [0, 2\pi], \quad (6.2)$$

indeed the two phases cancel out due to the presence, in each term, of a creation and an annihilation operator (for which $c_i^\dagger \rightarrow e^{-i\varphi} c_i^\dagger$ holds), and the Hamiltonian remains invariant. Notice that the invariance holds because the transformation is defined with a constant φ , independently from the lattice site. On the contrary, if we allow a site-dependent φ_i the Hamiltonian is not invariant anymore: an invariant Hamiltonian can be built adding an additional operator, as we will see in the next section.

It shall be clear by now the connection with group theory revised in the previous section: the Hamiltonian of (6.1) is invariant under the action of the transformation defined in (6.2), that is, the unitary representation of an Abelian group (two different phase rotations commute). In particular, (6.2) defines a $U(1)$ symmetry, the invariance in under rotation in the complex plane, for which the group parameter is the rotation angle ϕ , the labels of the irreps are $\ell \in \mathbb{Z}$, and the phase $\varphi_\ell = \phi\ell$.

There is, however, an additional step to be carefully identified before moving forward: as we have seen before, the transformation has to be applied to all lattice sites at the same time, with the same phase. This springs from the fact that we moved from a single-body system described in Sect. 6.1, to a many-body setting: the many-body Hilbert space is the tensor product of the single body ones, thus we have that irreps assume the form $U(g) = \bigotimes_j W_j(g)$, where $W_j(g)$ is the local representation of group element g at site j , and it does not depend explicitly on j . We refer to this particular kind of symmetries as *global pointlike* symmetries. In particular, Abelian global pointlike symmetries acts on the many-body wave function $|\Psi\rangle = \sum_{\vec{\alpha}} \psi_{\alpha_1\alpha_2\dots\alpha_N} |\alpha_1\alpha_2\dots\alpha_N\rangle$, as

$$U(g)|\Psi\rangle = \bigotimes_j W_j(g)|\Psi\rangle = \sum_{\vec{\alpha}} \psi_{\alpha_1\alpha_2\dots\alpha_N} \prod_j e^{i\varphi\ell_j} |\alpha_j, \ell_j\rangle, \quad (6.3)$$

where we have explicitly introduced the quantum number ℓ_j which labels each local basis state $|\alpha_j\rangle$ according to how it transforms under $W(g)$. It should now became clear that as it is possible to label each single-body state according to the quantum number ℓ , it is also possible to label and characterize each many-body basis state $|\alpha_1\alpha_2\dots\alpha_N\rangle$ according to a global quantum number (or *charge sector*). Indeed, from (6.3) follows straightforwardly that

$$\prod_j e^{i\varphi\ell_j} |\alpha_j, \ell_j\rangle = e^{i\sum_j \varphi\ell_j} \prod_j |\alpha_j, \ell_j\rangle. \quad (6.4)$$

That is, every many-body basis state transforms under the action of the global pointlike symmetry as the single-body basis states, but with a phase which is the sum of all local phases $\varphi_{\ell_1,\dots,\ell_N} = \sum_j \varphi\ell_j$. In particular, for $U(1)$ global pointlike symme-

tries, we have $\varphi_{\ell_1, \dots, \ell_N} = \phi \sum_j \ell_j \equiv \phi_{\mathcal{N}}$, and the many-body quantum number \mathcal{N} characterize the global charge sector of the state, in this case the total number of particles in the system.

The tensor network ansätze introduced in Chap. 5 can be improved exploiting the symmetric properties introduced above. If the Hamiltonian conserve the total number of particles, i.e., is invariant under $U(1)$ global pointlike symmetry, it exists an eigenbasis that simultaneously diagonalizes \hat{H} and $\hat{\mathcal{N}}$. This means that we can label the system's eigenstates according to \mathcal{N} and look for, e.g., the ground state for fixed particle number (charge sector) \mathcal{N} or the time evolution of an initial state with well-defined particle number. Then, by definition, these states are composed by a superposition of states belonging to a single charge sector, that is,

$$|\Psi^{\mathcal{N}}\rangle = \sum_{\vec{\alpha} \in \mathcal{U}_{\mathcal{N}}} \psi_{\alpha_1 \alpha_2 \dots \alpha_N} |\alpha_1 \alpha_2 \dots \alpha_N\rangle; \quad \mathcal{U}_{\mathcal{N}} \equiv \left\{ \{\alpha_i, \ell_i\}_i \mid \sum \ell_i = \mathcal{N} \right\}, \quad (6.5)$$

where in the definition of the set $\mathcal{U}_{\mathcal{N}}$ we have again explicitly introduced the quantum numbers ℓ_i . Our aim is to write a tensor network ansatz to describe states defined in the previous equation.

These states, and in general any a state which is invariant under the action of the symmetry, according to (6.3) obey the relation

$$U(\phi)|\Psi^{\mathcal{N}}\rangle = e^{i\phi_{\mathcal{N}}} |\Psi^{\mathcal{N}}\rangle. \quad (6.6)$$

We look for a tensor network ansatz which obeys, by construction, to the condition in (6.5). A possible solution has been put forward in [224] and consists in forming a tensor network with *symmetric invariant tensors*, that is, where each tensor composing the network is invariant under the application of the symmetry transformation on each index. To introduce such an object, we label the basis states according to the charge sector ℓ_j they belong to. Moreover, we identify possible different states belonging to the same charge sector with a degeneracy index τ_j , that is, we replace the original state index α_j with a couple of indexes $\alpha_j \prec \{\ell_j, \tau_j\}$. Given a generic tensor $S_{\alpha_1, \dots, \alpha_n} \equiv T_{\{\ell_1, \tau_1\}, \dots, \{\ell_n, \tau_n\}}$ the invariance condition reads, as depicted in Fig. 6.1,

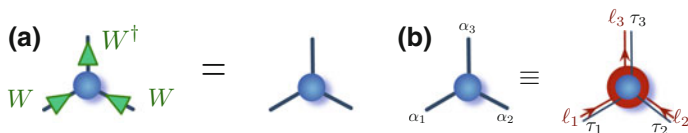


Fig. 6.1 **a** The symmetric tensor condition of (6.7): the tensor remain invariant after the application of the group generators W^\diamond on each index. **b** The class of symmetric tensors satisfying such condition can be represented splitting the index α into a structure index ℓ (containing no variational parameters) and degeneracy index τ (containing variational parameters). The arrows on the charge indexes recall the representation (direct W or inverse W^\dagger) under which the tensor is invariant. The degeneracy indexes can be equipped with an orientation as well via a QR decomposition as described in Chap. 5

$$S_{\{\ell_1, \tau_1\}, \dots, \{\ell_n, \tau_n\}} \stackrel{!}{=} \bigotimes_j W_{\ell_j, \tau_j}^{\diamond_j} S_{\{\ell_1, \tau_1\}, \dots, \{\ell_n, \tau_n\}} = e^{i\phi \sum_j (-1)^{\diamond_j} \ell_j} S_{\{\ell_1, \tau_1\}, \dots, \{\ell_n, \tau_n\}} \quad (6.7)$$

where $\diamond = 1, -1$ specify if the representation on the link has to be inverted or not, such that $W W^{-1} = \mathbb{1}$. Given that the condition in (6.7) has to be fulfilled for every ϕ , it implies that the elements of the tensor S can be different from zero only when $e^{i\phi \sum_j (-1)^{\diamond_j} \ell_j} = 1$, that is, when $\sum_j (-1)^{\diamond_j} \ell_j = 0$. This corresponds to the fact that the sum or irreps on each index has to sum to the identical irrep. In other words, for abelian symmetries, the sum of incoming and outgoing charges has to be zero: the overall charge is conserved passing through the tensor. The condition just stated can be recast in a specific form of the tensor S , separating the structural part dictated by the charge conservation condition, and the *degeneracy tensor* which contains the non-zero elements which can be used to perform variational algorithms. In conclusion, in presence of an abelian symmetry, the a symmetric tensor can be written as

$$S_{\{\ell_1, \tau_1\}, \dots, \{\ell_n, \tau_n\}} \equiv T_{\tau_1, \dots, \tau_n} \delta_{\sum_j (-1)^{\diamond_j} \ell_j, 0}, \quad (6.8)$$

where the *structural tensor* $\delta_{\sum_j (-1)^{\diamond_j} \ell_j, 0}$ is a kronecker delta and imposes the symmetry condition: the tensor can thus be written in block-diagonal form, and each operation acting on it can be done block by block, resulting in an improved performance of all algorithms [31, 225–227].

A simple example might help to clarify the theoretical construction presented above. We consider a tensor network composed of two lattice sites, each of them equipped with a Hilbert space whose local basis $\alpha_j = 0, 1$ labels the number of particles in each lattice site. We can build an MPS for the two sites, composed by two $d \times m$ tensors $S_{\alpha_j, \beta}$ depicted in Fig 6.2a. If we exploit the charge conservation, we first split the global indexes in the charge and degeneracy indices, $\alpha_j < \{\ell_j, \tau_j\}$ and $\beta < \{m, \nu\}$, where $\ell_i = 0, 1$. The symmetric condition implies

$$S_{\alpha_1, \beta} = T_{\tau_1, \nu} \delta_{\ell_1, m}, \quad S_{\alpha_2, \beta} = T_{\tau_2, \nu} \delta_{\ell_2, -m}, \quad (6.9)$$

which, specialized for the sector of zero particles in the system (i.e. $\tau_i = \nu = 0$), results in a single variational parameter for the overall many body state! Although this might appear confusing at first glance, we shall remember that we have constrained the system to be in the sector of global charge zero. Being this equivalent to the set of states with zero particles (composed only by the state $|00\rangle$), it is not surprising

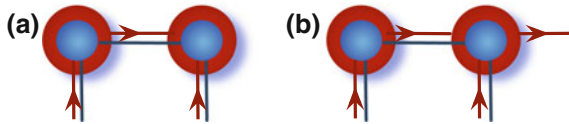


Fig. 6.2 **a** Symmetric MPS of two sites with structural (directed red links) and degeneracy link (black lines) **b** Symmetric MPS with symmetry charge selector (open directed red link)

that the symmetric ansatz results in a single variational parameter: the amplitude of such state (the phase is global and can be removed). However, it would be desirable to have the possibility of changing the overall symmetry sector, for example to study the set of states with a single particle. The selection of different symmetry sector can be promptly done slightly modifying the symmetric tensor network ansatz, that is, adding a link which can work as charge selector, as depicted in Fig. 6.2b. Notice that this structure is equivalent of the first two tensors of an MPS with N particles, i.e., whenever we cut a symmetric tensor network, the resulting free index brings the label of all possible charge sectors (number of particles) present in the partition. In our example, as each lattice site can have at most one particle: in two lattice sites the only possible states are those with overall up to two particles: As for system subsets the conservation of the number of particles does not hold, we shall indeed keep all the possibilities. In conclusion, proceeding as before, one obtains that $S_{\alpha_1, \beta}$ remains unchanged, while the new rank-three tensor for the second lattice site becomes

$$S_{\alpha_2, \beta_1, \beta_2} = T_{\tau_2, \nu_1, \nu_2} \delta_{\ell_2 + m_1 + m_2, 0}. \quad (6.10)$$

It can be easily seen now that the structural tensor imposes that non-zero element appear when $m_2 = 0, 1, 2$ for which $\ell_2 = m_1 = 0, \ell_2 = 1, m_1 = 0$ or $\ell_2 = 0, m_1 = 1$, and $\ell_2 = m_1 = 1$ respectively. That is, the originally 4×4 tensor S is now replaced by a block diagonal tensor with three blocks (one for each charge sector): two non-degenerate and the charge-one sector of degeneracy two. One can keep on this exercise adding more lattice sites and explore the full power of such construction.

Finally, it shall be clear to the reader how to construct a global symmetric tensor network ansatz with charge symmetry selection: an additional open *charge selector link* is added and then used to project over the desired overall system charge sector. We conclude this introduction to symmetric tensor network showing that indeed the construction fulfill the condition of (6.6). In Fig. 6.3 we depict such demonstration: the global pointlike operator is applied to the physical indexes, an identity $\mathbb{1} = WW^{-1}$ is inserted on each internal index of the network and the charge selector index. The invariance of every single tensor under the action of the symmetry group of (6.7) results in the demonstration of the thesis. In the presence of a charge selector, a global phase appears \mathcal{N} being the charge of the sector selected by the charge selector link.

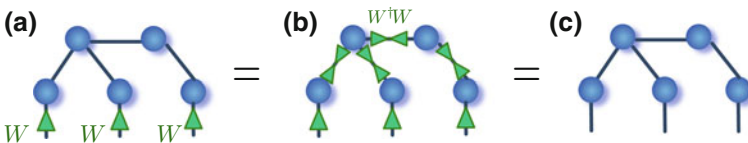


Fig. 6.3 Symmetric tensor networks: **a** The transformation $\bigotimes_j W_{\alpha_j}(g)$ is applied to every physical index of the TN. **b** The identity $W^\dagger W = \mathbb{1}$ is inserted in each internal index of the network and **c** given that each tensor satisfy the symmetric tensor condition of (6.7), the whole tensor network results invariant under the action of the group

We conclude this section stressing the fact that the construction introduced here for Abelian global symmetries, can be readily generalized to non-Abelian ones, and also to combinations of different symmetries for systems presenting more than one global symmetry, as the conservation of the total spin and the total number of particles. However, despite this generalization is somehow straightforward, it becomes quickly highly technical and thus we refer the interested reader to more specialized literature, which should be smoothly addressable after the introduction presented here [224, 228, 229].

6.3 Quantum Link Formulation of Gauge Symmetries

In the previous section, we have seen how we can introduce a tensor network ansatz which, by construction, respect global symmetries. However, there is another very relevant class of symmetries that would be desirable to exploit, that is, *gauge symmetries*. In this section, we define gauge symmetries and show that, similarly to the global symmetries, it is possible to construct a gauge invariant tensor network ansatz. To do that, we introduce briefly the *quantum link formulation* of lattice gauge theories as it can readily be encoded in the tensor network language. The discussion on the comparison between Wilson and quantum link formulation of lattice gauge theories goes beyond the scope of this book, the interested reader can find it in more technical literature [230, 231]. However, since tensor network methods require finite local Hilbert space, we consider it as one of the methods to truncate the local Hilbert space dimensions, an operation that shall be performed by any tensor network approach to lattice gauge theories. Notice that quantum link models in the limit of large representations of the link degree of freedom converge to the standard Wilson representation [91]. Finally, we end the section with a quick overview of the potential application of gauge invariant tensor networks and of other possible approaches to attack this extremely challenging problem. Once more, we specialize our presentation to Abelian symmetries in one dimension and a simple Hamiltonian to get rid, at this introductory stage, from the unnecessary technicalities, and to present the central concepts as clearly as possible. Extension to higher dimensions, more complex theories, and non-Abelian symmetries are possible and, despite being technically more challenging, are based on the ideas presented hereafter.

The main point of lattice gauge theories stems from the promotion of the global invariance of (6.2) to a local one, that is,

$$c_i \rightarrow e^{i\varphi_i} c_i, \quad \varphi_i \in [0 : 2\pi], \quad (6.11)$$

where the phase factor φ depends on the lattice index i and clearly $c_i^\dagger \rightarrow e^{-i\varphi_i} c_i^\dagger$. Notice that now we are requesting the Hamiltonian to be invariant under the action of each unitary representation of the local group $U_i(g)$, not only of the product of all N of them (one for each lattice site). In this sense, we are requesting the system to obey a much more strong constraint with respect to that introduced in the previous section:

not only the total number of particle shall be conserved, also the dynamics at each lattice site is highly constrained. As we will see, this is equivalent to impose Gauss' law in QED (and its generalization in other theories). Consequently, exploiting this extensive number of symmetries can bring to a large gain in terms of numerical efficiency.

Applying the transformation defined in (6.11) to the hopping Hamiltonian of (6.1), one obtains

$$H = -t \sum_i (e^{i\varphi_{i,i+1}} c_i^\dagger c_{i+1} + e^{i\varphi_{i+1,i}} c_{i+1}^\dagger c_i), \quad (6.12)$$

where $e^{i\varphi_{i,i+1}} = e^{i(\varphi_{i+1} - \varphi_i)}$. As one might have expected, the Hamiltonian is clearly not invariant unless $\varphi_i = \varphi \forall i$, that is, in the case of global point-like symmetry studied in the previous section. A possibility to build a simple invariant Hamiltonian is then to add, an object that transforms in such a way that the phases in (6.12) cancel out. This line of reasoning leads to the introduction of new operators, the *parallel transporters* \mathcal{U} which are responsible for the matter-fields coupling in the theories of fundamental interactions [232, 233]. The idea is to define an operator with support on the link between two lattice sites that transforms under the action of the group, for an $U(1)$ gauge symmetry, as

$$U_i(g) \mathcal{U}_{i,i+1} U_{i+1}(g)^\dagger = \mathcal{U}_{i,i+1} e^{-i\varphi_{i,i+1}}; \quad (6.13)$$

and use them to cancel the unwanted phases appearing in (6.12). Recalling that the phases φ in $U(1)$ symmetric systems are related to the charge ℓ on the lattice sites, we can look for their conjugate variable $E_{i,i+1} = -i\partial/\partial\varphi_{i,i+1}$ which obeys to the commutation relations

$$[E_{i,i+1}, \mathcal{U}_{i,i+1}] = \mathcal{U}_{i,i+1}; \quad [E_{i,i+1}, \mathcal{U}_{i,i+1}^\dagger] = -\mathcal{U}_{i,i+1}^\dagger, \quad (6.14)$$

and zero otherwise. It can be shown, that for theories such as QED, the charge ℓ and the field $E_{i,i+1}$ are indeed the electric charge and fields respectively, while for more complex theories such as QCD, the charges are related to the number of quarks present on the link, while the field represents the gauge fields [78, 233, 234].

In the quantum-link formulation of lattice gauge theories the link operators \mathcal{U} and the field operator are given by spin operators $S_{i,i+1}$, such that

$$\mathcal{U}_{i,i+1} \equiv S_{i,i+1}^+; \quad \mathcal{U}_{i,i+1}^\dagger \equiv S_{i,i+1}^-; \quad E_{i,i+1} \equiv S_{i,i+1}^z. \quad (6.15)$$

One can readily show that the commutation relations given in (6.14) are indeed satisfied with this choice. It shall be evident that the link operators \mathcal{U} , \mathcal{U}^\dagger are the raising and lowering operators of one gauge field quanta, whose value is given by the expectation value of S^z . It is now possible to write a gauge invariant Hamiltonian in the quantum link formulation, that is,

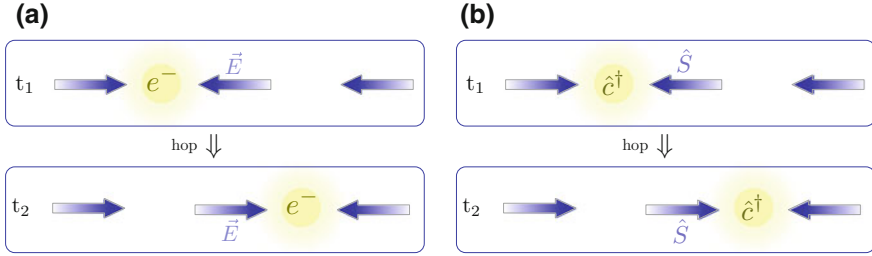


Fig. 6.4 **a** An electron hops from one lattice site at time t_1 to the next one at time t_2 , the electric field between the two sites shall change accordingly. **b** The correspondent process in the quantum link description, when the matter hops from one site to the other, the spin on the link change his state under the action of Hamiltonian (6.16) and in particular of \hat{S}^+ or \hat{S}^- (i.e., for spin one-half it flips sign)

$$H_t = -t \sum_{i,i+1} (c_i^\dagger S_{i,i+1}^+ c_{i+1} + c_{i+1}^\dagger S_{i,i+1}^- c_i); \quad (6.16)$$

where the tunneling dynamics of a matter field is accompanied by a spin flip on the link between the two lattice sites. This is indeed reminiscent of what happens in electrodynamics (either classical or quantum) where, as depicted in Fig. 6.4, if an electron hops the electric field shall change accordingly not to violate Gauss' law. In this discrete one dimensional version, the familiar classical Gauss' law $\rho = \nabla \mathbf{E}$ can be expressed as

$$c_i^\dagger c_i = S_{i+1,i}^z - S_{i,i-1}^z \Rightarrow n_i = \Delta S_i^z \quad (6.17)$$

that is, the number of charges on the link shall be equal to the difference of the electric field entering and exit the lattice site. The condition in (6.17) can be rewritten introducing a local operator $G_i = c_i^\dagger c_i - \Delta S_i^z$ and imposing that its expectation value on each lattice site is identically zero, that is, imposing that the *physical state* has to fulfill the generalized Gauss' law

$$G_i |\psi\rangle = 0. \quad (6.18)$$

All states that do not satisfy (6.18) are unphysical states and form the gauge-variant space that shall be neglected. Finally, notice that the operator G_i are the generator of the group symmetry. Thus, one can reverse the construction presented here and start defining the generators of the symmetry group G_i according to the desired symmetry the theory shall fulfill: starting from (6.18), it is then possible to write the correspondent Abelian or non-Abelian gauge theory [78].

6.4 Lattice Gauge Invariant Tensor Networks

Before presenting the gauge invariant version of tensor networks, we shall make an additional formal step, rewriting the quantum link formulation in terms of other operators as this will be instrumental in the following theoretical construction. The necessary step is to reformulate the spin operator S_i defined on a link in terms of particles that live on the neighbouring lattice sites, which are conventionally referred to as *rishons*. The central idea, is to map a spin degree of freedom on a double well system with particles hopping between them. Once more, an example – depicted in Fig. 6.5 – helps to give the intuitive idea: a spin $1/2$ particle has two possible states, up and down, and they are formally equivalent to the possible states of a single particle hopping in a double-well potential in second quantization formalism. Indeed, we have that

$$\hat{S}_{i+1,i}^+ = \hat{r}_{i,L}^\dagger \hat{r}_{i+1,R}; \quad \hat{S}_{i+1,i}^z = \frac{1}{2}(\hat{n}_{i+1,R}^r - \hat{n}_{i,L}^r); \quad (6.19)$$

where $\hat{r}_{i,\diamond}^\dagger$ ($\hat{r}_{i,\diamond}$) are the rishon creation (annihilation) operators in the left or right ($\diamond = L, R$) well of the double well potential between the i -th and $i + 1$ -th sites and $\hat{n}_{i,\diamond}^r = \hat{r}_{i,\diamond}^\dagger \hat{r}_{i,\diamond}$ the correspondent number operator. The rishon fields are completely arbitrary, i.e. they can be either bosonic or fermionic as they appear always in pairs, ensuring that the gauge fields are bosonic operators. Strictly speaking, the introduction of the bilinear representation in terms of rishons of the parallel transporter \mathcal{U} can be formulated also without introducing the spin representation. Finally, the introduction of the rishons introduces an additional gauge symmetry,

$$\hat{N}_{i,i+1}^r |\psi\rangle = (\hat{n}_{i,L}^r + \hat{n}_{i+1,R}^r) |\psi\rangle = N_{i,i+1}^r |\psi\rangle, \quad (6.20)$$

that is, the total number of rishons $N_{i,i+1}^r$ is conserved in each link: in the example of Fig. 6.5 we have indeed one rishon per link. This additional gauge symmetry will play a major role in the following tensor network formulation.

With the introduction of the rishon representation of the gauge field, we are now ready to start building the gauge invariant tensor network ansatz. We aim to embed



Fig. 6.5 Rishon representation of the gauge degree of freedom: a spin one-half is formally equivalent to a particle (a rishon) that hops in a double well potential. The spin quantization axis can be defined orthogonal to the symmetry axis of the double well potential: the spin pointing to the left (right) represents the occupied left (right) well. Upper diagram is a view from above of the two well potential, used in Fig. 6.6 to graphically depict different spin ice states via rishon states. The generalization to higher spin representation (higher number of rishons per link) can be obtained increasing the number of rishons in the double well

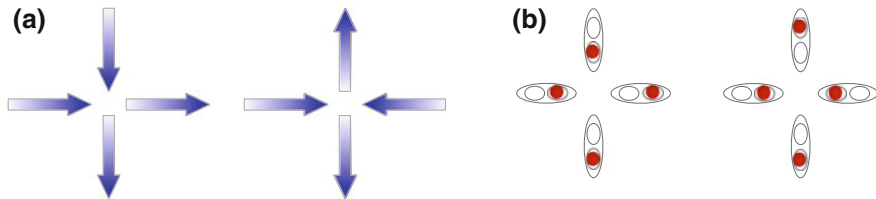


Fig. 6.6 **a** Two possible configurations of spin ice ($S = 1/2$) systems: an equal number of spin points in- and out-ward every lattice site. **b** The same two states for quantum spin ice in their rishon representation, depicted with the same graphical notation as in Fig. 6.5

the gauge constraint of (6.18) exactly in the tensor structure: to achieve this goal we exploit the fact that the spin operators $S_{i+1,i}$ written in terms of the rishons can be split in commuting operators which have support only on disjoint dressed sites. Hereafter, we will explicitly construct such an ansatz for a simple case and then present the general result. The example we consider is a simplified version of the *quantum spin ice* model, a paradigmatic model to study frustrated magnetism [102, 235–237]: It is the quantum counterpart of the classical spin ice, which is defined over a two-dimensional lattice, where a spin lives on each link. Assuming that the spin can have only two possible configurations $\{+, -\}$ (pointing in the two directions of the link), spin ice models look for the lowest energy configuration of a given Hamiltonian (whose particular expression is not relevant here) under the constraint that the number of spins pointing towards each lattice site is equal to the number of spins pointing outwards. Two exemplary states fulfilling such condition are depicted in Fig. 6.6a together with their rishon representation (Fig. 6.6b): it should be clear by now that it introduces a gauge constraint that can be written in the form of (6.18): the magnetic flux through the contour line around every lattice site shall be null. The quantum spin ice is the quantum version of such a model, where the classical variables are replaced with quantum ones, namely spin one-half represented by the standard Pauli matrices. Here, the quantum spins are the physically relevant quantity (they are not a representation of physical gauge fields), however, they play exactly the role of the gauge fields, and everything we said before can be applied straightforwardly. Moreover, notice that differently from (6.16), in this model (and in related ones, e.g., quantum dimer models) there are no matter fields: the Hamiltonian is only a function of spin variables, and thus the gauge invariant tensor network construction is simpler than that necessary in theories with gauge field and matter coupling. However, despite the simplification in the detailed calculations, all the necessary steps to attack the general cases are present. For the same reason, we consider the one-dimensional version of the quantum spin ice, which despite being physically not exciting and possibly of little interest, it is a perfect example for our purposes.

The Gauss law for the one-dimensional spin ice model can be written as

$$G_i|\psi\rangle = (\sigma_{i,i-1}^z + \sigma_{i+1,i}^z)|\psi\rangle = 0; \quad (6.21)$$

indeed, out of the possible four spin configurations of the two spins living on the links connected to the i -th site, only the two configurations $|+, -\rangle$ and $|-, +\rangle$ obeys the condition (6.21). Equivalently, using (6.19) one can write the quantum spin ice Gauss' law in the rishon representation as

$$G_i|\psi\rangle = (n_{i,R}^r - n_{i,L}^r)|\psi\rangle = 0, \quad (6.22)$$

assuming that the total number of rishons in each link $N_{i,i+1}^r$ is not only constant but equal on every link (in our example indeed $N_{i,i+1}^r = 1$). Finally, of all four possible states on the site $|n_{i,R}^r, n_{i,L}^r\rangle \in \{|0, 0\rangle, |1, 0\rangle, |0, 1\rangle, |1, 1\rangle\}$, only the two states $\mathcal{H}_G^1 \equiv \{|0, 0\rangle, |1, 1\rangle\}$ are gauge invariant. Thus, one can define the gauge invariant local basis of the i -th site accordingly, reducing the local Hilbert space dimension d from four to two.

However, when building the composite Hilbert space of two or more sites, one shall take into account the constraint on the total number of rishons per link $N_{i,i+1}^r = 1$. Indeed, when composing two local Hilbert spaces $\mathcal{H}^2 = \mathcal{H}_G^1 \otimes \mathcal{H}_G^1$, four possible states appear but only two satisfy condition in (6.20), namely the gauge invariant space (with respect to the local number of rishons, not the original one defined by (6.18)) is $\mathcal{H}_G^2 = \{|00\rangle \otimes |11\rangle, |11\rangle \otimes |00\rangle\}$. The construction of the gauge invariant space of two sites can be achieved applying the corresponding projector to \mathcal{H}^2 , which can be written as

$$\hat{P}_{N_{i,i+1}^r} = \delta_{n_{i,L}^r + n_{i+1,R}^r, N_{i,i+1}^r}. \quad (6.23)$$

It shall now be clear how it is possible to build quantum link representation of the gauge invariant space of L lattice sites: starting from the standard tensor product of the local Hilbert spaces in the computational basis, the Gauss' law can be enforced applying the projector that selects the gauge invariant states satisfying (6.17). This projector is local in the sense that lives on the left and right rishon Hilbert space with the same index, where the operators $c_{i,R}^r, c_{i,L}^r$ have support. Notice that this is possible only because we split the spin degree of freedom in two independent ones with the constraints given in (6.20). Finally, the application of the projectors $P_{N_{i,i+1}^r}$ enforce the condition (6.20). In the following, we show that this theoretical construction is valid in general and that can be straightforwardly encoded in tensor network language.

The gauge constraint in (6.17) can be imposed keeping only the gauge invariant states among all that generated by the tensor product of the local basis composed by the two rishons $|i_R\rangle, |i_L\rangle$ (and, if present, also the basis of the matter degree of freedom $|i_c\rangle$), that is, the gauge invariant basis can be written as

$$|g\rangle_i = \sum_{i_R, i_c, i_L} K_{i_R, i_c, i_L}^g |i_R\rangle |i_c\rangle |i_L\rangle, \quad (6.24)$$

where the linear operator K_i is an isometry such that $K_i K_i^\dagger = \mathbb{1}$, and $K_i^\dagger K_i = P_{G_i}$ where P_{G_i} is the projector on the gauge invariant subspace for the i -th site. Consequently $K_i P_{G_i} = K_i$, a property that we shall use to conveniently write the tensor

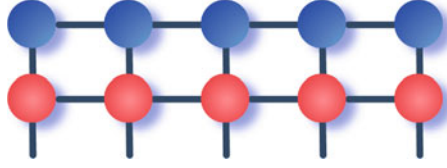


Fig. 6.7 Gauge invariant tensor network: a MPS (blue tensors) contains the variational parameters, while the non-variational MPO $P_{N_r}^G$ (red tensors) constraints the state in the gauge invariant subspace

network ansatz. Similarly, we can enforce the constraint of (6.20) applying the projector given in (6.23) on each link. Of course, to obtain a gauge invariant many-body state in the quantum link representation, we shall apply such operators to every site and links, resulting in the operators

$$K = \bigotimes_i K_i; \quad P = \bigotimes_i P_{G_i}; \quad P_{N_r} = \bigotimes_i P_{N_{r,i+1}}. \quad (6.25)$$

We can enforce the link and the gauge constraint applying first P_{N_r} and then K to a generic many-body wave function, obtaining

$$KP_{N_r}|\psi\rangle = KPP_{N_r}|\psi\rangle = KP_{N_r}P|\psi\rangle = KP_{N_r}K^\dagger K|\psi\rangle = P_{N_r}^G|\psi^G\rangle \quad (6.26)$$

as it can be shown that $[P, P_{N_r}] = 0$, $P_{N_r}^G = KP_{N_r}K^\dagger$, and $|\psi^G\rangle$ is a generic gauge invariant many-body state (i.e. written in the local gauge invariant bases $|g\rangle_i$). We can now introduce the tensor network ansatz: an MPS state with $|g\rangle_i$ to account for $|\psi^G\rangle$ in (6.26) and an MPO (without variational parameters) for the projector $P_{N_r}^G$ as depicted in Fig. 6.7. Moreover, it can be shown that the MPO has a very compact and diagonal representation [59]. Finally, starting from the introduced gauge invariant ansatz, one can straightforwardly apply the machinery introduced in Chap. 5 to perform equilibrium and out-of-equilibrium investigation of lattice gauge theories.

The aforementioned approach has been recently applied to investigate one and two dimensional Abelian and non-Abelian lattice gauge theories, studying in and out of equilibrium phenomena, such the ground state properties, the string breaking, the Schwinger mechanism, and scattering of mesons [55, 59, 79, 238, 239]. Notice that the construction presented here can be generalized with little effort to higher dimensional systems.

We conclude this introductory chapter mentioning that there are alternative formulations of lattice gauge theories compatible with tensor networks. Indeed, for specific group choices, other formulations of discrete gauge theories have also been introduced - see, e.g., [55–77, 114, 240]. In particular cases such as the study of QED in one-dimension (the Schwinger model) it is possible – integrating the out the gauge degree of freedom – to map exactly the lattice gauge theory to a system of long-range interacting spins. This equivalent spin model can be studied exploiting

again tensor network methods, providing also quantitative results at the continuum limit [56, 58, 62–64, 66, 75]. Interestingly, it has also been shown [62] – at least for the Schwinger model – that also at the continuum limit the population of the spin representation of the gauge fields decays exponentially with the square of the charge sector, corroborating the possibility that already with small spin representation of the quantum link formulation, one can correctly describe the main features the low-energy physics of lattice gauge theory.

6.5 Exercises

1. For the Hamiltonian (6.1), check M that $[H, \hat{\mathcal{N}}] = 0$, and find other operators that commutes or not commutes with $\hat{\mathcal{N}}$. Check that $[G, H]=0$.
2. Write explicitly the tensor in (6.10) and extend the construction to a third site. Generalize numerically this construction to L sites.
3. Exploiting the relation (6.26) evaluate numerically the dimension of the gauge invariant Hilbert space of the one-dimensional spin one sector as a function of the number of sites.
4. Compute the operator K and P_{N^r} for the two-dimensional quantum spin ice model.

Part III
Applications

Chapter 7

Many-Body Quantum Systems at Equilibrium



The understanding of the physics of many-body quantum systems at equilibrium encompasses a plethora of processes at the heart of our understanding of nature. Indeed, the properties of materials, of chemical molecules, and lattice gauge theories for high energy physics belong to this class, just to name a few. In this chapter, we present the necessary basis to enter into this fascinating field, mostly from the perspective of a condensed matter physicist, thus focusing on the properties of quantum matter. However, hereafter we do not present a complete introduction to these fascinating topics for which extensive literature exists, see, e.g., [137, 141, 222, 241, 242]: We introduce a useful collection of concepts to start working in the field, in particular having in mind to address this class of problems with TN methods. We first present the basics of phase transitions, quickly reviewing Landau theory of ferromagnetism and the statistical quantum-classical correspondence among system of different dimensionality. Finally, we present local and non-local (entanglement) measures that has been used to characterize such phenomena that can be easily addressed via TN methods.

7.1 Phase Transitions

Everybody is familiar with *phase transitions*, that is, drastic change of matter appearance and properties as a function of temperature: water freezes at zero and boils at hundred Celsius degrees. Less familiar might be other kinds of phase transitions, which occur in magnetic materials [14]. Indeed, experimentally one can verify that below a precise temperature some material sharply exhibits non-zero spontaneous magnetization, even in absence of external magnetic fields. Moreover, the magnetization as a function of the temperature is non-differentiable at the *critical temperature* T_c , where the magnetization appears (see Fig. 7.1a). The discontinuity of the derivative at the *critical point* signals that something drastic is occurring in the system: the

system's microscopic components are rearranging themselves, and a novel collective phenomenon arises, which necessarily calls for correlations spread across the system (all magnetic spins align along the same direction) and, as we will see, the spontaneous breaking of a symmetry occurs.

The first attempt to model such phenomena from a phenomenological and macroscopical point of view has been the Landau theory of ferromagnetism [243, 244]. The idea is to describe the main aspects of the phenomena using an equation of state which respects the symmetry of the problem and relate the two most important thermodynamical quantities describing the system: the free energy F (that shall be minimized to have the system at equilibrium) and the magnetization M , the independent variable of the system. Assuming a power expansion of the function $F(M)$ we obtain

$$F(M) = F_0 + F_1M + F_2M^2 + F_3M^3 + F_4M^4, \quad (7.1)$$

where higher orders are neglected assuming that they are not needed to explain the main features we are interested in. This is a completely arbitrary assumption justified only a posteriori by the success of the theory. We assume that there is no preferred direction, that is, the spins interact with each other with some specific Hamiltonian that cannot distinguish among “up” and “down” direction: the system displays a \mathbb{Z}_2 symmetry (invariant under parity operator). Thus, the odd terms in (7.1) shall vanish, $F_1 = F_3 = 0$. Finally, we assume that $F_4 > 0$ and the quadratic coefficient depends on the temperature (which shall appear somewhere) in the simplest possible way, i.e. $F_2(T) = F_2(T - T_c)$.

The ground state is by definition the state with a magnetization M that minimizes the free energy: the condition

$$\frac{\partial F}{\partial M} = 2M [F_2(T - T_c) + 2F_4M^2] = 0 \quad (7.2)$$

results in the solutions, sketched in Fig. 7.1a,

$$\begin{cases} M = 0 & T > T_c \\ M = \pm \left[\frac{F_2(T - T_c)}{2F_4} \right]^{1/2} & T < T_c \end{cases}, \quad (7.3)$$

which reproduces the typical experimental results.

Despite the tremendous success of this theory to describe the main features of such complex phenomena with an intuitive picture, there are of course limitations to its predictive power. Indeed, the quantitative values of T_c and the other constants shall be fitted from experiments and, more importantly, the square-root scaling law predicted by (7.3) (the first critical exponent we encounter, see Sect. 7.3.1) is typically not matching the measured one. Indeed, the prediction of (7.3) corresponds to the mean-field one, and more sophisticated approaches are necessary to extract the correct scaling values. Among others, TN methods are nowadays routinely applied to attack the problem of computing the different phases of quantum matter and their properties.

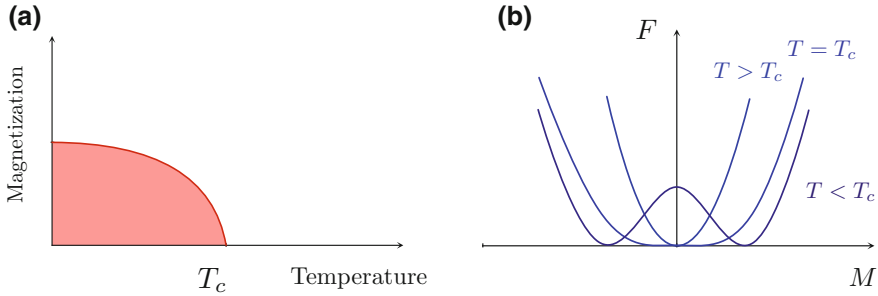


Fig. 7.1 **a** Sketch of the magnetization as a function of the temperature undergoing a phase transition at critical temperature T_c . **b** Landau theory of ferromagnetism, the free energy as a function of magnetization for different temperatures: at $T = T_c$ the stable minima at $M = 0$ becomes unstable, and the free energy becomes a quartic function with two degenerate minima reflecting the \mathbb{Z}_2 symmetry of the problem.

However, before proceeding with the quantum counterpart of phase transitions, it is worth spending some time having a closer look at the potential $F(M)$ as sketched in Fig. 7.1b. Imagine an experiment where the temperature of the system is lowered slowly, starting from a disordered state (no magnetization, the spins are randomly aligned and $T > T_c$): the system lays at the bottom of the single well and $M = 0$. Suddenly, at the critical temperature T_c the minima at $M = 0$ becomes unstable while two other solutions appear for which $M \neq 0$. However, the system still sits in the unstable minima at $M = 0$ until a small perturbation (introduced via the thermal activity of the material), unbalances it and force the system to relax to one of the two new minima, either the positive or the negative one. In conclusion, thermal fluctuations break the symmetry of the problem and allow for the appearance of the spontaneous nonzero magnetization. Finally, notice that once the system has broken the symmetry, it will relax to the new stable minima: this means that once the system underwent the phase transition, it will acquire some rigidity, that is, a resistance to a new change or to restore the original symmetry. For example, crystals (which break translational invariance) do not bend easily, and spontaneous magnetization lasts for an extended period of time (it costs energy to climb up the potential $F(M)$). Similarly, new excitations are present around the new minima, e.g. lattice waves in crystals, spin waves in ferromagnets etc.. Last but not least, as the symmetry breaking is due to thermal fluctuations which acts randomly along all the systems, it might occur that regions far apart break it differently: in such a case, different ordered domains appear, with defects at their interface such as domain walls in magnets, crystals oriented in different directions, etc. We will see in the next chapter how a simple and powerful theory, the Kibble-Zurek mechanism, allows us to describe such phenomena and predict the final density of defect using an elegant argument.

We shall now spend some time to recall that quantum and classical systems are deeply connected from the statistical mechanic's point of view. Thus, also quantum many-body systems display behaviors reflecting what has been introduced here, that is *quantum phase transitions*.

7.2 Quantum-Classical Statistical Correspondence

The deep connection between the classical and quantum worlds in terms of statistical mechanics springs from the fact that the partition function of a D -dimensional quantum system Z_q is equivalent to that of a $(D+1)$ -dimensional classical one Z [245, 246]. As all thermodynamical quantities can be derived from the partition function (for example, the free energy $F = -k_B T \log Z$), phase transitions also occur in quantum many-body systems. The equivalence mentioned above becomes apparent once the *transfer matrix* and the path integral methods to compute Z and Z_q respectively are introduced.

The transfer matrix approach exploits the fact that the classical partition function for N sites can be written as

$$Z_N = \sum_{\alpha_1, \dots, \alpha_N} e^{-\beta_T H(\alpha_1, \dots, \alpha_N)}, \quad (7.4)$$

where the sum runs over all accessible classical states and $\beta_T = 1/k_B T$, and we specialize it to one dimension with periodic boundary conditions, i.e. $\alpha_1 = \alpha_{N+1}$. Assuming nearest neighbour interactions and translational invariance, that is $H = \sum_j H(\alpha_j, \alpha_{j+1})$, the partition function can be rewritten as

$$Z_N = \sum_{\alpha_1} \cdots \sum_{\alpha_N} T_c(\alpha_1, \alpha_2) \cdots T_c(\alpha_{N-1}, \alpha_N) = \text{Tr}(T_c^N) \quad (7.5)$$

where the classical transfer matrix $T_c(\alpha_1, \alpha_2) = e^{-\beta_T H(\alpha_j, \alpha_{j+1})}$ is independent of j and has a (typically small) dimension such that it can be easily diagonalized to obtain its eigenvalues μ_i (we assume $\mu_1 > \mu_2 > \dots$). This condition results in a straightforward computation of the partition function at the thermodynamical limit as μ_1^N plus exponentially small corrections. In conclusion, for example, the free energy per site is

$$\lim_{N \rightarrow \infty} \frac{F}{N} = -\beta_T \log \mu_1. \quad (7.6)$$

The partition function of a quantum system described by the Hamiltonian operator \hat{H}_q , is by definition,

$$Z_q = \text{Tr}(e^{-\beta_q \hat{H}_q}) = \text{Tr}(e^{-\delta\tau \hat{H}_q} e^{-\delta\tau \hat{H}_q} \cdots e^{-\delta\tau \hat{H}_q}), \quad (7.7)$$

where we split the imaginary time in n intervals such that $\beta_q/n = \delta\tau \ll 1$. Indeed, the expression above is equivalent to the solution of time-dependent Schrödinger equation in imaginary time. In the spirit of Feynmann's idea that that the net transition amplitude between two states of the system can be calculated by summing amplitudes for all possible paths between them, we can insert identities (sums over a complete set of basis states, here for convenience labeled as $\mathbb{1} = \sum_{\gamma_j} |\gamma_j\rangle\langle\gamma_j|$) obtaining

$$Z_q = \sum_{\gamma_1} \cdots \sum_{\gamma_n} [e^{-\delta\tau\hat{H}_q}]_{\gamma_1, \gamma_2} \cdots [e^{-\delta\tau\hat{H}_q}]_{\gamma_{n-1}, \gamma_n} = \text{Tr}(T_q^n). \quad (7.8)$$

It shall be evident that (7.5) and (7.8) are formally equivalent, with the only difference that the number of multiplied matrices in (7.8) is not given by the number of elements in the system, but by the number of slices in the imaginary time evolution. This is indeed true in general: the partition function of a quantum Hamiltonian of D -dimensional is equivalent to that of a classical system with $D + 1$ dimensions provided that $\beta_T H = \delta\tau\hat{H}_q$ [245]. The difference between the dimensions comes from the fact that a quantum Hamiltonian is an operator, while for a classical system is a scalar. Thus, for example, the classical Hamiltonian of a chain of spins in an Ising chain is $H = -J \sum_{i=1}^N s_i s_{i+1}$ where the classical variable $s_i = -1, 1$ and the transfer matrix is

$$T_c = e^{\tilde{J}} \begin{pmatrix} 1 & e^{-\tilde{2}J} \\ e^{-\tilde{2}J} & 1 \end{pmatrix} = e^{\tilde{J}} (\mathbb{1} + e^{-\tilde{2}J} \sigma^x),$$

where $\tilde{J} = -J/kT$. For small enough $\delta\tau$, we can write the quantum transfer matrix as $T_q \simeq \mathbb{1} + H_q \delta\tau$, and thus the quantum and the classical partition function are equivalent provided that $n = N$, and $H_q \delta\tau = e^{-\tilde{2}J} \sigma^x$ (neglecting the rescaling factor $e^{\tilde{J}}$). Thus, the quantum partition function of a single spin (dimension $D = 0$) is equivalent to that of a chain of classical spins ($D = 1$). Notice, however, that the time dimension is infinite only in the limit $\beta_q \rightarrow \infty$, $T \rightarrow 0$. Given that a D -dimensional quantum system is statistically equivalent to a classical one in $D + 1$ -dimensions and that classical systems undergoes phase transitions for different temperatures; also quantum systems at $T = 0$ shall undergo phase transitions. However, the driving field will not be the system temperature (it is always zero) but some system's parameter individuated by the relation $\beta_T H = \delta\tau\hat{H}_q$.

7.3 Quantum Phase Transition

Quantum phase transitions, and in particular second-order quantum phase transition on which we will focus on here, describe the drastic changes of the ground state properties of some many-body quantum systems as a function of an external parameter. The role of temperature fluctuations is now played by quantum fluctuations, i.e., the uncertainty introduced by the Heisenberg principle, that is, by the presence of noncommuting terms in the Hamiltonians. As we have seen in the previous section, this shall correspond to the physics of classical phase transitions, where we expect symmetry breaking to occur, the divergence of correlations and in general discontinuous behaviors in the derivatives of the system properties. The characterization of these phenomena has been pursued studying different quantities and their scaling relations with the distance from the critical point (as for 7.3), in time and as a function

of the system size: critical system can be classified by means of the exponents of such scalings, allowing one to introduce *universality classes* which group all systems with the same exponents [137, 141, 241]. Universality classes are formed by systems characterized by the same symmetries, and the low-energy physics of such systems does not depend on the details of the Hamiltonian: the critical properties are dictated only by the symmetries of the Hamiltonian [141, 241]. Also the systems entanglement properties have been used to classify critical systems, providing one of the first strong and unexpected bridges between quantum information theory and strongly correlated systems [140, 247].

We are then interested in the study the ground state properties of a many-body Hamiltonian of the form

$$H(g) = H_0 + gH_1, \quad (7.9)$$

where the system is invariant under a symmetry group, and the competition between the two noncommuting terms H_i determines the system properties. Then, it shall be possible to find an *order parameter*, a local operator whose expectation value on the system ground state $\langle \hat{O}_i \rangle$ (that plays the role of the magnetization in the Landau theory) signals the occurrence of the quantum phase transition at some particular value of the parameter g_c : a change from the disordered phase where $\langle \hat{O}_i \rangle = 0$ to the ordered one where the symmetry is broken and $\langle \hat{O}_i \rangle \neq 0$. The point where this abrupt change occurs is the critical point g_c . Finding the right order parameter to identify a transition is a task which can be straightforward as for the case of the Ising model in the transverse field, or highly nontrivial for more complex theories, requiring experience and skills.

Hereafter, we introduce some of the most common ways used to characterize critical phenomena, with particular attention to those that are readily accessible using TN methods introduced in the previous chapters.

7.3.1 Critical Exponents

As we have seen before, a quantum phase transition is signaled by the expectation value of the order parameter that becomes non-zero in the ordered phase. In particular, the functional dependence in the ordered phase is given by

$$\langle \hat{O}_i \rangle \sim (g - g_c)^{-\beta}, \quad (7.10)$$

where β is the first critical exponent we introduce, and is referred as *spontaneous magnetization*. This quantity is easily accessible by TN simulations, being the expectation value of a local observable. The relation given in (7.10) is exact only at the thermodynamical limit, where the non-analyticity of the function appears.

Moreover, for every finite N , the symmetry breaking could not occur. Indeed, for every finite N , the symmetry-broken ground states have a non-zero coupling term between them, which restore a symmetric superposition of them as the true

ground state. In other words, the ground states are not exactly degenerate for finite N ; thus the symmetric superposition is still energetically favorable, even though the gap closes with N , and especially when describing such phenomena via numerical methods the results might be unstable. A common way to overcome this problem is to artificially break the symmetry, introducing a small field (that eventually will go to zero) to stabilize the computation. However, this procedure introduces another parameter which shall be carefully taken care of, and it does not solve the problem in the limit where the field goes to zero. As we will see later on, the solution to this problem lies in the exploitation of the calculation of the *structure factor*.

As already mentioned, the correlations $\mathcal{C}_{i,j}$ as given in (4.9) diverge at the critical point, meaning that if fitted as an exponential function of the distance between the sites $\ell = i - j$, $\exp(\ell/\xi)$, the correlation length ξ diverges. In other words, the correlation $\mathcal{C}_{i,j}$ has another functional dependence with the distance: a power law decay $\mathcal{C}(\ell) \propto \ell^{-\eta}$, where η is another critical exponent characterizing the system, the *anomalous dimension*. More generally, away from the critical point, the correlation scales as [141]

$$\mathcal{C}(\ell) \propto \ell^{-\eta} \exp(-\ell/\xi). \quad (7.11)$$

The previous relation is practically very important as in numerical investigations, apart from some trivial cases, the location of the critical point is always approximate and thus the correlation length always finite. Employing numerical methods it is possible to compute the correlation functions $\mathcal{C}(\ell)$ and fit the correlation length using the above relation.

Numerical simulations typically work at finite system sizes N (or effectively introduce a cut-off), and this aspect shall also be taken into account when studying critical systems to extract relevant quantities such as the critical exponent β . The solution to these problems comes from our theoretical understanding of critical system and from the self-similarity we expect at all scales, which allow us to introduce scaling relations. In particular, by means of finite-size scaling [141, 248] it is possible to acquire two critical exponents at once: the spontaneous magnetization β and the *correlation length divergence exponent* ν , which describes how the correlation length changes as a function of the distance from the critical point

$$\xi \sim (g - g_c)^{-\nu}. \quad (7.12)$$

In accordance to renormalization group analysis, the order parameter obeys a precise scaling with the system size L and the parametric distance from the critical point

$$\langle \hat{O}_i \rangle(g) \simeq L^{-\beta/\nu} f((g - g_c) \cdot L^{1/\nu}), \quad (7.13)$$

where f is a non-universal function, depending on the microscopic details of the model. Given that the relation (7.13) is valid, after computing the order parameter for different system sizes $\langle \hat{O}_i \rangle(g, L)$, one can first tune the unknown $\gamma_1 = \beta/\nu$ until all curves $\langle \hat{O}_i \rangle(g, L) \cdot L^{\gamma_1}$ cross in a single point, which individuate the critical point g_c (the only point invariant for different L , according to (7.13)). Then, rescaling the

curves $\langle \hat{O}_i \rangle (g_c + (g - g_c)L^{\gamma_2}, L) \cdot L^{\gamma_1}$ by means of $\gamma_2 = -1/\nu$ until all collapse onto one another. The two values γ_1, γ_2 uniquely defines the critical exponents ν, β [248].

Another important critical exponent is the *dynamical-scaling* exponent z which gives the scaling of the imaginary-time correlations with the distance from the critical point of an operator evaluated at different imaginary-time in $\mathcal{C}(\tau) = \langle \hat{O}(\tau) \hat{O}(0) \rangle \sim \exp(-\tau/\xi_\tau)$,

$$\xi_\tau = \xi^z \sim (g - g_c)^{-\nu z}. \quad (7.14)$$

Going back to the quantum-classical correspondence, if the system can be mapped to a homogeneous quantum system in $D + 1$ dimensions, then the time-direction of the Feynman integral in (7.8) is driven by the same Hamiltonian than in the space directions. That is, the correlations in space and time are equivalent, and $z = 1$. However, this is not always the case and in general $z \neq 1$. We also mention that using the analytical continuation, one can also infer on the real-time correlations of the system [245].

Finally, an important relation between correlations in imaginary time and the gap of the system is worth mentioning: indeed writing the evolved operator in the Heisenberg picture $\hat{O}(\tau) = e^{H\tau} \hat{O}(\tau) e^{-H\tau}$, it is easy to show that the correlations are given by

$$\mathcal{C}(\tau) = \sum_j e^{-(E_j - E_0)\tau} |\langle 0 | \hat{O}(\tau) | j \rangle|^2; \quad (7.15)$$

where we inserted a set of complete of system's eigenfunctions $|j\rangle$ and E_j are the corresponding eigenenergies. It shall then be clear that a finite principal gap $\Delta = E_1 - E_0$ implies that for (long) imaginary time the correlations decay exponentially, i.e. $\mathcal{C}(\tau) \sim e^{-\Delta\tau}$. Thus, on the contrary, if the correlations decay algebraically, the gap shall be vanishing, as it occurs at the quantum critical point. The expression above together with (7.14) allows relating the scaling of the gap around critical points with the critical exponents

$$\Delta \sim (g - g_c)^{\nu z}. \quad (7.16)$$

7.3.1.1 Structure Factor

As mentioned in the previous section, whenever one works on quantum systems with finite-size, spontaneous symmetry-broken phase does not occur. It is possible to overcome this problem by exploiting an order parameter definition insensitive to the symmetry breaking. For example, one can adopt the *square root of the structure factor density*, precisely

$$\mathbf{S}(k, g, L) = \sqrt{\frac{1}{N^2} \sum_{i,j} e^{ik(i-j)} \langle \psi(g, L) | \hat{O}_i \otimes \hat{O}_j | \psi(g, L) \rangle}, \quad (7.17)$$

where $|\psi(g, N)\rangle$ is the many-body ground state calculated at g and size N . For example, if applied to study a standard ferromagnetic transition, instead of computing the order parameter based on $\hat{O}_i = \sigma_i^x$, which will be always zero at finite sizes, one can evaluate the structure factor at $k = 0$: it can be shown to coincide to the standard ferromagnetic order parameter $\bar{m} = N^{-1} \sum_j \langle O_j \rangle$ exactly at the thermodynamical limit. Indeed, given that by definition $\langle \hat{O}_i \hat{O}_j \rangle = \langle \hat{O}_i \rangle \langle \hat{O}_j \rangle + \mathcal{C}(\ell)$, then the correlation function $\mathcal{C}(\ell)$ becomes irrelevant towards quantity in (7.17). In fact, we have that either $\mathcal{C}(\ell)$ decays to zero either exponentially (gap different from zero) or algebraically (critical point). In both cases, we have

$$\left| \sum_{i,j}^N \frac{\mathcal{C}(\ell)}{N^2} \right| \leq \frac{1}{N} \int_0^\infty |\mathcal{C}(\ell)| dx \rightarrow 0. \quad (7.18)$$

In conclusion, we have that

$$\mathbf{S}(0, g, N) = \sqrt{\sum_{i,j}^N \frac{\langle \hat{O}_i \rangle \langle \hat{O}_j \rangle}{N^2}} \simeq \sqrt{\bar{m}^2} = |\bar{m}|, \quad (7.19)$$

which shows that $\mathbf{S}(0, g, N)$ is equal to the ferromagnetic order parameter without suffering from finite-size symmetry breaking issues, since it is based on two-point correlation measurements and not on local observations.

7.4 Entanglement Measures

As we have seen in Chap. 5, the von Neumann entropy of the singular values of a system bipartition plays a fundamental role in determining the efficiency of TN methods. Hereafter, we review another important relation between entanglement measures and the physics of the system of interest. The von Neumann entropy is only one of the possible measures of entanglement that can be used to characterize many-body quantum systems [140]: One of the first attempts to connect entanglement and physical properties of the system, in particular in systems undergoing a quantum phase transition, has been made using the concurrence, a commonly used entanglement measure of two qubits [247, 249]. Given the reduced density matrix of two spins one-half embedded in a larger system $\rho_{i,j}$, it is possible to calculate the concurrence C diagonalizing the matrix $R = \sqrt{\rho_{i,j}} \tilde{\rho}_{i,j} \sqrt{\rho_{i,j}}$, and computing

$$C = \max(\lambda_1 - \lambda_2 - \lambda_3 - \lambda_4, 0), \quad (7.20)$$

where $\lambda_1 \geq \dots \geq \lambda_4$ are the ordered squared root of the eigenvalues of R and $\tilde{\rho}_{i,j} = (\sigma^y \otimes \sigma^y) \rho_{i,j}^* (\sigma^y \otimes \sigma^y)$. It has been shown that the derivative with respect

to the parameter g of the concurrence between the two spins diverges exactly at the critical point, signaling the critical behavior and thus working as a non-local order parameter. A posteriori, this beautiful result can be easily interpreted, as the elements of the reduced density matrix are given by the two-point correlations in the systems [140], and thus, their divergence at the critical point is reflected in the concurrence. This observation has been one of the first witnesses of the deep relation between quantum information science and condensed matter theory and has triggered enormous research activities [140]. In particular, it is nowadays widespread the use of the von Neumann entropy of a system bipartition to characterize critical systems. Given a system bipartition in two subsystems A and B , the von Neumann entropy is defined as

$$S = -\text{Tr}(\rho_A \log \rho_A) = -\text{Tr}(\rho_B \log \rho_B) = -\sum_i p_i \log(p_i), \quad (7.21)$$

where ρ_A, ρ_B are the reduced density matrices of the subsystems and p_i their eigenvalues. As can be recalled from (5.43) and (5.44), this quantity – and its scaling with the system size for bipartitions of one-dimensional quantum systems – can be readily be computed using MPS. In the next section, we review some important results which relate the von Neumann entropy scaling and physical properties of the system.

7.4.1 Central Charge

The main feature of the scaling of the von Neumann entropy of a one-dimensional system is that it can be related to the central charge c of the correspondent conformal field theory [140, 242, 250]. Conformal field theory gives a classification of one-dimensional critical quantum systems based on symmetry considerations, independently from the detailed Hamiltonian of the system. The central charge – defined by the conformal anomaly of the Virasoro algebra – plays a central role being one of the unique elements which identify the conformal theory [242] and can be used as the critical exponents to classify the system of interest in a universality class.

For a discrete system with open boundary conditions divided in two parts of size ℓ and $N - \ell$ respectively with a single boundary, the scaling of the von Neumann entropy of the reduced density matrices reads

$$S = \frac{c}{6} \log \left(\frac{N}{\pi} \cdot \sin \frac{\pi \ell}{N} \right) + c'. \quad (7.22)$$

where c' is a non-universal constant [140, 250]. For periodic boundaries, or more generally in the case of a bipartition of the system with two boundaries, the scaling above is doubled, taking into account the two boundaries which contribute to the correlations between the two system's partitions. More generally, the von Neumann

entropy can also be used to measure the correlation length of the system ξ , as slightly away from criticality, it scales for $\ell \rightarrow \infty$ as

$$S \sim \frac{c}{6} \log(\xi). \quad (7.23)$$

An extension to the finite temperature case is also available, where the effective temperature $\beta_T = 1/(k_B T)$ introduces a cutoff to the correlation length and thus plays the role of an effective system size (even at the thermodynamical limit), resulting in a scaling of the form [140]:

$$S = \frac{c}{6} \log\left(\frac{\beta_T}{\pi} \cdot \sin \frac{\pi \ell}{\beta_T}\right) + c'. \quad (7.24)$$

It is worth mentioning that the above results are still valid in the case of disordered systems, provided that the proportionality factor – the central charge – is renormalized in a system-dependent way [140]: for example, for a spin one-half Ising model in transverse field with uncorrelated disorder $\tilde{c} = c \ln 2$ [251]. However, the renormalization depends also on the correlations of the disordered terms, and eventually, correlated noise might result again in the clean case scenario [252–255].

Finally, the relations above can be extended to Rényi entropies,

$$S_q = \frac{1}{1-q} \log(\text{Tr}(\rho_A^q)), \quad (7.25)$$

where the von Neumann entropy is the special case $q \rightarrow 1$. Indeed, in this case we have that (7.22) shall be corrected by the factor

$$S_q = \frac{c}{6} \left(1 + \frac{1}{q}\right) \log\left(\frac{N}{\pi} \cdot \sin \frac{\pi \ell}{N}\right) + c'. \quad (7.26)$$

7.4.2 Topological Entanglement Entropy

We conclude this very concise overview of the use of von Neumann entropy to extract relevant information on the physics of correlated quantum system introducing briefly the concept of *topological entanglement entropy* [37, 140, 242, 256, 257]. Topological systems do not break any symmetry; thus they cannot be characterized by a local order parameter. The ground state is generally not unique, and on a two-dimensional surface, the degeneracy grows exponentially with the number of holes in the surface (genus). Local excitations on top of the degenerate ground states are protected by large energy gaps, making the topological state robust against local perturbations [242]. Finally, topological phases cannot be distinguished by local observations, and thus a global characteristic shall be used such as the entanglement

between system bipartitions [256, 257]. In particular, it has been shown that the Von Neumann entropy can be used to characterize the topological order of the system: it obeys an area law, i.e., scaling with the boundary of the size bipartition L

$$S = \alpha \cdot L - \gamma \quad (7.27)$$

where α is a nonuniversal constant but $\gamma = \log D \geq 0$, the total quantum dimension of the theory, characterizes the topological content of the state. For example, in the case of discrete gauge theories, D is simply the number of elements in the gauge group, while for a Laughlin state in a fractional quantum Hall system with filling factor $\nu = 1/r$ where r is an odd integer, $D = \sqrt{r}$. Despite the challenges still present in applying TN methods to two dimensional systems, in the last years, they have been used to extract the topological entropy of different systems, see, e.g., [258–262].

7.5 Exercises

1. Compute the mean field critical exponents of the one-dimensional Ising model in transverse field via tensor networks (i.e., using $m = 1$) and confirm the analytical predictions.
2. Compute the ferromagnetic order parameter of the one-dimensional Ising model in transverse field both directly and using the structure factor. Compare the results.
3. Perform a finite size scaling to compute the critical exponents of the critical Ising model in transverse field (compute the order parameter for different size and values of the transverse field) both with RG and DMRG and compare the results.

Chapter 8

Out-of-Equilibrium Processes



Out-of-equilibrium processes are ubiquitous; they surround us and life is possible because of temporary out-of-equilibrium fluctuations of the otherwise equilibrium (dead) state. Indeed, any system in the real world is in contact with the rest (the environment) and subject to changing conditions (temperature, interaction strength, etc.) which drive the system out of its equilibrium state. Moreover, any useful process, from information processing to energy harvesting, is based on the fact that a system is driven out of equilibrium and goes into another (possibly equilibrium) state. This is true also at the quantum level and, as we will see in this chapter, it is of tremendous interest to be able to study systems that change their state and to understand when this happens, how this can be efficiently driven, and to predict the result of such transformation. As we have seen, this is now possible in many scenarios for one-dimensional many-body quantum systems using TN methods introduced in the previous chapters.

In particular, the typical scenario is that of a system driven from a somehow trivial state to a more complex one. Many different scenarios that have been recently subject to intense studies can be seen as the change of a system state, from a simple, initial, given one to a more complex but significant state. The first example of such processes is the preparation of most experimental setups: typically the system is found in some equilibrium state from which another more interesting state has to be reached. A relevant instance of this scenario is any quantum information protocol and, in particular, a quantum computation: a register of qubits has to be driven from a trivial state (e.g., $|00000\rangle$) to a state which encodes the problem solution. Similarly, a quantum simulation is nothing else than a system evolving from an initial state into another less trivial one, for example from an insulating state to a superconducting one. This process is also an instance of a more general class of processes, that is, the crossing of a quantum phase transition (an adiabatic quantum computation). As most of the hard problems in computer science can be recast in an adiabatic computation form [263] and adiabatic computation is equivalent to the circuit-base one [264], their study is of utmost importance. Finally, all these transformations can also be

studied in the presence of an environment, and, in particular, it has been shown that it is possible to engineer the environments to achieve interesting states [265–267].

In this chapter, we introduce the adiabatic theorem on which adiabatic computation is based on. Then, we present the Kibble-Zurek mechanism, a powerful theory that allows predicting, under very general assumptions, the error introduced when the adiabatic condition is slightly violated. We present some approaches to go beyond the adiabatic one, and we briefly review some results on out-of-equilibrium open quantum systems. Finally, we briefly mention some other directions which have been recently explored, related to problems in high-energy physics, open systems and complexity.

8.1 Adiabatic Quantum Computation

In this section, we introduce the *adiabatic quantum computation* or *quantum annealing* or. We start presenting the adiabatic theorem, the theoretical argument that enables the adiabatic quantum computation approach. We then present the simplest scenario where the adiabatic theorem can be applied, the Landau-Zener crossing and end the section with an overview of the possible applications of such method to study the quantum matter and classical hard problems.

8.1.1 Adiabatic Theorem

The adiabatic theorem states that if a time-dependent Hamiltonian $H(t)$ varies slow enough and the system is prepared in an eigenstate of the initial Hamiltonian $|\psi(0)\rangle = |E_i(0)\rangle$ (e.g. the ground state), then the system remains in the correspondent time-dependent eigenstate for the whole evolution, i.e. $|\psi(t)\rangle = |E_i(t)\rangle$, $\forall t$. The idea on which the adiabatic theorem is based is highly intuitive and is not different from a classical one: if the system is in its ground state (e.g., a particle at the minimum of a potential), slowly varying the conditions (moving the potential or deforming it) will not excite it.

A simple intuitive derivation of the above theorem can be obtained as follows [268]: given the unitary transformation $U(t)$ that diagonalizes at every time the Hamiltonian, $U^\dagger(t)H(t)U(t) = D(t)$, the time-dependent Schrödinger equation (3.32) can be rewritten in the diagonal basis $|\phi(t)\rangle = U^\dagger(t)|\psi(t)\rangle$ as

$$i\hbar \frac{\partial |\phi(t)\rangle}{\partial t} = D(t)|\phi(t)\rangle - i\hbar U^\dagger(t) \frac{\partial U(t)}{\partial t} |\phi(t)\rangle; \quad (8.1)$$

that is, the net effect of the time-dependent basis we have introduced is the second term on the right hand side, appearing as a correction to the Schrödinger equation. However, if the Hamiltonian is slowly varying, also the change of basis $U(t)$ will

be only slowly time-dependent, and thus its derivative will be small. The adiabatic approximation consists precisely in neglecting the second term in the right hand side of (8.1), which results in a set of decoupled differential equations for the amplitude of probabilities of each eigenvector. They remain constant in time with only a varying time-dependent phase: if the system starts in a single eigenvector, its occupation probability will remain constantly one, i.e. if the system starts in $|E_0(0)\rangle$, it will remain in $|E_0(t)\rangle \forall t$.

Although the result presented above is already very powerful, clearly an important piece of information is missing: one should quantify what “slowly” means and possibly compute the perturbative corrections to the adiabatic approximation. This analysis can be performed [269], resulting in a bound for the probability of begin away from the ground state after starting in the initial ground state $p(t)$ of

$$p(t) \leq 4 \sum_{k \neq 0} \max_{s' \in [0, s(t)]} |a_{k0}(s')|^2 \delta^2 + \mathcal{O}(\delta^3), \quad (8.2)$$

where the Hamiltonian $H(s(t))$ changes in time according to a parametric function $s(t)$, such that $ds/dt = \delta \cdot v(s(t))$, and δ sets the scale of parameter variation. The transition amplitudes from the ground state to the k -th ($k > 0$) normalized instantaneous eigenstate are given by

$$a_{k0}(s) = \hbar \frac{\langle \tilde{E}_k(s) | dH/ds | \tilde{E}_0(s) \rangle}{\Delta_k(s)^2}, \quad (8.3)$$

where $\Delta_k(s) = E_k(s) - E_0(s)$ and we have set all eigenvectors free phases such that the final accumulated Berry phases are zero [269]. In conclusion, parametrizing the Hamiltonian change as

$$H(s(t)) = s(t)H_0 + (1 - s(t)) H_1 \quad (8.4)$$

with $s(0) = 0$ and $s(T_f) = 1$, one can show that the process is adiabatic (error probability $p(t) < \delta^2$) if

$$a(s)v(s) \leq 1, \quad (8.5)$$

where $a(s)$ can be bounded as

$$a(s) \leq 2\hbar \frac{\|H(T_f) - H(0)\|}{\min_s (\Delta_1(s))^2}; \quad (8.6)$$

which results in an adiabatic running time of

$$T_{ad} \propto \frac{1}{\min_s (\Delta_1(s))^2}. \quad (8.7)$$

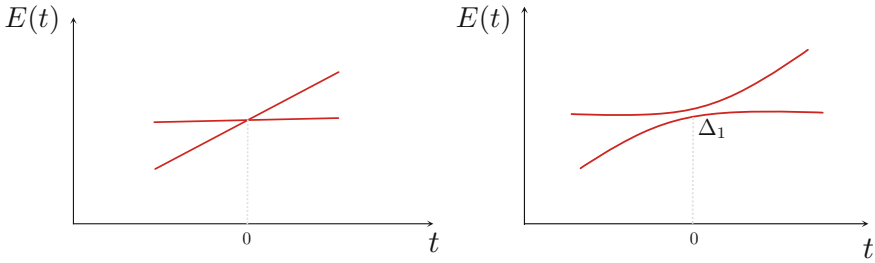


Fig. 8.1 Level crossing for $V = 0$ (left) and for $V \neq 0$ (right) in the Landau Zener crossing defined by the Hamiltonian in (8.8)

The previous expression is only an upper bound for the timescale of an adiabatic process: indeed, it is possible to adapt the velocity to the instantaneous gap performing a global adiabatic optimization that allows for a speed up [270, 271].

8.1.2 Applications

The simplest scenario where the adiabatic theorem can be applied is a paradigmatic scenario, the *Landau-Zener crossing* [272, 273], that is, the computation of the final excitation probability in a two level quantum system with a time-dependent diagonal term and constant coupling of the form

$$H(t) = H_0 + H_1(t) = \begin{pmatrix} E_1 & V \\ V & E_0(t) \end{pmatrix} \quad (8.8)$$

where $E_0(t) = E_1 + vt$ for $t \in [-T; T]$. Figure 8.1 sketches the time dependence of the energy levels of such system, where easily it can be shown that the minimal gap is given by $\min_t \Delta_1(t) = 2V$. Thus, the adiabaticity condition can be expressed as

$$2V^2/\hbar v \gg 1. \quad (8.9)$$

Independently, the Landau-Zener formula predicts the final excitation probability after the crossing as

$$p = \exp \left[-\frac{2\pi V^2}{\hbar v} \right], \quad (8.10)$$

which is in perfect agreement with the condition expressed in (8.9): if the adiabaticity condition holds, the argument of the exponential is large and correspondingly the excitation probability p is small, that is, an adiabatic crossing occurred.

More generally, the adiabatic condition of (8.5) plays a crucial role in adiabatic quantum computation or simulated quantum annealing [274–276]. The scenario is

similar to that presented above: a system is prepared in the ground state of an initial Hamiltonian H_0 , then a competing term H_1 is switched on while the initial one is switched off, as in (8.4). The idea is that while H_0 is a Hamiltonian whose ground state is easy to prepare or compute, the ground state of the Hamiltonian H_1 encodes the solution of a difficult problem. If the process is adiabatic, the crossing defined by (8.4) returns the desired solution. In particular, it can be shown that this computation architecture is equivalent to universal quantum computing [264], and for example, that Grover's algorithm to search in unstructured databases can be recast in this language [270]. Moreover, it can be shown that most hard problems in computer science can be recast as an adiabatic quantum computation [263]. An example will serve to clarify this last point: the *partitioning problem* is an NP-complete problem defined starting from a set of numbers n_1, \dots, n_N . The question is if it is possible to bipartite the set of numbers in two sets of equal values. The adiabatic quantum computation version of such simple to state but extremely hard to solve problem stems from the definition of the Hamiltonian $H_1 = (\sum_i n_i \sigma_i^z)^2$, where the n_i form the set of numbers to be bipartited and σ_i^z the standard Pauli matrices. Defining, e.g., $H_0 = \sum_i \sigma_i^x$ and performing the adiabatic protocol one could find the energy E_0 of the ground state of H_1 : if it is zero, which lower bounds all possible eigenenergies given that H_1 is a positive operator, the solution to that particular partitioning problem exists and can be read from the ground state. If instead $E_0 > 0$, the partition with equal values does not exist, but the mismatch has been minimized, solving an NP-hard problem.

However, what seems to be a strong attack strategy to extremely hard problems has a flaw: the adiabatic condition cannot be satisfied in the thermodynamical limit, as typically the minimal gap goes to zero either polynomially or even exponentially fast. That is, in the limit of large problem size, errors are unavoidably introduced hindering a clean solution to the problem. Nevertheless, huge efforts to exploit this strategy to the maximum have been performed, and active research both in numerical methods and in developing quantum annealers which could perform such process in the lab is under thriving activity.

Finally, another potentially huge field of application of these ideas is the crossing of a quantum phase transitions: the two competing Hamiltonians are now given by the physics of the system as in (7.9), and the parameter g is tuned in time, playing the role of the function parametrization $s(t)$. These processes naturally arise in many scenarios, in quantum science and condensed matter (e.g., during the loading of atoms in optical lattices, in superconductors or magnetic materials, etc.). Again, at the thermodynamical limit, the adiabatic condition cannot be satisfied due to the scaling of the critical gap given in (7.16). However, in this case, the structure of the problem and in particular its classification by means of the universality classes of the quantum phase transition allow us to predict the error introduced by the violation of the adiabaticity through a simple – yet extremely powerful – theory that we introduce in the next section.

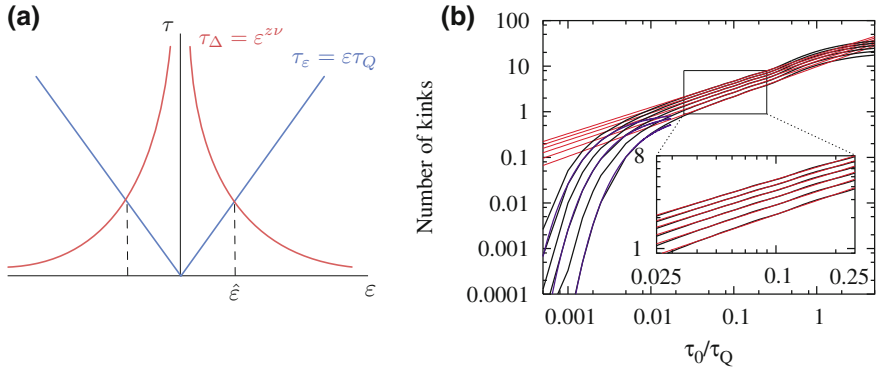


Fig. 8.2 **a** The Kibble-Zurek mechanism: equating the two typical system's timescales ($\tau_\varepsilon = |\varepsilon\tau_Q|$ and $\tau_\Delta = |\varepsilon(t)|^{-\nu z}$) it is possible to estimate the distance from the critical point where the system freezes $\hat{\varepsilon}$, the end of the adiabatic dynamics. **b** Verification of the Kibble-Zurek mechanism in the quantum Ising model in transverse field (from [277]): the correlation length (proportional to the inverse of the number of kinks) as a function of the inverse of τ_Q (black lines, different lines reports results for different system sizes) scales as predicted by (8.13) (red lines). The blue lines follow an exponential scaling, signaling the set up of the Landau-Zener scaling of (8.10)

8.2 Kibble-Zurek Mechanism

The Kibble-Zurek mechanism has been put forward to predict the final residual excitation energy on top of the ground state when crossing a quantum phase transition in finite time. Indeed, when crossing a gapless critical point of $H(t)$, for any finite time, the adiabatic theorem is violated, and the final excitation probability will be non-zero and, consequently, a finite density of defects in the ordered phase will be produced [278–280]. The Kibble-Zurek mechanism provides a scaling argument for the final density of defects based on the assumption that the dynamics can be divided in either adiabatic or impulsive, according to the distance from the critical point. Its setup follows that introduced in the previous section, where a system starts in a ground state of an initial Hamiltonian which is modified in time according to (8.4), with the additional constraint that the quench is performed linearly with a typical time-scale τ_Q . In summary, we assume

$$H(t) = \varepsilon(t)H_0 + H_1, \quad \varepsilon(t) = t/\tau_Q, \quad t \in [-\tau_Q/2 : \tau_Q/2]; \quad (8.11)$$

such that $\varepsilon(t) = g(t) - g_c$ determines the distance from the critical point g_c and the crossing occurs at $t = 0$. As we have seen in the previous chapter, at the critical point the correlation length diverges as in (7.12), and near the critical point the gap scales with the distance as $\Delta \sim (g - g_c)^{\nu z}$ [see (7.16)]. The system's gap introduced a natural time scale for the system relaxation time, i.e., the time needed to react to changes. For example, the variation of the distance to the critical point corresponds to a change of the correlation length. This typical timescale is nothing more than

$\tau_\Delta \sim 1/\Delta = |g(t) - g_c|^{-\nu z} = |\varepsilon(t)|^{-\nu z}$. Note that τ_Δ diverges at the critical point and the thermodynamical limit, that is, it is increasingly harder for the system to adapt to the new condition (increasing its correlations). This important time scale has to be compared with the other process timescale, that is, the rate of change of the order parameter dictated by the quench time scale. The inverse of the relative rate of change of $\varepsilon(t)$ is $\tau_\varepsilon = |\varepsilon/\dot{\varepsilon}| = |\varepsilon\tau_Q|$. Thus, depending on the relation between these two timescales, different dynamics will occur:

$\tau_\Delta \gg \tau_\varepsilon$: the system can easily relax to the new ground state while the system parameter is changed. Thus, it remains in the system instantaneous ground state.

$\tau_\Delta \ll \tau_\varepsilon$: the system is not able to relax to the new ground state while the system parameter is changed, that is, it remains frozen in its initial state while $\varepsilon(t)$ evolves.

The second condition will clearly appear at some point given the divergence of τ_Δ : once again this is equivalent to stating that the adiabatic condition is violated and thus different ordered domains (of dimension given by the correlation length) will appear, e.g., forming different crystals or magnetic domains. The Kibble-Zurek mechanism is based on the assumption that the whole dynamics can be approximatively described as either adiabatic or frozen. If so, one can individuate the parameter value $\hat{\varepsilon}$ where the system changes from one to the other simply equating the two main timescales. A graphical representation of this comparison is represented in Fig. 8.2a and results in

$$\hat{\varepsilon} \sim \tau_Q^{-\frac{1}{1+\nu z}}. \quad (8.12)$$

While approaching a critical point from negative ε , the system will then be in the ground state until $-\hat{\varepsilon}$, while it will be frozen (i.e. it will not change) after that until $\hat{\varepsilon}$. Thus, around criticality, the system will achieve the maximal correlation length corresponding to that at the ground state in $-\hat{\varepsilon}$ given, using (7.12), by

$$\hat{\xi} \sim \tau_Q^{\frac{\nu}{1+\nu z}}. \quad (8.13)$$

This powerful expression allows predicting the correlation length that can be achieved driving a system at criticality in finite time only as a function of the critical exponents of the theory. The KZ mechanism has been tested in a variety of quantum toy models at zero temperature, including ordered and disordered systems, as well as for crossing isolated or extended critical regions [277, 281–296]. We report in Fig. 8.2b the verification of the Kibble-Zurek mechanism in the Ising model in transverse field, obtained by means of DMRG simulations [277].

8.2.1 Crossover from Quantum to Classical Kibble-Zurek

The Kibble-Zurek scaling has been verified in many scenarios. However, its experimental verification in strongly correlated many-body quantum systems at zero tem-

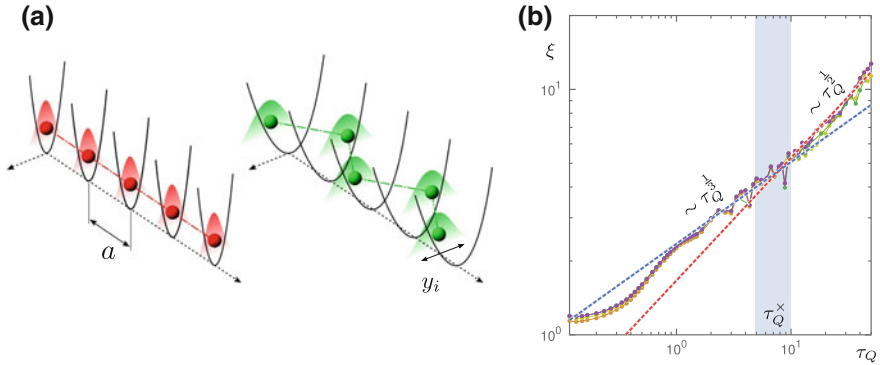


Fig. 8.3 **a** Linear to zig-zag quantum phase transition (from [297]): N ions at fixed distance a are trapped in a linear chain (for which $\langle \sum_i (-1)^i y_i \rangle = 0$, left) for tight confinements, while releasing it beyond a critical threshold results in a zig-zag configuration of minimal energy ($\langle \sum_i (-1)^i y_i \rangle > 0$ right). The system has a \mathbb{Z}_2 symmetry, thus it belongs to the Ising universality class. **b** Verification of the Kibble-Zurek mechanism in the ϕ^4 model (from [286]): final correlation length ξ as a function of the quench timescale τ_Q (full circles, different colours denoted different system sizes). For $\tau_Q < \tau_Q^x$ the scaling is classical $\xi \propto \tau_Q^{1/3}$ (blue dashed lines), while for $\tau_Q > \tau_Q^x$ the quantum scaling is found, $\xi \propto \tau_Q^{1/2}$ (red dashed line)

perature remain somehow elusive. Indeed, to achieve a clean agreement between the measured and the predicted scalings very stringent experimental constraints have to be satisfied [289, 292, 293, 298–300]. For example, in a beautiful experiment with trapped ions [288], it has been shown that the scalings found experimentally are those of the classical theory describing the system, and not the quantum ones. Also stimulated by these findings, a full characterization and simulation of the system has been performed in [297], exploiting the fact that trapped ions undergoing the linear to zig-zag structural quantum phase transition (represented in Fig. 8.3a) can be modelled under pretty general assumptions as a ϕ^4 model [301, 302],

$$H = \sum_i p_i^2 + \varepsilon(t)y_i^2 + y_i^4 + (y_{i+1} - y_i)^2; \quad (8.14)$$

where the sum runs over all ions in the chain, y_i , p_i are the transverse displacement (whose expectation value $\langle \sum_i (-1)^i y_i \rangle$ is the order parameter for the transition) and its conjugate canonical momenta such that $[y_i, p_j] = i\hbar\delta_{i,j}$. The parameter $\hbar = \hbar/\sqrt{E_0 m a^2}$ plays the role of an effective \hbar , with a being the lattice constant, m the particle mass, and E_0 the typical pairwise interaction energy. It is then possible to simulate its dynamics while crossing the quantum phase transition employing TN methods [286]. It has then put forward the idea that the classical scalings appear because the system freezes far away from the critical point, in the region where quantum fluctuations are still negligible. This region can be individuated by means of the Ginzburg criteria [303, 304], and used to predict where the crossing between

classical and quantum scalings occurs, that is

$$\tau_Q^\times \sim \frac{\hbar}{\varphi} |\varepsilon|^{-1-z\nu}, \quad (8.15)$$

where φ is determined by the scaling of the gap around the critical point $\Delta_1 \simeq \varphi |\varepsilon|^{z\nu}$. A numerical verification of such relation is reported in Fig. 8.3b [286]. In conclusion, experiments willing to verify the quantum Kibble-Zurek scalings shall be devised to explore the tight region of parameters where not only quantum effects are present (i.e., low temperatures and low decoherence) but also for which the system freezes in the quantum critical region defined by (8.15).

8.3 Optimal Control of Many-Body Quantum Systems

We have seen that crossing a quantum phase transition or performing adiabatic quantum computation generating the minimal amount of errors is a nontrivial task to achieve as the adiabatic strategy is hindered by the gap closure. However, are there different strategies that one can adopt? The answer is positive, and the different strategies can be grouped in two main classes as depicted in Fig. 8.4a: (1) those that aim to speed up the process remaining adiabatic and (2) those that abandon the adiabatic condition and do not require the system to remain in the instantaneous ground state but only to reach the final state with minimal energy.

The former class of fast adiabatic passage can be achieved either adapting the functional time-dependence of the changing parameter $s(t)$ to the instantaneous gap, that is, replacing the bound obtained by the minimization in (8.6) by its actual time-dependent value and thus speeding up the whole evolution [270]. Another potent strategy is to add terms to the Hamiltonian such that the nonadiabatic terms, which couple the instantaneous ground state with the excited ones, are exactly canceled [305–313]. This class of solutions, which goes under the name of *shortcuts to adiabaticity*, is very powerful and have been adopted in different situations, mostly few-body or where simple descriptions of the system exist (see [309] for a review). However, for many-body quantum systems the required additional terms quickly become pretty complicated (i.e., k -body interacting or long range) and different or hybrid strategies might be required to bring them to the lab [314]. The second class of strategies is mainly based on optimal control theory, on which we will focus hereafter.

Quantum optimal control theory solves problems which might be recast in a functional minimization: given an initial state vector $|\psi_0\rangle$ and a dynamical law – in our case Schrödinger equation or its generalization to open and non-linear dynamics as we will see later on – depending on one (or more) external control field $\Gamma(t)$, a figure of merit F is introduced that depends on the control field. The figure of merit might include different terms usually quantifying the property of interest to be achieved and some constraints, the total field strength, the total time of the evolution, etc. The solution of the optimal control problem is then given by the control field

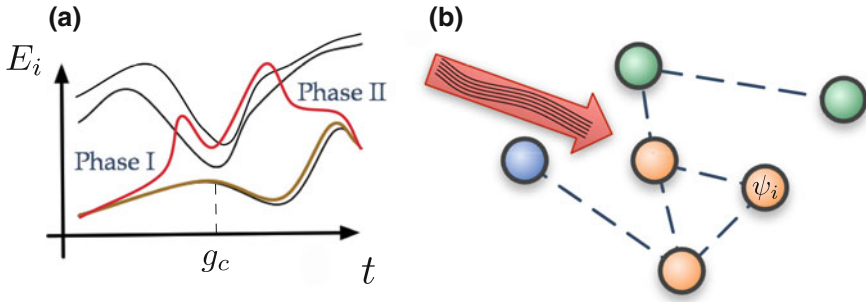


Fig. 8.4 **a** Eigenenergies E_i as a function of time (black lines) and instantaneous system's energy $E(t)$ of a system crossing a quantum phase transition via an adiabatic passage, either slow or fast by means, e.g. of counteradiabatic terms (gold line) or optimal passage (red line). **b** Sketch of a typical control of a many-body quantum systems problem: **a** an external control field (right arrow) acts on the system or a part of it, on external (e.g., particle position) or internal (e.g., atom levels) for a finite time with some finite energy inducing a transformation to a different state with different topology, particle numbers and/or internal states and that might also be a macroscopic superposition of different states of the system

that minimizes the figure of merit, that is, that solves a constrained functional minimization. Finding the solution to this problem is nontrivial also when the number of degrees of freedom is limited and raises many interesting questions. Optimal control of quantum systems has been achieved in many different settings and the conditions under which this is feasible has been analyzed widely regarding controllability of the system, finite resources as for example energy-time relations or due to other physical constraints [271, 315–336].

Less explored, due to the additional complexity in their numerical and analytical descriptions, are the theory and the applications of optimal control to many-body quantum systems. Here, additional dimensions of the problem appear, as sketched in Fig. 8.4b: only recently, the role played by the new naturally arising properties of a many-particle system (e.g., correlations or entanglement in the system, the tensor structure of the system Hamiltonian and the control fields, the coordination number and topology of the connections, and of the scalability with the number of constituents of the system) has been started to be investigated [271, 313, 318, 322, 324, 327, 335–351].

In general, a quantum optimal control problem can be recast in a functional minimization,

$$\min_{\Gamma(t)} F(\Gamma(t), \psi_0, T, \dots); \quad (8.16)$$

where F is a figure of merit that quantifies the quality of the result of the transformation, $\Gamma(t) \in \mathcal{G}$ is the control field and \mathcal{G} the space of all “physically reasonable” functions defined in $t \in [0, T]$; such that the system Hamiltonian can be written as $\hat{H} = \hat{H}_D + \Gamma(t) \hat{H}_C$ where \hat{H}_D is the drift part of the Hamiltonian that cannot be modified (e.g., an interaction between system components) and \hat{H}_C the operator cou-

pled linearly to the control field (e.g., the intensity of a laser or a magnetic field). The final value of the functional also depends on other degrees of freedom, for example, the initial state ψ_0 or the total duration of the evolution T . Moreover, additional constraints might be present such as the maximal available power of the control field or its bandwidth. Typical figures of merit are the fidelity of the final state with some desired goal state, the final expectation value of some observables or other properties such as final correlations of entanglement present in the system [316, 317, 323]. Here and in the following, we formulate the problem in terms of pure states, but it can be straightforwardly generalized to mixed states using density matrices and to (unitary) transformations [315–317, 323].

Many different algorithms have been successfully applied to minimize the functional in (8.16), mainly explicitly computing the gradient of the functional with respect to $\Gamma(t)$ and performing a gradient descent in the space of all possible control fields, as the Krotov or the GRAPE algorithms: Their elegant mathematical formulations and their successful applications can be found in the literature [316, 317, 323]. Here, we concentrate on yet another approach, the dressed chopped random basis (dCRAB) for its simple mathematical formulation and because it has been used to attack optimal control problems on many-body dynamics, such as the optimal crossing of a quantum phase transition, as we will see later on [321, 339, 352].

The dCRAB optimal control algorithm is based on the idea (inspired by tensor network methods) that it might be possible to reduce the space \mathcal{G} in which one searches the optimal control field $\Gamma(t)$ a priori, thus extremely simplifying the search. It should be noticed that this approximation is implicitly made in all known numerical algorithms: indeed, typically, the time is discretized, and the control field becomes a piecewise constant function with the ultraviolet cutoff Δt . However, this choice is seen as an approximation that shall be taken under control, typically with the condition $\Delta t \rightarrow 0$. On the contrary, we assume that it is possible to truncate the space \mathcal{G} to a small dimensional space $\mathcal{G}^* \subset \mathcal{G}$ and perform the minimization within this space to find an optimal minimum.

The first step in this program is to perform the truncation of the space: it shall be done in the most general way unless additional physical information are known: in this case, these information shall be exploited to introduce an educated guess which, however, it is typically not fundamental. An expansion into a truncated randomized basis of functions $\{h_i(t)\}_1^{N_c}$ is thus performed

$$\Gamma(t) = \sum_{i=1}^{N_c} c_i h_i(t), \quad (8.17)$$

The minimization defined in (8.16) is then recast in a multivariable minimization that can be attacked with standard methods, ranging from the conjugate gradient descent, genetic algorithms or, as typically done, by gradient-free minimization methods [122]. A typical choice for the basis functions $h_i(t)$ is a Fourier basis with randomized frequencies chosen within a bandwidth interval $\Delta\Omega$.

The crucial parameter here is the number of basis functions N_C needed in the expansion in (8.17) to find a satisfactory result. Early works have investigated the minimal number of independent parameters of the control function necessary to solve the optimal control problem and set a heuristic rule on the number N_C necessary to solve the optimal control problem, which scales as the Hilbert space size [353]. Recently, it has been proven that the number of coefficients necessary to reach an error $\epsilon > 0$ is a polynomial function of the dimension D_{W^+} of the space W^+ of states that can be reached in polynomial time with the Hilbert space size [330]. An upper bound for such dimension is the whole Hilbert space dimension, and for few qubits scenario, they typically coincide. However, in the many-body scenario they can drastically differ [180]. In particular, any many-body dynamics that admits an efficient numerical representation lives in a polynomial space, thus drastically reducing D_W^+ with respect to the exponentially large Hilbert space, resulting in an efficient solution of optimal control problems. Indeed, it is possible to show that, to reach a final error $\epsilon > 0$ the dimension of the space of functions to be searched in $D_q^* \equiv N_C$ can be bounded by a polynomial function of D_W^+ [330]. It is found heuristically that typically a linear lower bound can be saturated [352, 353]. In particular, exploiting a simple restarting and change of the random basis used to perform the optimization as introduced in the dCRAB formulation, allows escaping local traps with probability one for power-unconstrained problems [352]: the net results is that it is possible to solve the minimization problem with a series of one-dimensional $N_C = 1$ minimizations on different random directions in the search space. This approach has the additional advantage that minimizations in one dimension can be achieved with highly efficient simple algorithms, without the need to exploit sophisticated approaches such as evolutionary, gradient or higher orders algorithms. Finally, it has been shown that the minimal time needed to achieve an optimal transformation with finite precision is bounded by the control field bandwidth $\Delta\Omega$, i.e.

$$T \geq \frac{D_W}{\Delta\Omega}, \quad (8.18)$$

where D_W is the dimension of the set of reachable states [330].

We conclude this section presenting some results on the application of optimal control problem to quantum annealing. As shown in Fig. 8.4, the crossing of a quantum phase transition can also be achieved relaxing the adiabatic condition but simply imposing that the final state minimizes the energy of the final Hamiltonian. In such a way, we can aim to speed up the process not being constrained by the adiabatic condition. This analysis has been performed in [271] where it has been shown that it is indeed possible to speed up the quantum phase transition crossing: as a first test case the Lipkin-Meshkov-Glick (LMG) has been studied, a many-body model (Ising chain with infinite range interaction) that being integrable is perfectly suitable for such studies [354]. As shown in Fig. 8.5a, where the LMG model spectrum is reported as a function of time while the transverse field is varied either linearly (adiabatically and not) or optimally: it is clearly visible that while a fast naive passage ends in a highly energetic state, both the adiabatic and the optimal passage result in a

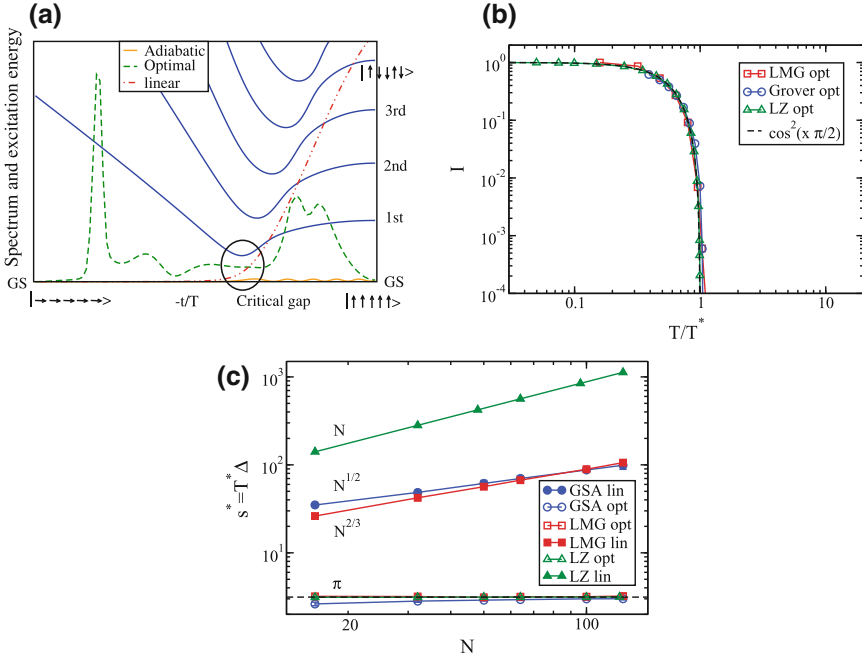


Fig. 8.5 **a** Adiabatic (yellow line), fast (red dot-dashed line) and optimal (green dashed line) passage of the quantum phase transition in LMG model. Blue full lines report the instantaneous eigenenergies of the system and the critical gap is highlighted. **b** Final infidelity as a function of the rescaled total time with $T^* = \pi/\Delta_1$ of transformation identifying the quantum speed limit of the process in different optimal passages (LMG model, Grover adiabatic algorithm, and Landau-Zener). **c** Scaling with system size N of the total time needed for linear and optimal annealing. For the Landau-Zener process we assumed an effective system size given by $N = \Delta_1^{-1}$. All figures are reported from [271]

good approximation of the ground state. Similar results can be obtained on different systems and protocols, such as the adiabatic Grover algorithm or the crossing of the superfluid to Mott insulator QPT in cold atoms in optical lattices [271, 318]. It is then natural to investigate how much one can compress the time necessary to cross a QPT and if a fundamental limit exists: in Fig. 8.5b we report the final error as a function of the total time of the process for the LMG model, the adiabatic Grover algorithm and, for comparison, a simple Landau-Zener crossing. As it can be clearly seen, in all these scenarios it exists a minimal time T^* below which the optimal transformation cannot be found and the error scales as $I = 1 - |\langle \psi(T) | \psi_G \rangle|^2 = \cos^2(T/T^*)$, where $T^* = \pi/\Delta_1$ and Δ_1 is the minimal gap of the system. Here T^* is the *quantum speed limit* of the process, the minimal time necessary for the transformation to occur, a consequence of the Heisenberg indetermination principle [325, 326, 329, 332, 333, 335, 336, 350, 355–361]. This result is in perfect agreement with the analytical solutions available for the two-level systems and supports the conjecture that also in more complex scenarios we can achieve the fastest transformation allowed

by nature, and that can be interpreted as an optimal two-level transition between the initial state and the final one. Finally, in Fig. 8.5c we report the scaling of the speed up as a function of the system size: the optimal crossing speeds-up the process by a square root factor in the system size (similar to the Grover algorithm gain) and by a prefactor of about one or two orders of magnitude. In conclusion, optimal annealing can introduce an advantage which however is still limited by the minimal gap of the system, even though in a more favorable way than the adiabatic passage.

Finally, we report that an optimal passage of a quantum phase transition has been experimentally verified in one-dimensional cold atoms in optical lattice undergoing the superfluid to Mott insulator transition [318].

8.4 Other Applications

In this section, we review some of the physical scenarios of out-of-equilibrium physics that can be addressed also or uniquely using TN techniques: we included them here as they have attracted a vast interest in the last years or we think that they will be subject to further investigations in the years to come.

The application of TN methods to lattice gauge theories in the context of high-energy physics is not limited to equilibrium properties as shown in Chap. 6. Indeed, they have been already applied to the study of dynamical phenomena such as the simulation of the real-time evolution of string breaking, of the Schwinger mechanism and of scattering processes. String breaking is an important phenomenon that occurs when the gauge field string connecting two charges in a gauge theory (e.g., the electric field between two charges, the gluon field between two quarks, etc.) is too costly energetically, and thus it is more convenient to break the string and produce from the vacuum particle-antiparticles pairs. The production of such pairs from the vacuum in QED is known as Schwinger mechanism [362]. Due to the sign problem in Monte Carlo methods, the study of these phenomena in the last decades has been mainly performed observing indirect features, such as the string tension or Wilson loops [242]. A semi-classical simulation of the real-time evolution of string-breaking has been presented only recently [363], followed by fully quantum simulations using TN, for a $U(1)$ and a $SU(2)$ lattice gauge theories [57, 68]. Moreover, the real-time evolution of the scattering process between to bounded particle-antiparticles states as been presented in [57], complemented by an analysis of the formation of entanglement created during the scattering process.

Another large field of the application of the techniques introduced in the previous chapters is the study of out-of-equilibrium many-body quantum systems in different forms: the field of out-of-equilibrium quantum phase transition is attracting increasing interest, that is, the characterization of the different phases a system in the presence of a bath that induces the system to equilibrate. In the presence of competing terms, different properties can arise in the fixed point of the dynamics as a function of the different weights of the unitary and/or non-unitary terms [364, 365]. On the one hand, this effect can be exploited to perform the so-called *reservoir*

engineering, that is, the development of a specific bath that drives the system into a state with interesting characteristics, as for example, the presence of entanglement to be exploited for quantum information protocols or the preparation of some desired state. Given that the state is a fixed point of the dissipative dynamics, it is inherently robust against perturbations, as already experimentally observed [265–267]. On the other hand, one can study the physics of such phenomena, as in the paradigmatic system of coupled cavities in presence of pumping and dissipation, that has been shown to display a quantum phase transition of light [366–368], or in the recent experimental verification in a string of trapped ions simulating the Ising model in transverse-field [369].

On the more statistical mechanics side, TN methods have also been employed to study fundamental questions related to the thermalization of many-body quantum systems, that is, to investigate if and under which conditions a closed many-body system acts as a thermal reservoir for its subsystems, and how and if the thermalization occurs [370, 371]. More generally, one can study the effects of quenches, the onset of many-body localization and of quantum chaos [294, 296, 371–379].

The possible applications of tensor networks in open quantum systems go well beyond the study of the properties of the steady states of the systems: for example it has been shown that it is possible to map system-environment quantum models to effective spin chains [380, 381], or apply them to the study of atoms coupled to optical fibers, e.g. to study of chiral quantum optics and photonic circuits with time delays and quantum feedback [382–384].

Finally, an increasing interest has grown in the application of concept of classical complexity to classify states of many-body quantum systems, also in the context of cellular automata, in particular the quantum counterpart of Conway’s game of life [385–388].

8.5 Exercises

1. Using the t-DMRG algorithm, simulate the adiabatic passage of the Ising model in transverse field through the quantum critical point for different system sizes and total times of the passage. Compute the final energy and final correlation length and compare them with the values on the exact final ground state.
2. Individuate the Landau-Zener and the Kibble-Zurek regime in the residual excitation energy computed in the previous exercise. Extract the system critical exponents and discuss possible differences between the theoretical and computed ones.

Appendix A

Hardware in a Nutshell for Physicists

Computational physics lies at the frontiers of physics in two ways: (i) from the biggest experimental efforts at CERN to the single table-top experiments in cold atoms, the support of dedicated software and hardware is fundamental. In both cases, without the automation allowed by software control most of these experiments would not be possible, and the postprocessing and the analysis of the data often requires terabytes of storage and thousands of lines of code; (ii) the complexity of the software dedicated to simulate physical process has become so high that the numerical experiments now matches real ones in terms of complexity (years to be set up, thousand of lines of code), resources (supercomputers, months of computation time), not to speak of the ability of description of reality. It is then necessary to know the tools needed to attack such fascinating challenges to be able, if necessary, to modify the smallest detail of the experimental apparatus, including the software and the hardware used to run it. Hereafter, we present some fundamental concepts of computer architecture that any physicist planning to set up numerical experiments, even small, should know and might serve as a starting point for more advanced reading.

A.1 Architecture

Modern computers are built according to the von Neumann architecture, whose sketch is presented in Fig. A.1, composed of the following components each dedicated to a given task:

Memory, dedicated to data and program instructions storage.

Central Processing unit (CPU), divided in a Control Unit that fetches the instructions/data from memory and decodes the instructions and coordinates the operations to be performed, and an Arithmetic Logic Unit (ALU) dedicated to performing data processing.

Input and Output (I/O) devices, the interfaces with external storage and the users.

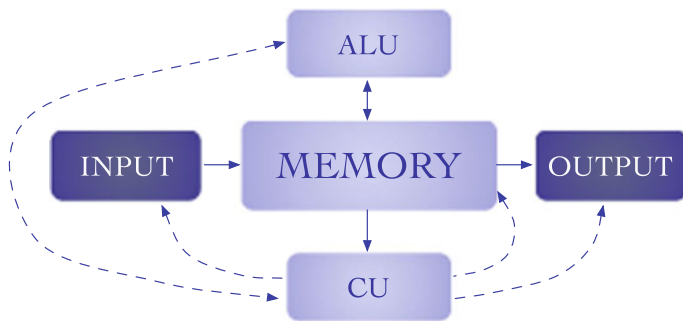


Fig. A.1 Von Neumann's computer architecture: the Control Unit (CU), Arithmetic Logical Unit (ALU), Memory, inputs and outputs devices, and their relations (Bold lines denote data/instruction flow, dashed ones control signal flow)

The main innovation introduced by this architecture, which nowadays might appear somehow natural, is that it allows building a general purpose computer, while previous architectures were built to perform a specific task, i.e., the program was hardwired, not stored. The von Neumann architecture allows instead to write the computer program (the set of instructions it has to perform) in its memory, making it a machine highly versatile, as witnessed from the fact that since its introduction it has essentially remained unchanged. Indeed, in the last decades, all computer components have been extensively revised and improved, but there has been no need to rethink this basic computer architecture apart from one innovation first introduced in the PDP-11 manufactured by DEC in the late sixties. This innovation is the *system bus*, dedicated hardware that transfers data among the components along a unique path shared by them all: The uniqueness of the bus simplifies the arbitration of the possibly conflicting accesses to the memory requested concurrently by the other components.

A.2 Data and Formats

As well known, the data in modern computers is stored in bits, using binary coding. That means that there is a collection of physical entities (the memory of the system) each of them having only two possible different states (referred as 0 and 1). What seems now natural is not the only possible choice: replacing this definition results in highly interesting consequences: a quantum-bit is defined by two quantum states $|0\rangle$ and $|1\rangle$, from which all quantum technologies are nowadays stemming from [7]. Coming back to classical bits, for example, one can write the number 42 as

$$101010, \tag{A.1}$$

which should be read as $1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$. In the jargon, 8 bits form a byte, 2 bytes form a ‘word’, 4 bytes a ‘long word’ and 8 bytes a ‘long integer’. Having defined positive numbers, the elementary operations can be constructed as usual. Here we briefly report the addition of two bytes:

$$\begin{array}{r} 2 \quad 00000001 \\ + 3 \quad 00000011 \\ = 5 \quad 00000101, \end{array} \tag{A.2}$$

where the sum of each digit is done modulus two and the carry (if present) is transferred to the digit on the left.

Less straightforward is the introduction of negative numbers, however, it can naturally be introduced recalling their definition: $-n$ is the number that summed to n results in zero. It is interesting to look for a representation that does not require new circuitry to perform additions - on the other side, one needs to introduce something to change the sign to an integer. Thus, we shall look for a number that added (using the binary adder, hence mod 2^8 , since for simplicity we keep assuming 8 bits) to n according to the previous definition results in zero. This is achieved by the *2s-complement representation* of the integers, that keeps the usual representation of the positives in $[0, 2^7)$ and defines the opposite of a number $n \in [-2^7, 0)$ as $(2^8 - n) \bmod 2^8$. Summing modulo 2^8 , we have $n + 2^8 - n = 0$. A simple example with $n = 1$, hence $-n = 2^8 - 1 = 11111111$ in base 2, helps to visualize the general solution:

$$\begin{array}{r} 1 \quad 00000001 \\ + -1 \quad 11111111 \\ = 0 \quad 00000000. \end{array} \tag{A.3}$$

Here is another interesting example:

$$\begin{array}{r} 21 \quad 00010101 \\ + -1 \quad 11111111 \\ = 20 \quad 00010100. \end{array} \tag{A.4}$$

In general, the problem of finding the 2’s complement representation of the opposite of n , that is computing $(2^8 - n) \bmod 8$ can be implemented as $((2^8 - 1) - n) + 1$. This is convenient, since the subtraction in the first addend (in binary with 8 digits) does not require borrows, and its result can be obtained just negating each bit of the subtrahend (a zero turns into a one and viceversa). For instance, $(2^8 - 3)$ becomes $(11111111 - 00000011 = 11111100$. The operation of flipping each bit in n is often denoted by \bar{n} , an operation which is useful in itself, like many other elementary operations on bits (OR, AND, XOR etc...). Hence, $-n$ can be implemented as $\bar{n} + 1$ almost at no cost, i.e., without requiring specific circuits.

In conclusion, with one byte, one can store the natural numbers from 0 to 255 and the integers from -128 to 127:

<i>Natural Integer</i>		
0	0	00000000
1	1	00000001
2	2	00000010
	⋮	
127	127	01111111
128	-128	10000000
	⋮	
253	-3	11111101
254	-2	11111110
255	-1	11111111

(A.5)

To understand which is which one needs to know a-priori which representation is intended, as it happens, for instance, when a decimal input is converted in binary.

A programming language has few predefined types, one of which is INTEGER: when one defines a variable of a given kind, a slot in the memory is reserved to store its value. The type of variable defines the amount of memory reserved and the biggest and smallest number one can store safely. If one exceed these limits during a computation, unpredictable errors might occur: this is the ‘overflow’ error. In FORTRAN (hereafter, we concentrate on this programming language for scientific calculations. Anyway, although the examples refers specifically to it, corresponding objects can be found in any programming language) there are the following kind of integers:

TYPE	Lenght	Range	Max
INTEGER * 2	2bytes	$[-2^{15} : 2^{15} - 1]$	$\approx 10^4$
INTEGER * 4	4bytes	$[-2^{31} : 2^{31} - 1]$	$\approx 10^9$
INTEGER * 8	8bytes	$[-2^{63} : 2^{63} - 1]$	$\approx 10^{19}$.

We now look at how real numbers are stored in a discrete binary memory according to the IEEE standard 754 [390]. We start from the fact that a real number can be written as $(-1)^S \times M \times 10^E$, where M is the mantissa, S determine the number sign, and the E the exponent specifying the order of magnitude of the number. We can then store a real number, up to some finite precision, in a binary register as follows. For sake of clarity, here we specify it for the single precision (4 bytes long or REAL*4 in FORTRAN):

Position	31	30 . . . 23	22 . . . 0
Content	<i>S</i>	<i>E</i>	<i>M</i>
Length	1bit	1 byte	3 byte – 1 bit

and the corresponding number is defined as $(-1)^S \times 2^{E_8 E_7 \dots E_0 - 127} \times 1.M_{22} M_{21} \dots M_0$, where $E_8 E_7 \dots E_0$ is the binary coding of the exponent E and $1.M_{22} M_{21} \dots M_0 = 1 + M_{22}/2 + M_{21}/2^2 + M_{20}/2^4 + \dots$ encodes the mantissa. These are the so called floating point numbers as the order of magnitude of the stored number changes according to the exponent E . Note the bias in the exponent definition, which allows encoding positive and negative exponents. In conclusion, the previous encoding allows to store safely numbers ranging from 2^{-127} to about 2^{128} , that is from about 10^{-39} to 10^{38} .

Does that mean that we have about eighty digits of precision available? A careful look at this question will result in a negative answer. Indeed, although we can store numbers that span eighty orders of magnitude, the precision is much less as what counts is the maximum extent of numbers on which we can safely perform operations, and this is related to the amplitude of the mantissa. For example, if we want to add two floating point numbers, we shall first of all write them with the same exponent, and then perform the binary sum of the mantissas. Thus, the two most far apart numbers that we can meaningfully add are 1111 . . . 1110 and 0000 . . . 0001, that is 16772215 and 1, which correspond to eight digits of precisions. Once again, this precision is a crucial element to be taken into account while performing numerical simulations as a wrong estimate of the needed precision might lead to uncontrolled results.

To increase precision, the only possibility is to extend the memory dedicated to store each number, that is, to use double precision (REAL*8 in FORTRAN) which reserves 8 bytes for each number, divided as follows: 1 bit for the sign, 11 bits for the exponent and 52 bits for the mantissa. A straightforward calculation yields that the range of numbers which can be stored is from the order of 10^{-308} to 10^{308} , with sixteen digits of precision. Some programming languages allow defining higher precision real numbers (quadruple precision or even more); however, one should use them responsibly. Indeed, increasing the precision typically implies also reducing the computational performances of the simulation. Given that only very few physical processes can be studied or achieved with such impressive precision (atomic clocks are nowadays at about eighteen sixteen digits of precision and struggling to improve even more [391]), the need of using higher than double precision should be carefully evaluated and motivated.

A.3 Memory and Data Processing

As stated previously, computers store data and instructions in the Memory, with the CPU in charge of the data processing. To describe the main components within a CPU, hereafter we concentrate on the old i8086 CPU – introduced at the end of the 70s and used worldwide in most desktop computers until the beginning of the 90s

– which provides a perfect playground to introduce the concepts. Clearly, modern CPUs have differences with respect to what described in the following. However, the main concepts are still valid and hopefully will provide the reader with the basic knowledge to begin exploring the modern CPU components and to understand the specific description provided by each vendor.

The core of the CPU is the Control Unit (CU) – called Execution Unit (EU) in the i8086 –, which implements a finite state machine (an automaton with a finite number of states and of possible transitions between them) that controls the computing processes.

The CU goes repeatedly through two steps: first it *fetches* an instruction from the *instruction queue* and then dispatches the sequence of orders (μ instructions) needed to perform it to the other involved units, e.g., the ALU for the arithmetic operations. Concurrently, the instruction queue is fed with the contents of consecutive Memory locations. Indeed, most of the time the next instruction to be executed is implicitly the next one in Memory, with the notable exception of the *jump* orders, which specify explicitly where to fetch the next instruction. When a jump is executed, the instruction queue is flushed, and feeding restart from the new location.

In von Neumann’s original architecture, instructions were fetched directly from the RAM: The instruction queue was introduced as performing one operation might be much faster than retrieving the next one from memory. Thus, more consecutive instructions are loaded at the same time to avoid the control unit to remain idle. This is the first example of a design tradeoff: augmenting the complexity and cost of the circuitry to gain in performance. The same idea lays behind the introduction of the *virtual memory* and of the *memory caches* that we discuss next.

The memory where data and instructions are stored during a computation in von Neumann’s architecture is such that the time needed to fetch information does not depend on its location: hence the name, somewhat confusing, of Random Access Memory (RAM).¹

The memory device devoted to permanently storing the data and all the software needed for the computer to run is the hard disk. However, retrieving information from the hard disk is typically much slower than accessing the RAM. Besides, the capacity of the available RAM is typically much smaller than that of the hard disk and thus data and instructions need to be swapped back and forth between them. Indeed, it has been early recognized that the set of instructions decoupled from the main program and dedicated to specific tasks (as those contained in single subroutines or functions) and the use of data structures such as vectors, introduce temporal and spatial correlations in the usage of data stored in memory [392]: The *principle of locality* in computer science states that programs tend to remain most of the time in contiguous blocks of memory. Thus, having an efficient way of handling contiguous blocks of memory can be highly rewarding.

¹Indeed, previous automatic calculators used different memory structures, where in general the time needed to retrieve a piece of information depended on its location. Nowadays, the same applies to the hard disks – the mechanical ones, not the solid state ones.

The desired information is identified in the memory by *addresses*, which give the position of a specific byte in the memory. The width in bits of the addresses defines the *address space*: e.g., with 32 bits one has an address space of 2^{32} bytes. To simplify memory management and to introduce the possibility to mimic a RAM larger than that physically present, according to the *virtual memory* concept each program is endowed with a larger address space than that of the real RAM, typically 4 GB versus 1 GB. The virtual memory is divided into *segments* of 64 KB each, so that any address can be seen as composed of a *base*, identifying the segment (the most significant 16 bits) and an *offset*, the least significant 16 bits, identifying a byte within the segment.

During execution, according to the locality principle, only few virtual segment are actually present in the RAM at any time: the operating system, in charge of memory management, holds a map of the virtual segments bases to the physical ones currently in the RAM. This map is exploited by the compiler to store in dedicated *registers* (fast memory slots in the CU) the actual base of the segments in use, so that the real addresses can be computed from the content of the register and the offset contained in the instruction. This is done exploiting the arithmetic capabilities of the ALU, i.e., no special hardware is needed for address calculation. The advantage of such a structure is that whenever it is needed to move blocks of contiguous memory, it is possible to specify only once the segment address, sparing half of the address length for each instruction.

In the i8086 architectures, there are five dedicated sixteen-bit registers to store the segment addresses and the instruction pointer (segment registers) plus eight sixteen-bits *main registers*: four to hold the offsets within the segments, and four to store the data locally and perform the operations, also at the single byte level. Doubling the number of bits used to specify addresses, one can access a memory of $2^{32} = 4$ GB. This was considered, until recently, far than enough for any reasonable application.

To further exploit the locality principle, cache memories have also been introduced: they are intermediate blocks of memory with a higher rate of response than the RAM and physically sit near the processor, so that also the communication to and from it and the CPU is much faster than from the CPU and the RAM. The following strategy is then adopted: whenever an instruction or some data need be retrieved from the RAM, a bigger block of memory contiguous to the indicated address is copied into the cache. According to the locality principle, most of the times this will speed up the computation, since the likely next needed information will already be present in the cache. In modern computers, there may be up to three levels (referred as cache L1, L2, and L3) with smaller and faster units nearer to the CPU.

An optimizing compiler will exploit the cache as much as possible. So, the code should be written to take advantage of the cache, or at least not to row against. Typically, when working with big matrices, knowing how they are stored in the memory (either by row or by column) and writing the operations accordingly, is fundamental to achieve high performances.

Nowadays, floating point operations are performed by a *coprocessor*, i.e., a dedicated piece of hardware (the i8087, in the first series). Initially, its performances were still pretty slow as it required one hundred *clock cycles* (the unit of time of the

processor, nowadays usually of the order of GHz, at the time one thousand times slower) to perform a single operation and hundreds of cycles to perform a complex trigonometric function. Nowadays, not only the clock is much faster but also the chip structures are improved, and only one clock cycle is necessary to perform standard floating point operations.

A.4 Multiprocessors

When one aims at pushing to the limit his capabilities to simulate (quantum) systems, a possible strategy is to achieve the best performance out of the available resources – the main topic of this book; the other possible strategy is to increase the available resources. Thus, one can – and eventually should – think of running the code in parallel, distributing the work to many independent processing units. For decades the possibility to run code in parallel has been possible exclusively on large supercomputers. However, nowadays there are different options we will briefly introduce in the following, together with pointers to the relevant literature to guide the interested reader.

The main difference between the possible parallel strategies relies on the interface between the RAM and the CPU. Indeed, a general parallel network can be depicted as in Fig. A.2, composed of different nodes of a cluster, each of them containing some local memory shared by several processing units. In modern architectures, each node might be composed of different CPUs each of them composed of independent cores, acting as an independent processing unit. Each core can process information independently from the others, executing its thread. Nowadays, in standard clusters, there are from eight to thirty-two cores in each node. The communication speed among cores and the local RAM can be considered instantaneous compared to that between nodes which introduces some overheads in the computation time as we will see later on. However, while one can add virtually an unlimited number of nodes (top500 supercomputers have typically more than ten thousand nodes [393]) and is typically limited only by practical considerations (installation and maintenance costs) scaling the numbers of cores inside the CPU is a highly nontrivial engineering challenge.

The different parallel architectures have to be addressed with the proper software tool, which we list hereafter for completeness together with some reference for further reading.

OpenMP API (Application Program Interfaces) define a set of instructions to perform multi-thread, shared memory parallel calculations on multi-core machines. There are different implementations for almost all the popular programming languages [394]. Moreover, modern compilers allow us also some automatized multi-thread optimization via specific options, typically in combination with standard mathematical libraries, e.g., LAPACK libraries in the Intel MKL (Math Kernel Libraries) implementation [130, 395]. The use of OpenMP, at least in its most straightfor-

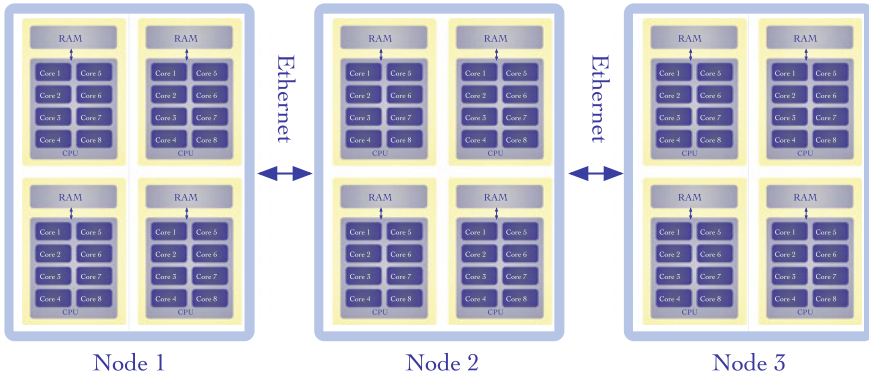


Fig. A.2 Cluster architecture: here 4 CPUs belong to each node, each of them composed by 8 cores (which share the same RAM). The different nodes are connected via Ethernet protocol or higher performing ones (InfiniBand)

ward applications, can be highly rewarding at a limited cost in time to exploit them, although it is always limited by the numbers of cores present in each node.

MPI (Message Passing Interface) is a library specification to perform parallel computation between different nodes also in heterogeneous networks, implemented in most of the standard programming languages [396]. Its use, however, requires a higher investment to learn the basics concerning what is required to use OpenMP. However, MPI allows the user to virtually scale the computational resources of orders of magnitudes. Unfortunately, this does not guarantee the same speedup in computational time. Indeed, the speedup strongly depends on the particular computation: Amdahl’s law states that any program has a part which is inherently sequential – it cannot be parallelized – as the loading of the data; and a part which can be parallelized [397]. The total computation time on a single processor is then $T = T_s + T_p$ (where T_s and T_p are the time spent in the sequential and in the parallel part respectively), while running the code on M processors results in at least a total time of $T_M = T_s + T_p/M$. Thus, in the limit of infinite processors, we have that $T_M \rightarrow T_s$. However, if one has to decide on how many processors run code in parallel, the interesting quantity to look at is the asymptotic gain per processors, that is

$$\lim_{M \rightarrow \infty} \frac{T}{T_M} = \frac{T_s + T_p}{T_s} = 1 + \frac{T_p}{T_s}.$$

Thus, in the limit of many processors, to achieve a high gain for each additional processor, we should aim to reduce T_s as much as possible, as in the limit of vanishing T_s the gain per process diverges. So, given that the condition $T_s \ll T_p$ holds, how much gain one could expect? Mainly, the losses arise in the communication between nodes during parallel algorithms and have two different sources:

1. Latency times, the time needed to establish a connection between nodes, which sum up to a time proportional to the number of transfers.

2. Transfer times, the time needed to transfer the data between nodes limited by physical factors as the speed of transfer, the bandwidth and the physical distance between nodes, which sum up to a total time proportional to the number of bits to be transferred.

Thus, we can classify some typical linear algebra operations, which lay at the heart of most of the algorithms for simulating quantum systems, studying the ratio between communication and computation time.

Operation	# bits sent	# operations	Ratio
$sa \cdot a$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$
$a \cdot \mathbf{v}$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(1)$
$\mathbf{v} \cdot \mathbf{v}$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(1)$
$\mathbf{O} \cdot \mathbf{v}$	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2)$	$\mathcal{O}(1)$
$\mathbf{O} \cdot \mathbf{O}$	$\mathcal{O}(n^2)$	$\mathcal{O}(n^3)$	$\mathcal{O}(n)$

where a , \mathbf{v} , \mathbf{O} are a scalar, a vector and a matrix respectively and n is their size. When the ratio is of the order one, the computation speedup will be limited by the network as the number of transfers scales as the number of operation to be performed. On the contrary, when the ratio scales with the problem size, we might expect substantial speedups coming from the parallelization for large n , that is, for big matrix size and “level 3” processes (composed by three nested loops).

Finally, different algorithms can be classified in three different classes according to their behaviour under parallelization, as even in the perfect scenario the gain is typically less than T_p/T_s . In this respect, one encounters:

1. 100% algorithms, formed by independent calculations such as those due to statistical sampling or exploration of the parameter space. Here the problem can typically be trivially paralleled, and one obtains a speedup linear with the number of processors M .
2. Semi-efficient algorithms, where the gain that can be achieved is still linear with M . However, for most of the time a significative number of processing units remains idle, wasting resources.
3. Costly algorithms where some gain is obtained but overheads should be added to allow for parallel computation.

To finalize this section it is worth mentioning that other parallelization architectures can be explored that, depending on the resources at hand and on the problem to be solved, could give significant speedups. Beowulf clusters connect standard machines in parallel and are pretty straightforward to set up [398]. This can be a solution to exploit to its maximum outdated or heterogeneous hardware, a cheap solution to enter in the world of parallel computation and extremely useful for didactic purposes. Another option which is attracting increasing interested in the last years is the use of OpenCL, a programming language to program parallel code running on different architectures also integrated with Graphical Processing Units (GPUs). GPUs are very powerful dedicated hardware initially developed to match the needs of

powerful graphical processing by the software game industries. They can be usefully exploited to speed up some parts of scientific computations [399, 400].

A.5 Exercises

1. Integer and real numbers have a finite precision. Explore the limits of INTEGER and REAL in FORTRAN:
 - a. Sum the numbers 2,000,000 and 1 with INTEGER*2 and INTEGER*4.
 - b. Sum the numbers $\pi \times 10^{32}$ and $\sqrt{2} \times 10^{21}$ in single and double precision.
2. Consider a quantum system formed by N (distinguishable) subsystems (spins, atoms, particles, etc..) each described by its wave function $\psi_i \in \mathbb{C}^D$ where \mathbb{C}^D is a D -dimensional Hilbert space.
 - a. Write a Fortran code to describe of N non-interacting systems and for a general N -body wavefunction $\Psi \in \mathbb{C}^{D^N}$. Comment their efficiency.
 - b. Write a code to compute the density matrix of a general N -body wave function $\rho = |\Psi\rangle\langle\Psi|$.
 - c. Characterize the functions introduced above for different N in terms of time and memory requirements. What is the biggest N you can reach?

Appendix B

Software in a Nutshell for Physicists

The goal of computational physics is to attack problems that cannot be treated with other methods, to complement experimental and theoretical studies, building trust on their results and exploring regimes of parameters out of their reach or where the analysis is particular difficult. This is particularly true for computational quantum physics, where the field of application of analytical tools is vastly challenged, e.g., in many-body systems. Especially when aiming to perform top-class academic research, the methods and tools are not only the bare application of state-of-the-art techniques: on the contrary, they are pushed to the limit, in a fast evolving scenarios where requirements and goals change very quickly. Indeed, differently from applied and commercial fields where software is developed, the specification of what the software shall do, under which circumstances and with which resources (software requirements specification) is not – and cannot be – precisely defined once for all, but keeps evolving as the use of the software increases the understanding of the problem under attack. Thus, when planning to develop scientific software, one should be aware that no matter how hard one tries, one will eventually ends to have to fulfill three requirements, that is to

- R1 Change the software adapting it to the evolving situation and the corresponding needs;
- R2 Work with a lot of data²;
- R3 Solve hard problems with either large-scale and/or inefficient algorithms.

Given that the three requirements R1, R2, and R3 above are so general and ubiquitous, professional computational physicists shall learn how to address them, as much as they learn how to solve an integral or a particular technique in the particular field of study they are devoted to. Indeed, the three aforementioned requirements call for the use of some strategies that have been developed in the last decades and in particular:

- S1 Software Engineering: writing good, flexible software, easy to debug and expand;

²Even though a computational physicist might not work with “Big Data” in proper sense, the amount of repetitive tasks requested by careful studies is typically far from what can be done by a human being.

- S2 Scripting: automatize work as much as possible as pre- and post-processing will require as much work as producing the data;
- S3 Optimization and HPC: be as efficient as possible, and be prepared to use “brute force” if needed.

This chapter initiates the reader into the aforementioned strategies, which shall be adopted to attack scientific computations in the most efficient way. To attack the seemingly overwhelming challenges described before, a good starting point are the set of priorities described below that anyone writing scientific software should follow as much as possible [120, 121]. Those priorities are valid in general for any scientific software (and most of them for any software), however, in the following they will be specialized for quantum physics for the sake of clearness. They are, in order or priority and accompanied by a clear motivation for such priority,

1. Correctness — to be useful;
2. Numerical Stability — to be trustful;
3. Accurate Discretization — to describe nature;
4. Flexibility — to be used;
5. Efficiency — to do a good job!

In the rest of this chapter, we introduce the reader to the most important aspects of each of them, that one should keep in mind to write good scientific software.

B.1 Correctness

The fundamental requirement for a good scientific software is obviously that they should be correct, that is, it shall perform the correct sequence of instructions, written in the correct syntax according to the programming language used and version and compiler compliant. However, even for highly skilled and experienced programmers, the probability of writing some lines of code without any error is essentially zero. Thus, actions and strategies to correct the unavoidable errors has to be developed and undertaken. The first support comes from the *compiler*, a program that translate the source code into a lower-level code. Typically, compilers manage to identify syntax errors and are constantly improved to include more and more possible error scenarios. However, even though the compilers runs without signalling errors and produces the compiled code, the executable program, might still crash due to many different causes such as, for example, a memory allocation problem (the program tries to write in a memory position that is not allocated), an *obi-one* (Off-by-one, a recursive index start from zero instead than one or vice-versa). Finding and correcting these and other errors is the “art of debugging” and requires time and experience. However, there are some good practices that can be adopted to speed up the process, briefly introduced below:

- Use the print statement to check the variables name together with compiler options which allows one to select lines of code for debugging (as, for example,

the `-d` option of the `ifort` compiler together with starting the line of the debugging test with a `d`):

```
1 d PRINT *, "The value of x is", x
```

This combination is a quick way to reduce the time of debugging as the lines of code written for this purpose are not cancelled but remain there, for later and repeated use when another bug appears.

- For more sophisticated checks, when multiple lines of code shall be written and/or it can be reused in other part of the program, some dedicated `SUBROUTINE` or `FUNCTION` might be defined, with an enabling global variable and extended flags and output. The simplest example of such code might assume the form of

```
1 FUNCTION Checkpoint
2 IF (DEBUG)
3     PRINT *, "The value of x is", x
4 ENDIF
5 END FUNCTION
```

- Finally, the use professional debuggers such as, for example the GNU debugger GDB, even with the help of a Graphical User Interface (DDD) [401, 402] is highly recommended above a certain level of code complexity.
- Producing output that is compatible with the input is a good practice that allows to automatically generate test of increasing complexity for the program. For example, if one is working with matrices or tensors, if the input and output styles match the output can be easily used as new input without the need of time-consuming format processing. This compatibility is also extremely useful in the data acquisition stage as in many cases the program might be restarted (e.g., due to computational limit time reached, additional check of convergence etc.).
- Include in the program sections “Errors and Exceptions” for known issues: even if one is not interested in correcting them, signalling them and including explicitly a test to exclude those exceptions will save an enormous amount of time during the debugging. For example, if while implementing a given algorithm a particular case is excluded, as it would require an additional effort not needed at the moment, including an exception might be highly beneficial when the same piece of code might be reused (see later, Sec. B.4) and might need exactly the excluded scenario: after months or even years, uncovering the exception might require hours of hard debugging.
- Perform testing against random input: even though random input is typically not ill-conditioned and thus it will possibly not cover all cases (for example random matrices are never degenerate [133, 403]), random testing allows one to perform a huge number of automated testing. Again, automated testing software is available for most common programming language.

If the previous practices are followed, it should be possible to implement different strategies to have an effective and efficient debugging. In particular, one can perform *Incremental Testing*, that is to test and debug every part of the program while writing

it: debugging different parts of the code separately is highly simpler than debugging them all together. However, even if all the parts are bug free, nothing assures that while combining them some errors will occur. Thus, another good strategy to adopt is *Regression Testing*, that is, to store the situations that produce bugs when they are spotted in a set of automated tests. Every time changes are made, one can rerun all tests which, provided they have been carefully planned and implemented, should highly improve the debugging speed. One can also embrace the *Extreme Programming* paradigm which relay, among others aspects, on designing and coding the tests even before writing the code itself. In such a way one should be able to easily automatize them and run them as much as possible [404].

Finally, even though the compiler compiles and the program runs without crashing, how can we be confident that the outcome is the correct one, that is, that the program is computing exactly what was written for? To increase the confidence in the output (and also to help in the debugging), it is useful to include some *proof of correctness* in the code, such for example:

- In every section of the code, even if they are trivial, include explicitly in form of comments and – if possible – of automated checks, *preconditions* and *postconditions*, with a clear message in case they are violated. This will cost the programmer some time, but will save an enormous amount of time when a bug as to be located and/or the code will return results which do not match your expectations. Typical examples of preconditions are the sign of a variable (e.g. if known to be positive), the norm of a vector, the dimensions of vectors, matrices, and tensors which typically should match. Postconditions can be individuated in some properties of the solution (e.g. eigenvector normalization, function monotonicity, expected result for some particular parameters, etc.) which should hold despite of the (unknown) value of the solution itself.
- Define loop-invariants if present as they can provide a proof for induction of the correctness of that part of the code: check the loop-invariant for a given value of the recursive index i , and check that if it is true for i then it is true also for $i + 1$.
- In physics and whenever it applies, typically one can individuate some trivial (e.g. wave function normalization, system energy, etc.) and less-trivial constants of motion which should be monitored.
- Comparisons with other programs results and with regimes where the solution of the problem is known (e.g. non-interacting case, single or few particles, low dimensional, etc.) should be performed before exploring the new regimes.

B.2 Numerical Stability

Even though all the aforementioned practices are strictly followed and the code is up and running and has passed the numerical tests – and thus one can safely assume that the program is correct and executes the algorithm has been written for – there might be cases where the output presents strange results that cannot be trusted. So, even

though the program is formally correct, the results shall be discarded: this is typically due to *numerical instabilities* which are mainly the result of the difference between real-numbers and floating-point arithmetics. To detect and avoid such instabilities might not be easy but there are a number of typical scenarios – described below – that one shall avoid while writing code [121].

- Never use floating point variables to test for equality: the lines of code

```
1 REAL*8 x,y
2   x=y
3 PRINT *, (x.eq.y)
```

might result in a FALSE output as the test for equality depends on the details of the architecture and compiler and on the check on the digits beyond the finite precision of the double precision variables. A check on the difference between the two floating point variables shall instead always used, e.g.,

```
1 REAL*8 x,y
2   err= 1d-14
3   IF (abs(x-y).lt.err) THEN
4     PRINT *, "x is equal to y"
5   ENDIF
```

- Similarly, never use floating point variables as counters. Indeed the cycle

```
1 REAL*4 x,C
2 DO (x=0.0; x<= 1.0; x = x+ 1.0/C)
```

will increase the variable x by $1.0/C$ every cycle. So, the cycle will loop $C - 1$ or C times depending on the round-off error on the finite precision representation of the incremental fraction, drastically and randomly affecting the result of the computation.

- Be careful with the result of the subtraction of two finite precision variables a and b if $|a - b| \ll |a|, |b|$. Indeed, if the round off errors of the two variables have opposite signs one can show that the relative error on the difference $c \equiv a - b$ is proportional to $\Delta c/c \propto (a + b)/(a - b)$, which can easily explode beyond the floating point precision.³ This error can occur frequently as differential calculus is obviously based on differentials – very small differences between two quantities, e.g. the first order approximation of a derivative is $f'(x) \sim \Delta f/\Delta x$. When the possible values of the independent variable span different orders of magnitudes the computation of $\Delta x = x_2 - x_1$ can suffer exactly of the aforementioned problem, resulting in drastic reduction fo the result precision if not in a completely random results.
- Finally, as all scientists know, the errors propagates during arithmetical calculations and given the fast growing number of operations that modern computers can do, they shall be carefully monitored. In particular, operations like factorial and repeated multiplications shall be handle carefully. For example the cycle

³To prove that simply consider the variables with their floating point errors and compute the relative error on their difference, i.e. $\hat{a} \equiv (1 + \varepsilon_a)a, \hat{b} \equiv (1 + \varepsilon_b)b, c \equiv (1 + \varepsilon_c)c$ and compute the relative error on c given that $\varepsilon_a \sim -\varepsilon_b$.

```

1  REAL*8 f
2  f=F
3  DO (k=0; x<N; k = k+1)
4    f = k*f

```

will give a completely wrong error as soon as $N \sim 20$. The reason for that is the propagation of the error in f_0 as, being a finite precision number one shall always assume $\hat{F} \equiv (1 + \varepsilon)F$: the error propagates rapidly (however being initially so small it will be hardly noticed) and then explodes in a few iterations corrupting the result in what seems an unpredictable behaviour.

B.3 Accurate Discretization

Typically, computational physicists develop codes to simulate physical processes, which in most cases can be described by a set of partial differential equations (PDEs). While simulating PDEs, to be sure that the developed program is providing useful information, the last check to be performed (after those described in the previous sections) is relation of the simulation results with the physical reality. Indeed, a simulation can be correct and numerically stable, however to be useful it should provide a correct description of the phenomena subject of study.

For example, the PDE can be a particular instance of the Schrödinger equation, which being a first order in time PDE, can be formally represented as

$$\begin{cases} \partial_t \psi(x, t) = \mathcal{L}[\psi(x, t)] \\ \psi(x, 0) = \psi_0(x) \end{cases}, \quad (\text{B.1})$$

where the ∂_t is the partial derivative with respect to the time variable, $\psi_0(x)$ represents the system's initial wave function and the Liouville operator $\mathcal{L} \equiv -i\hat{H}/\hbar$ is build from the system Hamiltonian \hat{H} . Solving (B.1) in all its possible variants is the main subject of this book, and the solution can be formally written (for time-independent Hamiltonians) as

$$\psi(x, t) = e^{\mathcal{L}t} \psi_0(x) \equiv U_t \psi_0(x), \quad (\text{B.2})$$

where U_t is the time-ordered exponential, i.e. the time-evolution operator which maps the wave function at time zero to any following time t . Solving the problem numerically equals to computing the action of the time-evolution operator on the initial wave function. In this apparently simple operation, there is a formal step whose possible consequences shall not be underestimated: the mapping of a continuous space-time $\mathcal{C} \equiv (x, t)$ into a discrete one necessary, due to the floating point arithmetic of computer simulations. That is, setting the precision of the program and choosing the kind of variables, e.g. REAL*4 or REAL*8, we approximate the continuous space time with a discrete one $\mathcal{C} \approx \mathcal{C}_\Delta$, with $\Delta \equiv (\Delta x, \Delta t)$. Typically, the first choice is to employ a linear uniform grid in space time, such that $x_\ell = \Delta x \cdot \ell$ and $t_n = \Delta t \cdot n$ with $\ell = 1, \dots, N_x$ and $n = 1, \dots, N_t$ even though more sophisticated

choices are possible, such as the use of adaptive grids or of finite elements methods [128]. However, independently from the particular grid choice, in general the discretization introduces an error such that

$$\begin{cases} U_t = U_t^\Delta + E^\Delta \\ \psi = \psi^\Delta + e^\Delta \end{cases}; \quad (\text{B.3})$$

where the first addends in the right hand sides of the equations are the discretized versions of the left hand sides and E^Δ and e^Δ the corresponding errors. In conclusion, in presence of a discretization procedure, to avoid obtaining correct but not significant results, the simulation *consistency* and *accuracy* of the results shall always be carefully checked and kept under control.

In particular, the consistency of the simulation with the physical reality requests that

$$\lim_{\Delta \rightarrow 0} U_t^\Delta = U_t \text{ and } \lim_{\Delta \rightarrow 0} E^\Delta = E_0 = 0. \quad (\text{B.4})$$

On the contrary, if $E_0 \neq 0$, then an anomaly is present which typically signals that a symmetry has been broken (parity, time-inversion, etc.) and which shall be recovered. Again, the use of physical invariants might be extremely useful as physical invariants, after discretization, shall become numerical invariants, that is constant quantities within the numerical error.

Finally, the accuracy of the discretization procedure shall be individuated, that is, the scaling of the errors E_Δ with Δ . Indeed, knowing the scaling power p , such that $E_\Delta \sim \Delta^p$, allow us to maximize the resources (time and memory) needed to achieve a desired precision. A paradigmatic example of such scaling is given by the discretization procedure of the first order derivative: while approximating the first order derivative via a discrete grid, the straightforward approach (the *forward derivative FD*) is

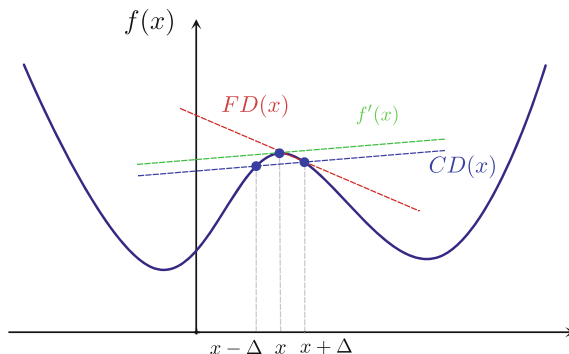
$$FD(x) = \frac{f(x + \Delta) - f(x)}{\Delta}. \quad (\text{B.5})$$

The textbook definition of the derivative assures that with the above definition $\lim_{\Delta \rightarrow 0} FD(x) = f'(x)$. However, the study of the scaling of the error with Δ suggests a better strategy to compute the discrete derivative: by means of a Taylor expansion of the function $f(x) = \sum_n a_n x^n$ and $f'(x) = \sum_n a_n n x^{n-1}$, it is possible to easily show that the error introduced by the discretization is of order $\mathcal{O}(\Delta)$, as $f(x \pm \Delta) = \sum_n a_n (x \pm \Delta)^n = f(x) \pm f'(x)\Delta + \sum_n [n(n-1)x^{n-2}/2]\Delta^2 + \dots$ and thus $FD(x) = f'(x) + \mathcal{O}(\Delta)$. Along the same lines, it can be shown that an alternative discretization procedure might be more favourable: the *central derivative*

$$CD(x) = \frac{f(x + \Delta) - f(x - \Delta)}{2\Delta} \quad (\text{B.6})$$

is such that $CD(x) = f'(x) + \mathcal{O}(\Delta^2)$. The reason for that can be easily grasped by means of the sketch presented in Fig. B.1, and lies again on the fact that the forward

Fig. B.1 The exact derivative (green dashed line), the forward and the central discrete derivative (red and blue dashed lines respectively) according to (B.5) and (B.6) of a function around an extremal point



derivative breaks the problem symmetry with respect to the point the derivative is computed in (the grid point $x + \Delta$ assumes a prominent role with respect to the previous one $x - \Delta$). Restoring the symmetry improves drastically the precision, especially on the functions obeying to such symmetry – the even functions and in particular the lower order ones, quadratic. Notice that the two methods to compute the derivative provided above require exactly the same number of operations, thus there is no reason to use the forward derivative instead of the central one. However, typically this is not the case, and to solve the same problem there exist low- and high-order algorithms with different (decreasing) efficiencies.

Despite the paradigmatic example provide above, the estimation of the errors introduced by the discretization procedure and the search of alternative higher-order schemes is typically highly non trivial for complex algorithms and most of the results are obtained via numerical benchmarking. In general, one should find the right trade-off between high-accuracy methods (whose efficiency typically scale badly with the number of grid points) and low-accuracy but efficient methods.

B.4 Flexibility

Once the first three priorities are satisfied (guaranteeing that the code is correct, numerically stable and describing correctly the physics of interest) the written software has the potential to be used, in some cases also for years. Depending on the the specific cases, it might become a part of another more complex software and eventually might become a standard for a group of users, a community of specialists or even non-expert users. While writing a computer program, one never really knows what a potential future user might want to do this it. Here the future user might be the very same programmer one month or a year later aiming to reuse some part of the already written software; another highly skilled programmer that found your code interesting or useful; a software company that is planning to integrate it in their commercial suite; the programmer's boss; a student that is looking for software to

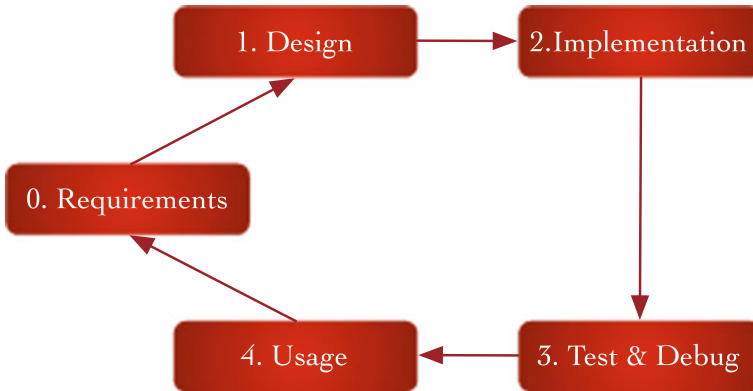


Fig. B.2 The software life cycle: after defining the software requirements, the software design phase enables the implementation phase. The source code is then tested and debugged and finally used. The code usage or changes in the environment will identify the need of additional or different requirements, which will trigger another round of the software lifecycle

develop a master or Ph.D. thesis; or even the IT director of a big software project or research institution that might transform the program in the core of a multi-user multi-purpose research software.

Every software undergoes what is called the *software lifecycle*, depicted in Fig. B.2: it is first described by the *requirements* or specifications that detail the properties of the software and its requested output; the specifications are then translated into a *software design* which assures that the requirements will be met; the design is then implemented in a code in a specific programming language. Finally, the software is tested, debugged and once it is reliable enough, used to produce data. Eventually, while the software usage will push the programmer to either change the specifications and/or the software design, thus closing the lifecycle. The lifecycle can go on even for decades for the more used and famous softwares, e.g. operative systems (Windows, OSX, Linux, etc.), large productivity (e.g. Office) or scientific (AUTOCAD, Matlab, etc.) suites. The changes in the requirements or in the software design might be triggered by different aspects that influence a software development such as, the need for new features (increasing systems size or variety, new protocols or data acquisition, etc.), bug fixes, new hardware, dependencies on other software, and changes of the targeted user (e.g. from experts to general audience).

Software engineering is the science dedicated to design and implement software, and thus to develop tools, concepts and strategies to cope with challenges that software development and its lifecycle pose. Sophisticated strategies and tools might be not relevant while programming small single-user software but they become more and more fundamental with the growing complexity of the software. As modern science is almost always based on the performances of complex software, it is highly probable that any scientists will encounter a large complex software in his academic life. Thus, it is important to know at least some basic concepts of software engineer-

ing. The first of all is that while programming one shall think generally, leave open doors and shall not specialize the software if it is not strictly necessary. If the generalization is not possible as one is writing an highly specialized part of the code, then one shall write a library that can be reused easily in the future. In general, one should aim to a good balance between being abstract and the concrete implementation that necessary are needed to write code. How to achieve that is a matter of experience, however, some general strategies shall be adopted to achieve the best possible result compatible with the developer experience and skills. Adopting them, even though they require an overhead of time while programming, will highly probably result in a huge increase of efficiency in the long run.

In particular, as said before, a software developer performs different tasks at different level of abstractions that goes from the software design to the implementation of the single line of code, and finally the interfaces between parts of program. Hereafter we briefly present some good habits that might help to focus on the important aspects of each of them and to avoid the most common errors.

Software design – When planning a novel software one shall go through the following list of points and try to answer to the related questions: depending on the complexity of the software to be developed some of them shall be carefully addressed and will influence the whole software development, while others might be skipped. In any case, spending some time on them, once again, might save a lot of time in later stages of developments and surely will help in focusing on the important aspects of the programming work while neglecting or at least postponing less important ones. A possible useful checklist is:

1. Specifications:

Program objectives: what are the essential goals of the software, what is not going to be included?

Performances: will the performances be fundamental for the success of the project? Will the program run on a single machine, a cluster or on HPC architecture?

Input/Output: what will be the format of the inputs and outputs and their compatibility (e.g. with other programs)?

2. Data structures and functions: what are the data structures that will be needed? Is there the need to define new ones with their own functions?
3. Libraries and language: Are there already available libraries to perform some needed tasks? Which programming language provides the better tools?
4. Coordination: Is there the need to coordinate different programmers and the software development? Which tools will be used to support them?
5. Portability: Is there the need to have a portable software (i.e. on different platforms and architectures)? If so, how that will be achieved?

For a typical scientific project the answers to many of these questions are eventually quite straightforward: in particular, for most of quantum physics related projects, standard tools such as FORTRAN, C++, python – together with LAPACK (Linear Algebra PACKage [130]) mathematical libraries and some versioning and debugging tools

– will do the job. However, the answering to these questions will force the scientist to better structure the software engineering part of his/her job and to avoid common errors which can undermine the success of months-long hard work.

Interfaces – Once the software structure has been individuated, the data structures and the needed functions have to be coded: this means typically to write numerous parts of code (class definitions, functions, etc.) which shall be later interfaced. This operation is highly non trivial and the strategy used to attack it might influence the final result. Indeed, during this operation, one shall try to exploit as much as possible:

1. **Mathematical abstractions:** in scientific software typically it is straightforward, in quantum physics linear algebra and complex analysis provides the necessary tools to describe the systems of interests.
2. **Information hiding:** Subroutines and functions shall allow easy access to the information they contain in the interfaces and nothing more. This will avoid unclear program parts dependencies, and allow one easier debugging and software update.
3. **Flexibility and modularity:** Abstractions helps to increase the software flexibility as well as modularity which is the key to write good, efficient, reusable and useful software. For example, due to these characteristics, the BLAS (Basic Linear Algebra Subprograms [395]) can and are easily used as a basis for most of the scientific software available.

Hereafter we briefly describe some possible ways to organize the work, together with some comments on their effectiveness [121].

- *Top-down:* The initial design is implemented starting from the higher levels, and refined in increasing detailed and concrete steps until the software is working. This strategy might be followed but it tends to reduce the final software modularization and increase the complexity of the debugging process.
- *Bottom-up:* The data-structures are defined together with the functions operating on them, after that the higher-levels are coded until the whole design is complete. This is an effective strategy which however requires a very careful planning and typically high coding expertise to be exploited at best.
- *Middle-our:* it is possible to follow an intermediate strategy that exploits the good properties of the two previous ones and reduces their risks, however it requires some courage as sometimes it requires to make some steps backwards. However, forcing the coder to refine the previously done job, tends to result in a easier to debug and higher quality code. The steps to follow are:
 1. Identify the overall instructions at the higher level;
 2. Identify the components needed to have a minimal working software;
 3. Code the minimal components to have a working software (in their minimal version);
 4. Test the software as you build it, keep the old tests and update them and re-apply them after changes;
 5. Build the software components of increasing complexity starting from near the bottom level;

6. Once the minimal version of the software is working and debugged, add functionalities to reach the final goal starting from point 1.

As it can be clearly seen, this strategy will force the coder to re-design the data-structure and to rewrite parts of the code while proceeding. This should be possible (even though clearly not painless) in a well-written, test-friendly code and by construction shall result in a information-hiding, flexible, and modular code.

Coding – Once the software is designed and the strategy chosen, the only thing left to do is the coding itself. Independently from the choices made, coding is a matter of knowledge, skill, patience, and experience. Hereafter some good practices are presented to guide the newcomers into the process of mastering this art.

1. A famous quote among programmers is “Premature optimization is the root of evil” which leaves little to add, if not that this is the reason why in this chapter efficiency is left as the very last point.
2. Variables and functions name shall be chosen carefully and properly. Indeed, the function

```
1 FUNCTION f(x)
2 REAL*8 x, f
3 f=x**2
4 END FUNCTION
```

is characterized by short, fast-to-write names which clearly have the advantage to increase writing speed. However, their names do not give any information about their usage or meaning, if embedded in other hundreds of such names, the coder will have hard times to remember all of them. Even more dangerous is the fact that one typically need to automatically find variables and functions in thousand lines of code: an automated search (or even worse a find and replace one, a common practice while restructuring codes) will miserably fail. On the contrary the same function written as

```
1 FUNCTION xsquare(xvariable)
2 REAL*8 xsquare, xvariable
3 xsquare=xvariable**2
4 END FUNCTION
```

slows down the coding without introducing any benefits. Thus, the best solution is to adopt a careful balance between the two extreme examples presented here, which requires the due attention in the definition of the variables name.

3. Commenting the code is always hard time for the coder, however it is a fundamental step for the aforementioned reasons. A good starting point towards good commenting practice is to look the professional libraries (e.g. BLAS libraries) and try to reproduce their standards which allow the user to compile the documentation as html from source code. However, here below some guidelines are reported:
 - A. Comment the interfaces as they will be the most frequent information you will need, as for example:

```

1 c  NORM() computes the norm of the vector a(i) (i=1 to n)
2 c
3 c  a input array
4 c  n length of array
5 c  norm= sqrt(sum a(i)^2) (flag=0)
6 c norm= -1 if n less or equal zero (flag=1)
7 c
8   SUBROUTINE NORM(a,n,flag)
9   ...

```

- B. Declare loop invariants and conditions.
- C. State explicitly warnings for unusual behaviour.
- D. Describe what is done, not how: the implemented algorithm can be explained in the code documentation or simply refer directly to the bibliographic reference.
- E. Avoid redundant comments which increase the code length and waste time twice: to write them and to read them while looking for useful information later on.
- F. Keep the comments updated as outdated comments can drastically slow down the debugging and future reuse of the code than no comments at all.

B.5 Efficiency

The code efficiency is the last priority of the list presented in this chapter as only when all the previous ones are correctly addressed it make sense to worry about it. However, it is still a fundamental issue that might make the difference between a state-of-the-art and an amateur software and thus might open or prevent the possibility of conducting cutting edge research. Software complexity is a highly developed theory and we refer the interested reader to the literature [127]. Here we only briefly recall two results on two different definition of complexity that have implications in computational physics:

Algorithmic complexity is possibly the most widespread definition of complexity and classifies algorithms in terms of the scaling of the resources (time or memory) needed to run an algorithm as a function of the input size N . This definition allows one, beside the fundamental introduction of complexity classes [127, 405], to quantify the performances of different algorithms and distinguish between difficult ones (e.g. scaling exponentially or even super-exponentially with N) and practical ones (polynomial scaling, $\mathcal{O}(N^p)$, with some low power p).

Smoothed complexity has been recently introduced as the scaling of the resources needed to run an algorithm as a function of the input size N *in presence of noise*. This apparent small difference is actually very important for two reasons: the first one is that noise is unavoidable in physical systems and also that finite precision arithmetic can be seen as a form of noisy arithmetic. The

second one is that very recently it has been shown that many important algorithms which solve very general problems do have super-polynomial algorithmic complexity, but are also characterized by polynomial smoothed complexity [406, 407]. This implies that inefficient algorithms in terms of algorithmic complexity can be practically used as, for example, the simplex algorithm to solve linear programming problems [408].

Independently from the complexity measure chosen, in most cases there are different algorithms with different complexities to solve a given problem. The difference in complexity usually shows up in the difficulty and length of the algorithm to be coded, typically with the better scaling algorithm involving longer coding. However, the quest for efficiency calls always for a better scaling algorithm than a slower one, even if the difference is from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log N)$. Moreover, if the program is well organized and written, it should be easy to implement both algorithms, the first one as a quick start and as testbed for the more performing one.

Additional practices that allow boosting efficiency are the use of compiler optimization flags (e.g. the `-O#` and `-fast` options of the `ifort` compiler), the use of a profilers to find program bottlenecks (e.g. `gprof` or similar) and the carefully handling of memory allocation.

The final steps towards efficient computation are the exploitation of parallel architectures and in general of High Performance Computing tools, together with the use of supercomputers and big computational clusters at national and international levels [126]. However, these additional steps might require huge investments which shall be considered only when all other options have been explored and the only remaining possible step to increase efficiency is to invest in huge computational power.

A final considerations is in order: in modern professional scientific computing there is no room for the NIH (Not Invented Here) syndrome. So, there is no room for self-written libraries for standard algorithms (e.g. linear algebra). It is much more effective to spend time to learn how to use professional available libraries (e.g. BLAS, LAPACK, etc...) than starting from scratch. Indeed, usually “95% of execution time is spent in 5% of the code” and for scientific (quantum simulation) software this time is spent in basic (linear algebra) operations. Thus, the most efficient strategy is to rely on the work of years (in some cases decades!) of the maximal experts in the field: the probabilities to improve their codes without spending a similar effort in coding with similar skills is practically zero (unless of course you are part or aim to become part of the small community of high-level experts that are developing those libraries).

B.6 Exercises

1. Matrix multiplication is often the bottleneck of linear algebra computations.
 - a. Write explicitly the matrix-matrix multiplication loop in two different orders (column-row and viceversa):

- b. Use the FORTRAN intrinsic function.
- c. Use the LAPACK subroutine

Increase the matrix size and use the FORTRAN Function CPUTIME to monitor the code performance.

2. Include documentation, comments, pre- and post- conditions, error handling, and checkpoints in the previous exercise.

References

1. D.J. Thouless, *The Quantum Mechanics of Many-Body Systems* (Courier Dover Publications, New York, 2013)
2. M.D. Schwartz, *Quantum Field Theory and the Standard Model* (Cambridge University Press, Cambridge, MA, 2014)
3. P.A. Lee, N. Nagaosa, X.-G. Wen, Doping a Mott insulator: Physics of high-temperature superconductivity. *Rev. Mod. Phys.* **78**, 17–85 (2006)
4. A. Mann, High-temperature superconductivity at 25: Still in suspense. *Nature* **475**, 280–282 (2011)
5. A. Szabo, N. S. Ostlund, *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory* (1996)
6. J.P. Dowling, G.J. Milburn, Quantum technology: The second quantum revolution. *Phil. Trans. R. Soc. Lond. A* **361**, 1655–1674 (2003)
7. M.A. Nielsen, I.L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, MA, 2010)
8. G. Kurizki, P. Bertet, Y. Kubo, K. Mølmer, D. Petrosyan, P. Rabl, J. Schmiedmayer, Quantum technologies with hybrid systems. *Proc. Natl. Acad. Sci. U.S.A.* **112**, 3866–3873 (2015)
9. Y.-C. Cheng, G.R. Fleming, Dynamics of light harvesting in photosynthesis. *Annu. Rev. Phys. Chem.* **60**, 241–62 (2009)
10. G.D. Scholes, G.R. Fleming, A. Olaya-Castro, R. van Grondelle, Lessons from nature about solar light harvesting. *Nat. Chem.* **3**, 763–74 (2011)
11. X.-G. Wen, Topological orders in rigid states. *Inter. Jour. Mod. Phys. B* **4**, 239–271 (1990)
12. B. Sutherland, *Beautiful Models: 70 Years of Exactly Solved Quantum Many-Body Problems* (World Scientific, Singapore, 2004)
13. J. Zhang, R. Dong, Exact diagonalization: The Bose-Hubbard model as an example. *Eur. Jour. Phys.* **31**, 591 (2010)
14. L.P. Kadanoff, More is the same; phase transitions and mean field theories. *J. Stat. Phys.* **137**, 777–797 (2009)
15. C. Hätting, Electronic structure : Hartree-Fock and correlation methods. *Multiscale Simul. Methods Mol. Sci.* **42**, 77–120 (2009)
16. C.F. Fischer, General Hartree-Fock program. *Comput. Phys. Commun.* **43**, 355–365 (1987)
17. V.L. Ginzburg, *Fiz. Tverdogo Tela*, **2**, 2031 (1960). English Transl. *Soviet Physics-Solid State* **2**, 1824 (1960)

18. R.Y. Rubinstein, D.P. Kroese, *Simulation and the Monte Carlo Method* (Wiley, New York, 2011)
19. E.Y. Loh, J.E. Gubernatis, R.T. Scalettar, S.R. White, D.J. Scalapino, R.L. Sugar, Sign problem in the numerical simulation of many-electron systems. *Phys. Rev. B* **41**, 9301–9307 (1990)
20. M. Schiró, M. Fabrizio, Real-time diagrammatic Monte Carlo for nonequilibrium quantum transport. *Phys. Rev. B* **79**, 153302 (2009)
21. M. Troyer, U.-J. Wiese, Computational complexity and fundamental limitations to fermionic quantum Monte Carlo simulations. *Phys. Rev. Lett.* **94**, 170201 (2005)
22. K. Wilson, The renormalization group: Critical phenomena and the Kondo problem. *Rev. Mod. Phys.* **47**, 773–840 (1975)
23. S.R. White, Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.* **69**, 2863–2866 (1992)
24. U. Schollwöck, The density-matrix renormalization group in the age of matrix product states. *Ann. Phys. (NY)* **326**, 96–192 (2011)
25. R. Orus, A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Ann. Phys.* **349**, 117 (2014)
26. I. Peschel, X. Wang, M. Kaulke, K. Hallberg (eds.), *Density-Matrix Renormalization- a New Numerical Method in Physics*, Lecture Notes in Physics (Springer, Berlin, 1999)
27. K.A. Hallberg, New trends in density matrix renormalization. *Adv. Phys.* **55**, 477–526 (2006)
28. S. Sachdev, Viewpoint: Tensor networks-a new tool for old problems. *Physics* **2**, 90 (2009)
29. I.P. McCulloch, From density-matrix renormalization group to matrix product states. *J. Stat. Mech. Theory Exp.* **2007**, P10014–P10014 (2007)
30. R. Orús, Advances on tensor network theory: Symmetries, fermions, entanglement, and holography. *Eur. Phys. Jour. B* **87**, 1–18 (2014)
31. P. Silvi, F. Tschirsich, M. Gerster, J. Jünemann, D. Jaschke, M. Rizzi, and S. Montangero, *The Tensor Networks Anthology: Simulation Techniques for Manybody Quantum Lattice Systems*, <http://arxiv.org/abs/1710.03733>
32. M. Srednicki, Entropy and area. *Phys. Rev. Lett.* **71**, 666–669 (1993)
33. T. Xiang, J. Lou, Z. Su, Two-dimensional algorithm of the density-matrix renormalization group. *Phys. Rev. B* **64**, 104414 (2001)
34. O. Legeza, J. Sólyom, Quantum data compression, quantum information generation, and the density-matrix renormalization-group method. *Phys. Rev. B* **70**, 205118 (2004)
35. G. Vidal, J.I. Latorre, E. Rico, A. Kitaev, Entanglement in quantum critical phenomena. *Phys. Rev. Lett.* **90**, 227902 (2003)
36. M.B. Plenio, J. Eisert, J. Dreifbig, M. Cramer, Entropy, entanglement, and area: Analytical results for harmonic lattice systems. *Phys. Rev. Lett.* **94**, 060503 (2005)
37. J. Eisert, M. Cramer, M.B. Plenio, Area laws for the entanglement entropy. *Rev. Mod. Phys.* **82**, 277–306 (2010)
38. M. Lubasch, J.I. Cirac, M.-C. Bañuls, Algorithms for finite projected entangled pair states. *Phys. Rev. B* **90**, 064425 (2014)
39. F. Verstraete, J.J. García-Ripoll, J.I. Cirac, Matrix product density operators: Simulation of finite-temperature and dissipative systems. *Phys. Rev. Lett.* **93**, 207204 (2004)
40. M. Kliesch, D. Gross, J. Eisert, Matrix-product operators and states: NP-hardness and undecidability. *Phys. Rev. Lett.* **113**, 160503 (2014)
41. L. Tagliacozzo, G. Evenbly, G. Vidal, Simulation of two-dimensional quantum systems using a tree tensor network that exploits the entropic area law. *Phys. Rev. B* **80**, 235127 (2009)
42. G. Evenbly, G. Vidal, Class of highly entangled many-body states that can be efficiently simulated. *Phys. Rev. Lett.* **112**, 240502 (2014)
43. M. Gerster, P. Silvi, M. Rizzi, R. Fazio, T. Calarco, S. Montangero, Unconstrained tree tensor network: An adaptive gauge picture for enhanced performance. *Phys. Rev. B* **90**, 125154 (2014)
44. G. Vidal, Entanglement renormalization. *Phys. Rev. Lett.* **99**, 220405 (2007)
45. J. Eisert, *Entanglement and Tensor Network States*, *Modelling and Simulation* **3**, 520 (2013)

46. S. Östlund, S. Rommer, Thermodynamic limit of density matrix renormalization. *Phys. Rev. Lett.* **75**, 3537–3540 (1995)
47. F. Verstraete, D. Porras, J.I. Cirac, Density matrix renormalization group and periodic boundary conditions: A quantum information perspective. *Phys. Rev. Lett.* **93**, 227205 (2004)
48. A.J. Daley, C. Kollath, U. Schollwoeck, G. Vidal, Time-dependent densitymatrix renormalization-group using adaptive effective Hilbert spaces. *J. Stat. Mech.* **2004**, P04005 (2004)
49. D. Perez-Garcia, F. Verstraete, M.M. Wolf, J.I. Cirac, Matrix product state representations. *Quantum Inf. Comput.* **7**, 401 (2007)
50. F. Verstraete, J.J. Garcia-Ripoll, J.I. Cirac, Matrix product density operators: Simulation of finite-temperature and dissipative systems. *Phys. Rev. Lett.* **93**, 207204 (2004)
51. R. Orús, G. Vidal, Infinite time-evolving block decimation algorithm beyond unitary evolution. *Phys. Rev. B* **78**, 155117 (2008)
52. G.M. Crosswhite, A.C. Doherty, G. Vidal, Applying matrix product operators to model systems with long-range interactions. *Phys. Rev. B* **78**, 035116 (2008)
53. G. Vidal, Efficient classical simulation of slightly entangled quantum computations. *Phys. Rev. Lett.* **91**, 147902 (2003)
54. J. Haegeman, J.I. Cirac, T.J. Osborne, I. Pižorn, H. Verschelde, F. Verstraete, Time-dependent variational principle for quantum lattices. *Phys. Rev. Lett.* **107**, 070601 (2011)
55. E. Rico, T. Pichler, M. Dalmonte, P. Zoller, S. Montangero, Tensor networks for lattice gauge theories and atomic quantum simulation. *Phys. Rev. Lett.* **112**, 201601 (2014)
56. S. Kühn, J.I. Cirac, M.-C. Bañuls, Quantum simulation of the Schwinger model: A study of feasibility. *Phys. Rev. A* **90**, 042305 (2014)
57. T. Pichler, M. Dalmonte, E. Rico, P. Zoller, S. Montangero, Real-time dynamics in $u(1)$ lattice gauge theories with tensor networks. *Phys. Rev. X* **6**, 011023 (2016)
58. M.C. Bañuls, K. Cichy, J.I. Cirac, K. Jansen, S. Kühn, Density induced phase transitions in the schwinger model: A study with matrix product states. *Phys. Rev. Lett.* **118**, 071601 (2017)
59. P. Silvi, E. Rico, T. Calarco, S. Montangero, Lattice gauge tensor networks. *New J. Phys.* **16**, 103015 (2014)
60. Spectral calculations with matrix product states, M.C. Bañuls, K. Cichy, J. Ignacio Cirac, K. Jansen, S. Kühn, Efficient basis formulation for $(1 + 1)$ -dimensional $SU(2)$ lattice gauge theory. *Phys. Rev. X* **7**, 041046 (2017)
61. K. Zapp, R. Orús, Tensor network simulation of QED on infinite lattices: Learning from $(1+1)d$, and prospects for $(2+1)d$. *Phys. Rev. D* **95**, 114508 (2017)
62. B. Buyens, S. Montangero, J. Haegeman, F. Verstraete, K. Van Acoleyen, Finite-representation approximation of lattice gauge theories at the continuum limit with tensor networks. *Phys. Rev. D* **95**, 094509 (2017)
63. M.C. Bañuls, K. Cichy, J.I. Cirac, K. Jansen, S. Kühn, Density induced phase transitions in the schwinger model: A study with matrix product states. *Phys. Rev. Lett.* **118**, 071601 (2017)
64. M.C. Bañuls, K. Cichy, J. Ignacio Cirac, K. Jansen, S. Kühn, H. Saito, Towards overcoming the Monte Carlo sign problem with tensor networks, EPJ Web Conf. 137, ed. by Y. Foka, N. Brambilla, V. Kovalenko, 04001 (2017)
65. B. Buyens, J. Haegeman, H. Verschelde, F. Verstraete, K. Van Acoleyen, Confinement and string breaking for QED_2 in the Hamiltonian picture. *Phys. Rev. X* **6**, 041040 (2016)
66. M.C. Bañuls, K. Cichy, K. Jansen, H. Saito, Chiral condensate in the Schwinger model with matrix product operators. *Phys. Rev. D* **93**, 094512 (2016)
67. T. Osborne, *Lattice Gauge Theories and Tensor Networks*, <https://github.com/tobiasosborne/Latticegaugetheoryandtensornetworks>
68. S. Kühn, E. Zohar, J.I. Cirac, M.C. Bañuls, Non-Abelian string breaking phenomena with matrix product states. *J. High Energy Phys.* **2015**, 130 (2015)
69. J. Haegeman, K. Van Acoleyen, N. Schuch, J.I. Cirac, F. Verstraete, Gauging quantum states: From global to local symmetries in many-body systems. *Phys. Rev. X* **5**, 011024 (2015)
70. L. Tagliacozzo, G. Vidal, Entanglement renormalization and gauge symmetry. *Phys. Rev. B* **83**, 115127 (2011)

71. L. Tagliacozzo, A. Celi, M. Lewenstein, Tensor networks for lattice gauge theories with continuous groups. *Phys. Rev. X* **4**, 041024 (2014)
72. T. Byrnes, P. Sriganesh, R. Bursill, C. Hamer, Density matrix renormalization group approach to the massive Schwinger model. *Phys. Rev. D* **66**, 013002 (2002)
73. E. Ercolessi, P. Facchi, G. Magnifico, S. Pascazio, F. V. Pepe, Phase Transitions in Z_n Gauge Models: Towards Quantum Simulations of the Schwinger- Weyl QED <http://arxiv.org/abs/1705.11047>
74. B. Buyens, J. Haegeman, K. Van Acoleyen, H. Verschelde, F. Verstraete, Matrix product states for gauge field theories. *Phys. Rev. Lett.* **113**, 091601 (2014)
75. M. Bañuls, K. Cichy, J. Cirac, K. Jansen, The mass spectrum of the Schwinger model with matrix product states. *J. High Energy Phys.* **2013**, 158 (2013)
76. S. Morita, Y. Ozeki, H. Nishimori, Gauge theory for quantum spin glasses. *J. Phys. Soc. Japan* **75**, 014001 (2006)
77. M.C. Bañuls, K. Cichy, J.I. Cirac, K. Jansen, H. Saito, Matrix product states for lattice field theories, in Proceedings of 31st International Symposium Lattice F. Theory, 332 (2013)
78. U.-J. Wiese, Towards quantum simulating QCD. *Nucl. Phys. A* **931**, 246–256 (2014)
79. M. Dalmonte, S. Montangero, Lattice gauge theory simulations in the quantum information era. *Contemp. Phys.* **7514**, 1–25 (2016)
80. K.G. Wilson, The renormalization group: Critical phenomena and the Kondo problem. *Rev. Mod. Phys.* **47**, 773–840 (1975)
81. I. Montvay, G. Muenster, *Quantum Fields on a Lattice* (Cambridge University Press, Cambridge, MA, 1994)
82. M. Creutz, *Quarks, Gluons and Lattices* (Cambridge University Press, Cambridge, MA, 1997)
83. T. DeGrand, C. DeTar, *Lattice Methods for Quantum Chromodynamics* (World Scientific, Singapore, 2006)
84. C. Gattringer, C.B. Lang, *Quantum Chromodynamics on the Lattice* (Springer, Berlin, 2010)
85. E. Fradkin, J.E. Moore, *Phys. Rev. Lett.* **97**, 050404 (2006)
86. J. Kogut, L. Susskind, Hamiltonian formulation of Wilson’s lattice gauge theories. *Phys. Rev. D* **11**, 395–408 (1975)
87. J. Kogut, *An introduction to lattice gauge theory and spin systems* *Rev. Mod. Phys.*, 31659 (1979)
88. D. Horn, Finite matrix models with continuous local gauge invariance. *Phys. Lett. B* **100**, 149 (1981)
89. P. Orland, D. Rohrlich, Lattice gauge magnets: Local isospin from spin. *Nucl. Phys. B* **338**, 647 (1990)
90. S. Chandrasekharan, U.J. Wiese, Quantum link models : A discrete approach to gauge theories. *Nucl. Phys. B* **492**, 455–471 (1997)
91. B. Schlittgen, U.-J. Wiese, Low-energy effective theories of quantum spin and quantum link models. *Phys. Rev. D* **63**, 085007 (2001)
92. C. Lacroix, P. Mendels, and F. Mila, (eds.), *Introduction to Frustrated Magnetism* (Springer Series in Solid-State Sciences Vol. 164, 2010)
93. R.P. Feynman, Simulating physics with computers. *Int. J. Theor. Phys.* **21**, 467–488 (1982)
94. J.I. Cirac, P. Zoller, Goals and opportunities in quantum simulation. *Nat. Phys.* **8**, 264–266 (2012)
95. I. Bloch, J. Dalibard, S. Nascimbene, Quantum simulations with ultracold quantum gases. *Nat. Phys.* **8**, 267–276 (2012)
96. D. Jaksch, P. Zoller, The cold atom Hubbard toolbox. *Ann. Phys. (NY)* **315**, 52–79 (2005)
97. E. Lieb, T. Schultz, D. Mattis, Two soluble models of an antiferromagnetic chain. *Ann. Phys.* **466**, 407–466 (1961)
98. E. Lieb, *The Hubbard Model: Some Rigorous Results and Open Problems*, <http://arxiv.org/abs/cond-mat/9311033>
99. H.P. Büchler, M. Hermele, S.D. Huber, M.P.A. Fisher, P. Zoller, Atomic quantum simulator for lattice gauge theories and ring exchange models. *Phys. Rev. Lett.* **95**, 40402 (2005)

100. H. Weimer, M. Müller, I. Lesanovsky, P. Zoller, H.P. Büchler, A Rydberg quantum simulator. *Nat. Phys.* **6**, 382–388 (2010)
101. L. Tagliacozzo, A. Celi, P. Orland, M.W. Mitchell, M. Lewenstein, Simulation of non-Abelian gauge theories with optical lattices. *Nat. Commun.* **4**, 2615 (2013)
102. A.W. Glaetzle, M. Dalmonte, R. Nath, I. Rousochatzakis, R. Moessner, P. Zoller, Quantum spin-ice and dimer models with Rydberg atoms. *Phys. Rev. X* **4**, 041037 (2014)
103. P. Schauss, J. Zeiher, T. Fukuhara, S. Hild, M. Cheneau, T. Macri, T. Pohl, I. Bloch, C. Gross, Crystallization in Ising quantum magnets. *Science* **347**, 1455–1458 (2015)
104. H. Bernien, S. Schwartz, A. Keesling, H. Levine, A. Omran, H. Pichler, S. Choi, A.S. Zibrov, M. Endres, M. Greiner, V. Vuletić, M.D. Lukin, Probing many-body dynamics on a 51-atom quantum simulator. *Nature* **551**, 579–584 (2017)
105. J.I. Cirac, P. Maraner, J.K. Pachos, Cold atom simulation of interacting relativistic quantum field theories. *Phys. Rev. Lett.* **105**, 190403 (2010)
106. J. Casanova, L. Lamata, I.L. Egusquiza, R. Gerritsma, C.F. Roos, J.J. García-Ripoll, E. Solano, Quantum simulation of quantum field theories in trapped ions. *Phys. Rev. Lett.* **107**, 260501 (2011)
107. S.P. Jordan, K.S.M. Lee, J. Preskill, Quantum algorithms for quantum field theories. *Science* **80**(336), 1130–1133 (2012)
108. D. Banerjee, M. Dalmonte, M. Müller, E. Rico, P. Stebler, U.-J. Wiese, P. Zoller, Atomic quantum simulation of dynamical gauge fields coupled to fermionic matter: From string breaking to evolution after a quench. *Phys. Rev. Lett.* **109**, 1–5 (2012)
109. D. Banerjee, M. Bögli, M. Dalmonte, E. Rico, P. Stebler, U.-J. Wiese, P. Zoller, Atomic quantum simulation of $U(N)$ and $SU(N)$ non-Abelian lattice gauge theories. *Phys. Rev. Lett.* **110**, 125303 (2013)
110. S. Barrett, K. Hammerer, S. Harrison, T.E. Northup, T.J. Osborne, Simulating quantum fields with cavity QED. *Phys. Rev. Lett.* **110**, 1–6 (2013)
111. L. Tagliacozzo, A. Celi, A. Zamora, M. Lewenstein, Optical Abelian lattice gauge theories. *Ann. Phys. (NY)* **330**, 160–191 (2013)
112. D. Marcos, P. Rabl, E. Rico, P. Zoller, Superconducting circuits for quantum simulation of dynamical gauge fields. *Phys. Rev. Lett.* **111**, 110504 (2013)
113. P. Hauke, D. Marcos, M. Dalmonte, P. Zoller, Quantum simulation of a lattice Schwinger model in a chain of trapped ions. *Phys. Rev. X* **3**, 041018 (2013)
114. E. Zohar, M. Burrello, Formulation of lattice gauge theories for quantum simulations. *Phys. Rev. D* **91**, 054506 (2015)
115. S. Notarnicola, E. Ercolessi, P. Facchi, G. Marmo, S. Pascazio, F.V. Pepe, Discrete Abelian gauge theories for quantum simulations of QED. *J. Phys. A Math. Theor.* **48**, 30FT01 (2015)
116. E. Zohar, A. Farace, B. Reznik, J.I. Cirac, Digital quantum simulation of \mathbb{Z}_2 lattice gauge theories with dynamical fermionic matter. *Phys. Rev. Lett.* **118**, 070501 (2017)
117. E. Zohar, J.I. Cirac, B. Reznik, Quantum simulations of lattice gauge theories using ultracold atoms in optical lattices. *Reports Prog. Phys.* **79**, 014401 (2016)
118. V. Kasper, F. Hebenstreit, F. Jendrzejewski, M.K. Oberthaler, J. Berges, Implementing quantum electrodynamics with ultracold atomic systems. *New J. Phys.* **19**, 023030 (2017)
119. E.A. Martinez, C.A. Muschik, P. Schindler, D. Nigg, A. Erhard, M. Heyl, P. Hauke, M. Dalmonte, T. Monz, P. Zoller, R. Blatt, Real-time dynamics of lattice gauge theories with a few-qubit quantum computer. *Nature* **534**, 516–519 (2016)
120. S. Succi, *An Introduction to Computational Physics* (Scuola Normale Superiore, 2002)
121. S. Oliveira, D. Stewart, *Writing Scientific Software - a Guide to Good Style* (Cambridge University Press, Cambridge, MA, 2006)
122. W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Numerical Recipes in Fortran; The Art of Scientific Computing*, 2nd edn. (Cambridge University Press, New York, NY, 1993)
123. W. Gibbs, *Computation in Modern Physics*, 3rd edn. (World Scientific Publishing Company, Singapore, 2006)
124. R. Van de Geijn, E. Quintana-Ortí, *The Science of Programming Matrix Computations* (na, 2008)

125. D.P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods* (Academic Press, London, 1982)
126. K. Dowd, C. Severance, *High Performance Computing, A Nutshell handbook* (O'Reilly, 1998)
127. K. Mehlhorn, P. Sanders, *Algorithms and Data Structures: The Basic Toolbox, SpringerLink: Springer e-Books* (Springer, Berlin, 2008)
128. C. Johnson, *Numerical Solution of Partial Differential Equations by the Finite Element Method, Dover Books on Mathematics Series* (Dover Publications, Incorporated, New York, 2012)
129. S.C. Brenner, C. Carstensen, *Finite element methods, in Encyclopedia of Computational Mechanics* (Wiley, New York, 2004)
130. E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, D. Sorensen, *Lapack Users' Guide*, 3rd edn, Software, Environments, and Tools (Society for Industrial and Applied Mathematics (SIAM), 1999)
131. C. Lanczos, An iterative method for the solution of the eigenvalue problem of linear differential and integral. *J. Res. Nat. Bureau Standards* **45**, 2133 (1950)
132. J. Liesen, Z. Strakos, *Krylov Subspace Methods: Principles and Analysis, Numerical Mathematics and Scientific Computation* (OUP, Oxford, 2012)
133. G.W. Anderson, A. Guionnet, O. Zeitouni, *An Introduction to Random Matrices* (Cambridge University Press, Cambridge, MA, 2009)
134. C. Froese Fischer, General Hartree-Fock program. *Comput. Phys. Commun.* **43**, 355–365 (1987)
135. C. Cohen-Tannoudji, B. Diu, F. Laloë, *Quantum Mechanics* (Wiley, New York, NY, 1977)
136. R. Bulla, T. Costi, T. Pruschke, Numerical renormalization group method for quantum impurity systems. *Rev. Mod. Phys.* **80**, 395–450 (2008)
137. L. Carr, *Understanding Quantum Phase Transitions* (CRC Press, Boca Raton, FL, 2010)
138. E. Barouch, B.M. McCoy, Statistical mechanics of the XY model. II. Spincorrelation functions. *Phys. Rev. A* **3**, 786–804 (1971)
139. P. Pfeuty, The one-dimensional Ising model with a transverse field. *Ann. Phys.* **57**, 79–90 (1970)
140. L. Amico, R. Fazio, A. Osterloh, V. Vedral, Entanglement in many-body systems. *Rev. Mod. Phys.* **80**, 517–576 (2008)
141. G. Mussardo, *Statistical Field Theory* (Oxford University Press, Oxford, 2009)
142. D. Amit, V. Martin-Mayor, *Field Theory, the Renormalization Group, and Critical Phenomena: Graphs to Computers*, 3rd edn. (World Scientific, Singapore, 2005)
143. I. Peschel, X. Wang, M. Kaulke, K. Hallberg, *Density-Matrix Renormalization—a New Numerical Method in Physics: Lectures of a Seminar and Workshop Held at the max-planck-institut für physik komplexer systeme, Dresden, Germany, August 24th to September 18th, 1998*, Lecture Notes in Physics (Springer, Berlin, 2014)
144. S.R. White, Density-matrix algorithms for quantum renormalization groups. *Phys. Rev. B* **48**, 10345–10356 (1993)
145. U. Schollwöck, The density-matrix renormalization group. *Rev. Mod. Phys.* **77**, 259–315 (2005)
146. A.L. Malvezzi, An introduction to numerical methods in low-dimensional quantum systems. *Brazilian J. Phys.* **33**, 36 (2003)
147. A. Royer, Wigner function in Liouville space: A canonical formalism. *Phys. Rev. A* **43**, 44–56 (1991)
148. I. Bengtsson, K. Życzkowski, *Geometry of Quantum States: An Introduction to Quantum Entanglement* (Cambridge University Press, Cambridge, MA, 2007)
149. E.C.G. Sudarshan, P.M. Mathews, J. Rau, Stochastic dynamics of quantum-mechanical systems. *Phys. Rev.* **121**, 920–924 (1961)
150. A. Jamiolkowski, Linear transformations which preserve trace and positive semidefiniteness of operators. *Rep. Math. Phys.* **3**, 275–278 (1972)
151. M.-D. Choi, Completely positive linear maps on complex matrices. *Lin. Alg. Appl.* **10**, 285 (1975)

152. P. Arrighi, C. Patricot, On quantum operations as quantum states. *Ann. Phys. NY* **311**, 26–52 (2004)
153. S. Lang, *Complex Analysis, Graduate Texts in Mathematics* (Springer, New York, 2013)
154. V. Strassen, Gaussian elimination is not optimal. *Numer. Math.* **13**, 354–356 (1969)
155. S. Östlund, S. Rommer, Thermodynamic limit of density matrix renormalization. *Phys. Rev. Lett.* **75**, 3537–3540 (1995)
156. B. Pirvu, V. Murg, J.I. Cirac, F. Verstraete, Matrix product operator representations. *New J. Phys.* **12**, 025012 (2010)
157. F. Fröwis, V. Nebendahl, W. Dür, Tensor operators: Constructions and applications for long-range interaction systems. *Phys. Rev. A* **81**, 062337 (2010)
158. J. Cui, J.I. Cirac, M.C. Bañuls, Variational matrix product operators for the steady state of dissipative quantum systems. *Phys. Rev. Lett.* **114**, 220601 (2015)
159. E. Mascarenhas, H. Flayac, V. Savona, Matrix-product-operator approach to the nonequilibrium steady state of driven-dissipative quantum arrays. *Phys. Rev. A* **92**, 022116 (2015)
160. Y.-Y. Shi, L.-M. Duan, G. Vidal, Classical simulation of quantum manybody systems with a tree tensor network. *Phys. Rev. A* **74**, 022320 (2006)
161. A.J. Ferris, Area law and real-space renormalization. *Phys. Rev. B* **87**, 125139 (2013)
162. M. Gerster, M. Rizzi, F. Tschirsich, P. Silvi, R. Fazio, S. Montangero, Superfluid density and quasi-long-range order in the one-dimensional disordered Bose-Hubbard model. *New J. Phys.* **18**, 015015 (2016)
163. V. Murg, F. Verstraete, Ö. Legeza, R.M. Noack, Simulating strongly correlated quantum systems with tree tensor networks. *Phys. Rev. B* **82**, 205105 (2010)
164. P. Pippian, S.R. White, H.G. Evertz, Efficient matrix product state method for periodic boundary conditions. *Phys. Rev. B* **81**, 081103 (2008)
165. Limitations and constructive methods, G.D. las Cuevas, N. Schuch, D. Pérez-García, J. Ignacio Cirac, Purifications of multipartite states. *New J. Phys.* **15**, 123021–123026 (2013)
166. A.H. Werner, D. Jaschke, P. Silvi, M. Kliesch, T. Calarco, J. Eisert, S. Montangero, Sup. Mat. Positive tensor network approach for simulating open quantum many-body systems. *Phys. Rev. Lett.* **116**, 237201 (2016)
167. F. Verstraete, M.M. Wolf, D. Perez-Garcia, J.I. Cirac, Criticality, the area law, and the computational power of projected entangled pair states. *Phys. Rev. Lett.* **96**, 220601 (2006)
168. N. Schuch, M.M. Wolf, F. Verstraete, J.I. Cirac, Computational complexity of projected entangled pair states. *Phys. Rev. Lett.* **98**, 140506 (2007)
169. G. Vidal, Entanglement renormalization. *Phys. Rev. Lett.* **99**, 220405 (2007)
170. G. Vidal, Class of quantum many-body states that can be efficiently simulated. *Phys. Rev. Lett.* **101**, 110501 (2008)
171. M. Rizzi, S. Montangero, G. Vidal, Simulation of time evolution with multiscale entanglement renormalization ansatz. *Phys. Rev. A* **77**, 052328 (2008)
172. G. Evenbly, G. Vidal, Class of highly entangled many-body states that can be efficiently simulated. *Phys. Rev. Lett.* **112**, 240502 (2014)
173. S. Anders, H.J. Briegel, W. Dür, A variational method based on weighted graph states. *New J. Phys.* **9**, 361 (2007)
174. F. Mezzacapo, N. Schuch, M. Boninsegni, J.I. Cirac, Ground-state properties of quantum many-body systems: Entangled-plaquette states and variational Monte Carlo. *New J. Phys.* **11**, 083026 (2009)
175. N. Schuch, M.M. Wolf, F. Verstraete, J.I. Cirac, Simulation of quantum many-body systems with strings of operators and monte carlo tensor contractions. *Phys. Rev. Lett.* **100**, 040501 (2008)
176. G. Evenbly, Hyperinvariant tensor networks and holography. *Phys. Rev. Lett.* **119**, 1–5 (2017)
177. M. Suzuki, Fractal decomposition of exponential operators with applications to many-body theories and monte carlo simulations. *Phys. Lett. A* **146**, 319–323 (1990)
178. G. De Chiara, M. Rizzi, D. Rossini, S. Montangero, Density matrix renormalization group for dummies. *J. Comput. Theor. Nanosci.* **5**, 1277–1288 (2008)

179. J. Huyghebaert, H. De Raedt, Product formula methods for time dependent Schrödinger problems. *J. Phys. A. Math. Gen.* **23**, 5777 (1990)
180. D. Poulin, A. Qarry, R. Somma, F. Verstraete, Quantum simulation of time-dependent Hamiltonians and the convenient illusion of Hilbert space. *Phys. Rev. Lett.* **106**, 170501 (2011)
181. S.R. White, A.E. Feiguin, Real-time evolution using the density matrix renormalization group. *Phys. Rev. Lett.* **93**, 076401 (2004)
182. G. Vidal, Efficient simulation of one-dimensional quantum many-body systems. *Phys. Rev. Lett.* **93**, 040502 (2004)
183. T.J. Osborne, *A Renormalisation-Group Algorithm for Eigenvalue Density Functions of Interacting Quantum Systems*, <https://arxiv.org/abs/condmat/0605194>
184. A. Werner, D. Jaschke, P. Silvi, M. Kliesch, T. Calarco, J. Eisert, S. Montangero, Positive tensor network approach for simulating open quantum many-body systems. *Phys. Rev. Lett.* **116**, 237201 (2016)
185. F. Fröwis, V. Nebendahl, W. Dür, Tensor operators: Constructions and applications for long-range interaction systems. *Phys. Rev. A* **81**, 062337 (2010)
186. M.P. Zaletel, R.S.K. Mong, C. Karrasch, J.E. Moore, F. Pollmann, Timeevolving a matrix product state with long-ranged interactions. *Phys. Rev. B* **91**, 165112 (2015)
187. M. Rizzi, S. Montangero, G. Vidal, Simulation of time evolution with multiscale entanglement renormalization ansatz. *Phys. Rev. A* **77**, 052328 (2008)
188. J. Haegeman, C. Lubich, I. Oseledets, B. Vandereycken, F. Verstraete, Unifying time evolution and optimization with matrix product states. *Phys. Rev. B- Condens. Matter Mater. Phys.* **94**, 1–10 (2016)
189. J. Haegeman, J.I. Cirac, T.J. Osborne, I. Pizorn, H. Verschelde, F. Verstraete, Time-dependent variational principle for quantum lattices. *Phys. Rev. Lett.* **107**, 165116 (2011)
190. A.J. Ferris, Fourier transform for fermionic systems and the spectral tensor network. *Phys. Rev. Lett.* **113**, 010401 (2014)
191. G. Ehlers, J. Sólyom, Ö. Legeza, R.M. Noack, Entanglement structure of the Hubbard model in momentum space. *Phys. Rev. B* **92**, 235116 (2015)
192. G. Ehlers, S.R. White, R.M. Noack, Hybrid-space density matrix renormalization group study of the doped two-dimensional Hubbard model. *Phys. Rev. B* **95**, 1–16 (2017)
193. F. Schrodli, P. Silvi, F. Tschirsich, R. Fazio, S. Montangero, Density of states of many-body quantum systems from tensor networks. *Phys. Rev. B* **96**, 094303 (2017)
194. A.J. Daley, Quantum trajectories and open many-body quantum systems. *Adv. Phys.* **63**, 77–149 (2014)
195. A. Molnar, N. Schuch, F. Verstraete, J.I. Cirac, Approximating Gibbs states of local Hamiltonians efficiently with projected entangled pair states. *Phys. Rev. B* **91**, 045138 (2015)
196. S.R. White, Minimally entangled typical quantum states at finite temperature. *Phys. Rev. Lett.* **102**, 190601 (2009)
197. M.C. Bañuls, K. Cichy, J.I. Cirac, K. Jansen, H. Saito, Thermal evolution of the Schwinger model with matrix product operators. *Phys. Rev. D* **92**, 034519 (2015)
198. B.-B. Chen, Y.-J. Liu, Z. Chen, W. Li, Series-expansion thermal tensor network approach for quantum lattice models. *Phys. Rev. B* **95**, 161104 (2017)
199. S. Iblisdir, R. Orús, J. Latorre, Matrix product states algorithms and continuous systems. *Phys. Rev. B* **75**, 104305 (2007)
200. F. Verstraete, J.I. Cirac, Continuous matrix product states for quantum fields. *Phys. Rev. Lett.* **104**, 190405 (2010)
201. M. Ganahl, J. Rincón, G. Vidal, Continuous matrix product states for quantum fields: An energy minimization algorithm. *Phys. Rev. Lett.* **118**, 220402 (2017)
202. P. Silvi, T. Calarco, G. Morigi, S. Montangero, Ab initio characterization of the quantum linear-zigzag transition using density matrix renormalization group calculations. *Phys. Rev. B* **89**, 094103 (2014)
203. G.K.-L. Chan, S. Sharma, The density matrix renormalization group in quantum chemistry. *Annu. Rev. Phys. Chem.* **62**, 465–481 (2011)

204. N. Nakatani, G.K.-L. Chan, Efficient tree tensor network states (TTNS) for quantum chemistry: Generalizations of the density matrix renormalization group algorithm. *J. Chem. Phys.* **138**, 134113 (2013)
205. S. Wouters, D. Van Neck, The density matrix renormalization group for ab initio quantum chemistry. *Eur. Phys. J. D* **68**, 272 (2014)
206. M. Motta, D.M. Ceperley, G.K.-L. Chan, J.A. Gomez, E. Gull, S. Guo, C.A. Jiménez-Hoyos, T.N. Lan, J. Li, F. Ma, A.J. Millis, N.V. Prokof'ev, U. Ray, G.E. Scuseria, S. Sorella, E.M. Stoudenmire, Q. Sun, I.S. Tupitsyn, S.R. White, D. Zgid, S. Zhang, Towards the solution of the many-electron problem in real materials: Equation of state of the hydrogen chain with state-of-the-art many-body methods. *Phys. Rev. X* **7**, 031059 (2017)
207. B. Swingle, Entanglement renormalization and holography. *Phys. Rev. D* **86**, 065007 (2012)
208. J. Molina-Vilaplana, P. Sodano, Holographic view on quantum correlations and mutual information between disjoint blocks of a quantum critical system. *J. High Energy Phys.* **2011**, 11 (2011)
209. C. Beny, Causal structure of the entanglement renormalization ansatz. *New J. Phys.* **15**, 023020 (2013)
210. F. Pastawski, B. Yoshida, D. Harlow, J. Preskill, Holographic quantum error-correcting codes: Toy models for the bulk/boundary correspondence. *J. High Energy Phys.* **2015**, 149 (2015)
211. S. Singh, Tensor network state correspondence and holography. *Phys. Rev. D* **97**, 026012 (2018)
212. S. Singh, N.A. McMahon, G.K. Brennen, Holographic spin networks from tensor network states. *Phys. Rev. D* **97**, 026013 (2018)
213. E. Stoudenmire, D.J. Schwab, Supervised learning with tensor networks, in *Advances in Neural Information Processing Systems* 29, ed. by D.D. Lee, M. Sugiyama, U.V. Luxburg, I. Guyon, R. Garnett (Curran Associates, Inc., 2016), pp. 4799–4807
214. A. Cichocki, N. Lee, I. Oseledets, A.-H. Phan, Q. Zhao, M. Sugiyama, D.P. Mandic, Tensor networks for dimensionality reduction and large-scale optimization: Part 2 applications and future perspectives. *Found. Trends Mach. Learn.* **9**, 249–429 (2017)
215. E.M. Stoudenmire, S.R. White, *Itensor*, <http://itensor.org>
216. S. Al-Assam, S.R. Clark, D. Jaksch, The tensor network theory library. *J. Stat. Mech. Theory Exp.* **2017**, 093102 (2017)
217. D. Jaschke, M.L. Wall, L.D. Carr, Open source Matrix Product States: Opening ways to simulate entangled many-body quantum systems in one dimension. *Comput. Phys. Commun.* **225**, 59–91 (2018)
218. M. Dolfi, B. Bauer, S. Keller, A. Kosenkov, T. Ewart, A. Kantian, T. Giamarchi, M. Troyer, Matrix product state applications for the alps project. *Comput. Phys. Commun.* **185**, 3430–3440 (2014)
219. S.F. Keller, M. Reiher, Determining factors for the accuracy of DMRG in chemistry. *Chim. Int. J. Chem.* **68**, 200–203 (2014)
220. S. Keller, M. Dolfi, M. Troyer, M. Reiher, An efficient matrix product operator representation of the quantum chemical hamiltonian. *J. Chem. Phys.* **143**, 244118 (2015)
221. E. Noether, Invariant variation problems. *Transp. Theory Stat. Phys.* **1**, 186–207 (1971)
222. T. Lancaster, S. Blundell, *Quantum Field Theory for the Gifted Amateur* (OUP, Oxford, 2014)
223. M. Tinkham, *Group Theory and Quantum Mechanics*, *Dover Books on Chemistry* (Dover Publications, New York, NY, 2012)
224. S. Singh, H.-Q. Zhou, G. Vidal, Simulation of one-dimensional quantum systems with a global SU(2) symmetry. *New J. Phys.* **12**, 033029 (2010)
225. A. Weichselbaum, Non-Abelian symmetries in tensor networks: A quantum symmetry space approach. *Ann. Phys.* **327**, 2972–3047 (2012)
226. S. Singh, G. Vidal, Tensor network states and algorithms in the presence of a global SU(2) symmetry. *Phys. Rev. B* **86**, 195114 (2012)
227. P. Corboz, A.M. Läuchli, K. Penc, M. Troyer, F. Mila, Simultaneous dimerization and SU(4) symmetry breaking of 4-color fermions on the square lattice. *Phys. Rev. Lett.* **107**, 215301 (2011)

228. S. Singh, G. Vidal, Tensor network states and algorithms in the presence of a global SU(2) symmetry. *Phys. Rev. B* **86**, 195114 (2012)
229. A. Weichselbaum, Non-Abelian symmetries in tensor networks: A quantum symmetry space approach. *Ann. Phys. (NY)* **327**, 2972–3047 (2012)
230. S. Chandrasekharan, Confinement, chiral symmetry breaking and continuum limits in quantum link models. *Nucl. Phys. B - Proc. Suppl.* **73**, 739–741 (1999)
231. U.-J. Wiese, Ultracold quantum gases and lattice systems: Quantum simulation of lattice gauge theories. *Ann. Phys.* **525**, 777–796 (2013)
232. K.G. Wilson, Confinement of quarks. *Phys. Rev. D* **10**, 2445–2459 (1974)
233. M. Creutz, L. Jacobs, C. Rebbi, Monte Carlo computations in lattice gauge theories. *Phys. Rep.* **95**, 201–282 (1983)
234. O. Philipsen, The QCD equation of state from the lattice. *Prog. Part. Nucl. Phys.* **70**, 55–107 (2013)
235. K.A. Ross, L. Savary, B.D. Gaulin, L. Balents, Quantum excitations in quantum spin ice. *Phys. Rev. X* **1**, 021002 (2011)
236. S. Lee, S. Onoda, L. Balents, Generic quantum spin ice. *Phys. Rev. B* **86**, 104412 (2012)
237. N. Goldman, J.C. Budich, P. Zoller, Topological quantum matter with ultracold gases in optical lattices. *Nat. Phys.* **12**, 639 (2016)
238. P. Silvi, E. Rico, M. Dalmonte, F. Tschirsich, S. Montangero, Finite-density phase diagram of a non-Abelian lattice gauge theory with tensor networks. *Quantum* **1**, 9 (2017)
239. T. Pichler, M. Dalmonte, E. Rico, P. Zoller, S. Montangero, Real-time dynamics in U(1) lattice gauge theories with tensor networks. *Phys. Rev. X* **6**, 011023 (2016)
240. B. Buyens, K.V. Acoleyen, J. Haegeman, F. Verstraete, Results Schwinger model results for the massless Schwinger model. *Phys. Rev. Lett.* **113**, 091601 (2014)
241. S. Sachdev, *Quantum Phase Transitions* (Cambridge University Press, Cambridge, MA, 2011)
242. E. Fradkin, *Field Theories of Condensed Matter Physics, Field Theories of Condensed Matter Physics* (Cambridge University Press, Cambridge, MA, 2013)
243. L. Landau, Stochastic dynamics of quantum-mechanical systems. *Zh. Eksp. Teor. Fiz.* **7**, 19 (1937)
244. S. Blundell, *Magnetism in Condensed Matter, Oxford Master Series in Condensed Matter Physics* (OUP, Oxford, 2001)
245. S.L. Sondhi, S.M. Girvin, J.P. Carini, D. Shahar, Continuous quantum phase transitions. *Rev. Mod. Phys.* **69**, 315–333 (1997)
246. M. Suzuki, Relationship between d -dimensional quantum spin systems and $(d + 1)$ -dimensional Ising systems: Equivalence, critical exponents and systematic approximants of the partition function and spin correlations. *Prog. Theor. Phys.* **56**, 1454–1469 (1976)
247. A. Osterloh, L. Amico, G. Falci, R. Fazio, Scaling of entanglement close to a quantum phase transition. *Nature* **416**, 608–610 (2002)
248. M. Fisher, M. Barber, Scaling theory for finite-size effects in the critical region. *Phys. Rev. Lett.* **28**, 1516–1519 (1972)
249. T.J. Osborne, M.A. Nielsen, Entanglement in a simple quantum phase transition. *Phys. Rev. A* **66**, 032110 (2002)
250. P. Calabrese, J. Cardy, Entanglement entropy and quantum field theory. *J. Stat. Mech. Theory Exp.* **2004**, P06002 (2004)
251. G. Refael, J.E. Moore, Entanglement entropy of random quantum critical points in one dimension. *Phys. Rev. Lett.* **93**, 260602 (2004)
252. D. Binosi, G. De Chiara, S. Montangero, A. Recati, Increasing entanglement through engineered disorder in the random Ising chain. *Phys. Rev. B* **76**, 140405 (2007)
253. J.C. Getelina, F.C. Alcaraz, J.A. Hoyos, Entanglement properties of correlated random spin chains and similarities with conformally invariant systems. *Phys. Rev. B* **93**, 1–13 (2016)
254. J.A. Hoyos, N. Laflorencie, A.P. Vieira, T. Vojta, Protecting clean critical points by local disorder correlations. *EPL (Europhysics Lett.)* **93**, 30004 (2011)
255. N. Laflorencie, Quantum entanglement in condensed matter systems. *Phys. Rep.* **646**, 1–59 (2015)

256. A. Kitaev, J. Preskill, Topological entanglement entropy. *Phys. Rev. Lett.* **96**, 110404 (2006)
257. M. Levin, X.-G. Wen, Detecting topological order in a ground state wave function. *Phys. Rev. Lett.* **96**, 110405 (2006)
258. H.-C. Jiang, Z. Wang, L. Balents, Identifying topological order by entanglement entropy. *Nat. Phys.* **8**, 902–905 (2012)
259. H.-C. Jiang, R.R.P. Singh, L. Balents, Accuracy of topological entanglement entropy on finite cylinders. *Phys. Rev. Lett.* **111**, 107205 (2013)
260. S. Depenbrock, I.P. McCulloch, U. Schollwoeck, Nature of the spin-liquid ground state of the $S = 1/2$ Heisenberg model on the kagome lattice. *Phys. Rev. Lett.* **109**, 067201 (2012)
261. R. Orus, T.C. Wei, O. Buerschaper, A. Garcia-Saez, Topological transitions from multipartite entanglement with tensor networks: A procedure for sharper and faster characterization. *Phys. Rev. Lett.* **113**, 257202 (2014)
262. M. Gerster, M. Rizzi, P. Silvi, M. Dalmonte, S. Montangero, Fractional quantum Hall effect in the interacting Hofstadter model via tensor networks. *Phys. Rev. B* **96**, 195123 (2017)
263. A. Lucas, Ising formulations of many NP problems. *Front. Phys.* **2**, 5 (2014)
264. D. Aharonov, W. van Dam, J. Kempe, Z. Landau, S. Lloyd, O. Regev, Adiabatic quantum computation is equivalent to standard quantum computation. *SIAM Rev.* **50**, 755–787 (2008)
265. D. Kienzler, H.Y. Lo, B. Keitch, L. de Clercq, F. Leupold, F. Lindenfelser, M. Marinelli, V. Negnevitsky, J.P. Home, Quantum harmonic oscillator state synthesis by reservoir engineering. *Science* **347**, 53–56 (2015)
266. H. Krauter, C.A. Muschik, K. Jensen, W. Wasilewski, J.M. Petersen, J.I. Cirac, E.S. Polzik, Entanglement generated by dissipation and steady state entanglement of two macroscopic objects. *Phys. Rev. Lett.* **107**, 080503 (2011)
267. J.F. Poyatos, J.I. Cirac, P. Zoller, Quantum reservoir engineering with laser cooled trapped ions. *Phys. Rev. Lett.* **77**, 4728–4731 (1996)
268. D. Tannor, *Introduction to Quantum Mechanics: A Time-Dependent Perspective* (University Science Books, 2016)
269. J. Roland, *Adiabatic Quantum Computation* (Ph.D. Theiss, Universit'e libre de Brussels, 2004)
270. J. Roland, N.J. Cerf, Quantum search by local adiabatic evolution. *Phys. Rev. A* **65**, 042308 (2002)
271. T. Caneva, T. Calarco, R. Fazio, G.E. Santoro, S. Montangero, Speeding up critical system dynamics through optimized evolution. *Phys. Rev. A* **84**, 012312 (2011)
272. C. Zener, Non-adiabatic crossing of energy levels. *Proc. R. Soc. Lond. A: Math. Phys. Eng. Sci.* **137**, 696–702 (1932)
273. L. Landau, Zur theorie der energieubertragung II. *Physikalische Zeitschrift der Sowjetunion* **2**, 46–51 (1932)
274. E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, D. Preda, A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem. *Science* **292**, 472–476 (2001)
275. G.E. Santoro, E. Tosatti, Optimization using quantum mechanics: Quantum annealing through adiabatic evolution. *J. Phys. A: Math. Gen.* **39**, R393–R431 (2006)
276. G.E. Santoro, Theory of quantum annealing of an ising spin glass. *Science* **295**, 2427–2430 (2002)
277. W.H. Zurek, U. Dorner, P. Zoller, Dynamics of a quantum phase transition. *Phys. Rev. Lett.* **95**, 105701 (2005)
278. W.H. Zurek, Cosmological experiments in superfluid helium? *Nature* **317**, 505–508 (1985)
279. T.W.B. Kibble, Topology of cosmic domains and strings. *J. Phys. A: Math. Gen.* **9**, 1387–1398 (1976)
280. B. Damski, The simplest quantum model supporting the Kibble–Zurek mechanism of topological defect production: Landau-Zener transitions from a new perspective. *Phys. Rev. Lett.* **95**, 035701 (2005)
281. E. Canovi, E. Ercolessi, P. Naldesi, L. Taddia, D. Vodola, Dynamics of entanglement entropy and entanglement spectrum crossing a quantum phase transition. *Phys. Rev. B* **89**, 104303 (2014)

282. B. Gardas, J. Dziarmaga, W.H. Zurek, Dynamics of the quantum phase transition in the one-dimensional Bose-Hubbard model: Excitations and correlations induced by a quench. *Phys. Rev. B* **95**, 104306 (2017)
283. J. Dziarmaga, J. Meisner, W.H. Zurek, Winding up of the wave-function phase by an insulator-to-superfluid transition in a ring of coupled Bose-Einstein condensates. *Phys. Rev. Lett.* **101**, 115701 (2008)
284. J.-M. Cui, Y.-F. Huang, Z. Wang, D.-Y. Cao, J. Wang, W.-M. Lv, L. Luo, A. del Campo, Y.-J. Han, C.-F. Li, G.-C. Guo, Experimental trapped-ion quantum simulation of the Kibble-Zurek dynamics in momentum space. *Sci. Rep.* **6**, 33381 (2016)
285. A. Dutta, A. Rahmani, A. del Campo, Anti-Kibble-Zurek behavior in crossing the quantum critical point of a thermally isolated system driven by a noisy control field. *Phys. Rev. Lett.* **117**, 080402 (2016)
286. P. Silvi, G. Morigi, T. Calarco, S. Montangero, Crossover from classical to quantum Kibble-Zurek Scaling. *Phys. Rev. Lett.* **116**, 225701 (2016)
287. P.M. Chesler, A.M. García-García, H. Liu, Defect formation beyond Kibble-Zurek mechanism and holography. *Phys. Rev. X* **5**, 021015 (2015)
288. S. Ulm, J. Roßnagel, G. Jacob, C. Degünther, S.T. Dawkins, U.G. Poschinger, R. Nigmatullin, A. Retzker, M.B. Plenio, F. Schmidt-Kaler, K. Singer, Observation of the Kibble-Zurek scaling law for defect formation in ion crystals. *Nat. Commun.* **4**, 2290 (2013)
289. K. Pyka, J. Keller, H.L. Partner, R. Nigmatullin, T. Burgermeister, D.M. Meier, K. Kuhlmann, A. Retzker, M.B. Plenio, W.H. Zurek, A. del Campo, T.E. Mehlstäubler, Topological defect formation and spontaneous symmetry breaking in ion Coulomb crystals. *Nat. Commun.* **4**, 2291 (2013)
290. J. Dziarmaga, W.H. Zurek, Quench in the 1D Bose-Hubbard model: Topological defects and excitations from the Kosterlitz-Thouless phase transition dynamics. *Sci. Rep.* **4**, 5950 (2015)
291. J. Dziarmaga, Density of kinks just after a quench in an underdamped system. *Phys. Rev. Lett.* **81**, 1551–1553 (1998)
292. N. Navon, A.L. Gaunt, R.P. Smith, Z. Hadzibabic, Critical dynamics of spontaneous symmetry breaking in a homogeneous Bose gas. *Science* **347**, 167–170 (2015)
293. S. Braun, M. Friesdorf, S.S. Hodgman, M. Schreiber, J.P. Ronzheimer, A. Riera, M. del Rey, I. Bloch, J. Eisert, U. Schneider, Emergence of coherence and the dynamics of quantum phase transitions. *Proc. Natl. Acad. Sci.* **112**, 3641–3646 (2015)
294. F. Pellegrini, S. Montangero, G. Santoro, R. Fazio, Adiabatic quenches through an extended quantum critical region. *Phys. Rev. B* **77**, 140404 (2008)
295. J. Dziarmaga, Dynamics of a quantum phase transition and relaxation to a steady state. *Adv. Phys.* **59**, 1063–1189 (2010)
296. J. Dziarmaga, M. Tylutki, W.H. Zurek, Quench from Mott insulator to superfluid. *Phys. Rev. B* **86**, 144521 (2012)
297. P. Silvi, G. De Chiara, T. Calarco, G. Morigi, S. Montangero, Full characterization of the quantum linear-zigzag transition in atomic chains. *Ann. Phys.* **525**, 827–832 (2013)
298. C. Schneider, D. Porras, T. Schaetz, Experimental quantum simulations of many-body physics with trapped ions. *Reports on Progress in Physics* **75**, 024401 (2012)
299. M. Mielenz, J. Brox, S. Kahra, G. Leschhorn, M. Albert, T. Schaetz, H. Landa, B. Reznik, Trapping of topological-structural defects in coulomb crystals. *Phys. Rev. Lett.* **110**, 133004 (2013)
300. L. Corman, L. Chomaz, T. Bienaimé, R. Desbuquois, C. Weitenberg, S. Nascimbéne, J. Dalibard, J. Beugnon, Quench-induced supercurrents in an annular bose gas. *Phys. Rev. Lett.* **113**, 135302 (2014)
301. E. Shimshoni, G. Morigi, S. Fishman, Quantum zigzag transition in ion chains. *Phys. Rev. Lett.* **106**, 010401 (2011)
302. E. Shimshoni, G. Morigi, S. Fishman, Quantum structural phase transition in chains of interacting atoms. *Phys. Rev. A* **83**, 032308 (2011)
303. J. Als-Nielsen, R.J. Birgeneau, Mean field theory, the Ginzburg criterion, and marginal dimensionality of phase transitions. *Am. J. Phys.* **45**, 554–560 (1977)

304. D.J. Amit, The ginzburg criterion-rationalized. *J. Phys. C: Solid State Phys.* **7**, 3369 (1974)
305. S. Campbell, S. Deffner, Trade-off between speed and cost in shortcuts to adiabaticity. *Phys. Rev. Lett.* **118**, 100601 (2017)
306. J.G. Muga, X. Chen, E. Torrontegui, S. Ibáñez, I. Lizuain, A. Ruschhaupt, Shortcuts to quantum adiabatic processes. *J. Phys. Conf. Ser.* **306**, 012022 (2011)
307. A. del Campo, Shortcuts to adiabaticity by counterdiabatic driving. *Phys. Rev. Lett.* **111**, 100502 (2013)
308. E. Torrontegui, S. Martínez-Garaot, A. Ruschhaupt, J.G. Muga, Shortcuts to adiabaticity: Fast-forward approach. *Phys. Rev. A* **86**, 013601 (2012)
309. E. Torrontegui, S. Ibáñez, S. Martínez-Garaot, M. Modugno, A. del Campo, D. Guéry-Odelin, A. Ruschhaupt, X. Chen, J.G. Muga, *Advances in Atomic, Molecular, and Optical Physics*, Vol. 62, *Advances In Atomic, Molecular, and Optical Physics* (Elsevier, Amsterdam, 2013), pp. 117–169
310. C. Jarzynski, Generating shortcuts to adiabaticity in quantum and classical dynamics. *Phys. Rev. A* **88**, 040101 (2013)
311. E. Torrontegui, I. Lizuain, S. González-Resines, A. Tobalina, A. Ruschhaupt, R. Kosloff, J.G. Muga, Energy consumption for shortcuts to adiabaticity. *Phys. Rev. A* **96**, 022133 (2017)
312. A. del Campo, M.G. Boshier, Shortcuts to adiabaticity in a time-dependent box. *Sci. Rep.* **2**, 648 (2012)
313. W. Rohringer, D. Fischer, F. Steiner, I.E. Mazets, J. Schmiedmayer, M. Trupke, Non-equilibrium scale invariance and shortcuts to adiabaticity in a onedimensional Bose gas. *Sci. Rep.* **5**, 9820 (2015)
314. H. Saberi, T. Opatrny, K. Mølmer, A. Del Campo, Adiabatic tracking of quantum many-body dynamics. *Phys. Rev. A* **90**, 060301 (2014)
315. D. D'Alessandro, *Introduction to Quantum Control and Dynamics*, *Chapman & Hall/CRC Applied Mathematics & Nonlinear Science* (CRC Press, Boca Raton, FL, 2007)
316. C. Altafini, F. Ticozzi, Modeling and control of quantum systems: An introduction. *IEEE Trans. Automat. Contr.* **57**, 1898–1917 (2012)
317. C. Brif, R. Chakrabarti, H. Rabitz, Control of quantum phenomena: Past, present and future. *New J. Phys.* **12**, 075008 (2010)
318. S. van Frank, M. Bonneau, J. Schmiedmayer, S. Hild, C. Gross, M. Cheneau, I. Bloch, T. Pichler, A. Negretti, T. Calarco, S. Montangero, Optimal control of complex atomic quantum systems. *Sci. Rep.* **6**, 34187 (2016)
319. M.G. Bason, M. Viteau, N. Malossi, P. Huillery, E. Arimondo, D. Ciampini, R. Fazio, V. Giovannetti, R. Mannella, O. Morsch, High-fidelity quantum driving. *Nat. Phys.* **8**, 147–152 (2011)
320. Z. Sun, J. Liu, J. Ma, X. Wang, Quantum speed limits in open systems: Non-Markovian dynamics without rotating-wave approximation. *Sci. Rep.* **5**, 8444 (2015)
321. T. Caneva, T. Calarco, S. Montangero, Chopped random-basis quantum optimization. *Phys. Rev. A* **84**, 022326 (2011)
322. N. Wu, A. Nanduri, H. Rabitz, Optimal suppression of defect generation during a passage across a quantum critical point. *Phys. Rev. B - Condens. Matter Mater. Phys.* **91**, 041115 (2015)
323. C.P. Koch, Controlling open quantum systems: Tools, achievements, and limitations. *J. Phys. Condens. Matter* **28**, 213001 (2016)
324. J. Cui, R. van Bijnen, T. Pohl, S. Montangero, T. Calarco, Optimal control of Rydberg lattice gases. *Quantum Sci. Technol.* **2**, 035006 (2017)
325. S. Deffner, E. Lutz, Energytime uncertainty relation for driven quantum systems. *J. Phys. A Math. Theor.* **46**, 335302 (2013)
326. V. Giovannetti, S. Lloyd, L. Maccone, Quantum limits to dynamical evolution. *Phys. Rev. A* **67**, 052109 (2003)
327. P.M. Poggi, D.A. Wisniacki, Optimal control of many-body quantum dynamics: Chaos and complexity. *Phys. Rev. A* **94**, 033406 (2016)

328. M. Hirose, P. Cappellaro, Time-optimal control with finite bandwidth. *Quantum Inf. Process* **17**, 88 (2018)
329. R. Uzdin, R. Kosloff, Speed limits in Liouville space for open quantum systems. *EPL (Europhysics Lett.)* **115**, 40003 (2016)
330. S. Lloyd, S. Montangero, Information theoretical analysis of quantum optimal control. *Phys. Rev. Lett.* **113**, 010502 (2014)
331. M.R. Frey, Quantum speed limits primer, perspectives, and potential future directions. *Quantum Inf. Process.* **15**, 3919–3950 (2016)
332. D.P. Pires, M. Cianciaruso, L.C. Céleri, G. Adesso, D.O. Soares-Pinto, Generalized geometric quantum speed limits. *Phys. Rev. X* **6**, 021031 (2016)
333. M.M. Taddei, B.M. Escher, L. Davidovich, R.L. de Matos Filho, Quantum speed limit for physical processes. *Phys. Rev. Lett.* **110**, 050402 (2013)
334. S. Deffner, E. Lutz, Quantum speed limit for non-markovian dynamics. *Phys. Rev. Lett.* **111**, 010402 (2013)
335. J.J.W.H. Sørensen, M.K. Pedersen, M. Munch, P. Haikka, J.H. Jensen, T. Planke, M.G. Andreassen, M. Gajdacz, K. Mølmer, A. Lieberoth, J.F. Sherson, Exploring the quantum speed limit with computer games. *Nature* **532**, 210–213 (2016)
336. I. Brouzos, A.I. Streltsov, A. Negretti, R.S. Said, T. Caneva, S. Montangero, T. Calarco, Quantum speed limit and optimal control of many-boson dynamics. *Phys. Rev. A* **92**, 062110 (2015)
337. S. van Frank, A. Negretti, T. Berrada, R. Bücker, S. Montangero, J.-F. Schaff, T. Schumm, T. Calarco, J. Schmiedmayer, Interferometry with non-classical motional states of a Bose-Einstein condensate. *Nat. Commun.* **5**, 4009 (2014)
338. A. Rahmani, C. Chamon, Optimal control for unitary preparation of many-body states: Application to luttinger liquids. *Phys. Rev. Lett.* **107**, 016402 (2011)
339. P. Doria, T. Calarco, S. Montangero, Optimal control technique for manybody quantum dynamics. *Phys. Rev. Lett.* **106**, 190501 (2011)
340. T. Caneva, A. Silva, R. Fazio, S. Lloyd, T. Calarco, S. Montangero, Complexity of controlling quantum many-body dynamics. *Phys. Rev. A* **89**, 042322 (2014)
341. S. Lloyd, S. Montangero, Information theoretical analysis of quantum optimal control. *Phys. Rev. Lett.* **113**, 10502 (2014)
342. T. Caneva, T. Calarco, S. Montangero, Entanglement-storage units. *New J. Phys.* **14**, 093041 (2012)
343. M.H. Goerz, F. Motzoi, K.B. Whaley, C.P. Koch, Charting the circuit QED design landscape using optimal control theory. *npj Quantum Inf.* **3**, 37 (2017)
344. A. del Campo, K. Sengupta, Controlling quantum critical dynamics of isolated systems. *Eur. Phys. J. Spec. Top.* **224**, 189–203 (2015)
345. S. Rosi, A. Bernard, N. Fabbri, L. Fallani, C. Fort, M. Inguscio, T. Calarco, S. Montangero, Fast closed-loop optimal control of ultracold atoms in an optical lattice. *Phys. Rev. A* **88**, 021601 (2013)
346. A. Rahmani, T. Kitagawa, E. Demler, C. Chamon, Cooling through optimal control of quantum evolution. *Phys. Rev. A* **87**, 043607 (2013)
347. M.C. Tichy, M.K. Pedersen, K. Mølmer, J.F. Sherson, Nonadiabatic quantum state control of many bosons in few wells. *Phys. Rev. A* **87**, 063422 (2013)
348. A. del Campo, M.M. Rams, W.H. Zurek, Assisted finite-rate adiabatic passage across a quantum critical point: Exact solution for the quantum ising model. *Phys. Rev. Lett.* **109**, 115703 (2012)
349. A. Castro, J. Werschnik, E.K.U. Gross, Controlling the dynamics of many-electron systems from first principles: A combination of optimal control and time-dependent density-functional theory. *Phys. Rev. Lett.* **109**, 153603 (2012)
350. T. Caneva, M. Murphy, T. Calarco, R. Fazio, S. Montangero, V. Giovannetti, G.E. Santoro, Optimal control at the quantum speed limit. *Phys. Rev. Lett.* **103**, 240501 (2009)
351. M.M. Müller, A. Kölle, R. Löw, T. Pfau, T. Calarco, S. Montangero, Roomtemperature Rydberg single-photon source. *Phys. Rev. A* **87**, 053412 (2013)

352. N. Rach, M.M. Müller, T. Calarco, S. Montangero, Dressing the chopped-random-basis optimization: A bandwidth-limited access to the trap-free landscape. *Phys. Rev. A* **92**, 062343 (2015)
353. K.W. Moore, H. Rabitz, Exploring constrained quantum control landscapes. *J. Chem. Phys.* **137**, 134113 (2012)
354. C. Kittel, H. Shore, Development of a phase transition for a rigorously solvable many-body system. *Phys. Rev.* **138**, A1165–A1169 (1965)
355. S. Deffner, S. Campbell, Quantum speed limits: From Heisenberg’s uncertainty principle to optimal quantum control. *J. Phys. A Math. Theor.* **50**, 453001 (2017)
356. N. Margolus, L.B. Levitin, The maximum speed of dynamical evolution, *Physica D: Nonlinear Phenomena* **120**, Proceedings of the Fourth Workshop on Physics and Consumption, pp. 188–195 (1998)
357. G.C. Hegerfeldt, Driving at the quantum speed limit: optimal control of a two-level system. *Phys. Rev. Lett.* **111**, 260501 (2013)
358. M. Murphy, S. Montangero, V. Giovannetti, T. Calarco, Communication at the quantum speed limit along a spin chain. *Phys. Rev. A* **82**, 022318 (2010)
359. S. Ashhab, P.C. de Groot, F. Nori, Speed limits for quantum gates in multiqubit systems. *Phys. Rev. A* **85**, 052327 (2012)
360. X. Wang, M. Allegra, K. Jacobs, S. Lloyd, C. Lupo, M. Mohseni, Quantum brachistochrone curves as geodesics: Obtaining accurate control protocols for time-optimal quantum gates. *Phys. Rev. Lett.* **114**, 170501 (2015)
361. L. Levitin, T. Toffoli, Fundamental limit on the rate of quantum dynamics: The unified bound is tight. *Phys. Rev. Lett.* **103**, 160502 (2009)
362. T. Cohen, D. McGady, Schwinger mechanism revisited. *Phys. Rev. D* **78**, 036008 (2008)
363. F. Hebenstreit, J. Berges, D. Gelfand, Real-time dynamics of string breaking. *Phys. Rev. Lett.* **111**, 201601 (2013)
364. M. Heyl, A. Polkovnikov, S. Kehrein, Dynamical quantum phase transitions in the transverse-field ising model. *Phys. Rev. Lett.* **110**, 135704 (2013)
365. V. Gurarie, Viewpoint: Quantum phase transitions go dynamical. *Physics* **10**, 95 (2017)
366. A.D. Greentree, C. Tahan, J.H. Cole, L.C.L. Hollenberg, Quantum phase transitions of light. *Nat. Phys.* **2**, 856–861 (2006)
367. M.J. Hartmann, F.G.S.L. Brandão, M.B. Plenio, Strongly interacting polaritons in coupled arrays of cavities. *Nat. Phys.* **2**, 849–855 (2006)
368. D. Rossini, R. Fazio, Mott-insulating and glassy phases of polaritons in 1D arrays of coupled cavities. *Phys. Rev. Lett.* **99**, 186401 (2007)
369. P. Jurcevic, H. Shen, P. Hauke, C. Maier, T. Brydges, C. Hempel, B.P. Lanyon, M. Heyl, R. Blatt, C.F. Roos, Direct observation of dynamical quantum phase transitions in an interacting many-body system. *Phys. Rev. Lett.* **119**, 080501 (2017)
370. C. Gogolin, Equilibration and thermalization in quantum systems, Ph.D. thesis (Freien Universität Berlin, 2014)
371. A. Polkovnikov, K. Sengupta, A. Silva, M. Vengalattore, Colloquium: Nonequilibrium dynamics of closed interacting quantum systems. *Rev. Mod. Phys.* **83**, 863–883 (2011)
372. G. Benenti, G. Casati, S. Montangero, D. Shepelyansky, Efficient quantum computing of complex dynamics. *Phys. Rev. Lett.* **87**, 227901 (2001)
373. E. Canovi, D. Rossini, R. Fazio, G.E. Santoro, A. Silva, Quantum quenches, thermalization, and many-body localization. *Phys. Rev. B* **83**, 094431 (2011)
374. L. Santos, A. Polkovnikov, M. Rigol, Entropy of isolated quantum systems after a quench. *Phys. Rev. Lett.* **107**, 040601 (2011)
375. R. Dorner, J. Goold, C. Cormick, M. Paternostro, V. Vedral, Emergent thermodynamics in a quenched quantum many-body system. *Phys. Rev. Lett.* **109**, 160601 (2012)
376. J.A. Kjäll, J.H. Bardarson, F. Pollmann, Many-body localization in a disordered quantum ising chain. *Phys. Rev. Lett.* **113**, 107204 (2014)
377. J. Eisert, M. Friesdorf, C. Gogolin, Quantum many-body systems out of equilibrium. *Nat. Phys.* **11**, 124–130 (2015)

378. A.M. Läuchli, C. Kollath, Spreading of correlations and entanglement after a quench in the one-dimensional BoseHubbard model. *J. Stat. Mech. Theory Exp.* **2008**, P05018 (2008)
379. S. Wolff, A. Sheikhan, C. Kollath, Dissipative time evolution of a chiral state after a quantum quench. *Phys. Rev. A* **94**, 043609 (2016)
380. A.W. Chin, Á. Rivas, S.F. Huelga, M.B. Plenio, Exact mapping between system-reservoir quantum models and semi-infinite discrete chains using orthogonal polynomials. *J. Math. Phys.* **51**, 092109 (2010)
381. J. Prior, A.W. Chin, S.F. Huelga, M.B. Plenio, Efficient simulation of strong system-environment interactions. *Phys. Rev. Lett.* **105**, 050404 (2010)
382. P. Lodahl, S. Mahmoodian, S. Stobbe, A. Rauschenbeutel, P. Schneeweiss, J. Volz, H. Pichler, P. Zoller, Chiral quantum optics. *Nature* **541**, 473–480 (2017)
383. H. Pichler, P. Zoller, Photonic circuits with time delays and quantum feedback. *Phys. Rev. Lett.* **116**, 093601 (2016)
384. T. Ramos, H. Pichler, A.J. Daley, P. Zoller, Quantum spin dimers from chiral dissipation in cold-atom chains. *Phys. Rev. Lett.* **113**, 237203 (2014)
385. P. Arrighi, J. Grattage, The quantum game of life. *Phys. World* **25**, 23–26 (2012)
386. D. Bleh, T. Calarco, S. Montangero, Quantum game of life. *EPL (Europhysics Lett.)* **97**, 20012 (2012)
387. U. Alvarez-Rodriguez, M. Sanz, L. Lamata, E. Solano, Artificial life in quantum technologies. *Sci. Rep.* **6**, 20956 (2016)
388. M.A. Valdez, D. Jaschke, D.L. Vargas, L.D. Carr, Quantifying complexity in quantum phase transitions via mutual information complex networks. *Phys. Rev. Lett.* **119**, 225301 (2017)
389. M.A. Nielsen, I.L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, MA, 2010)
390. IEEE standard for binary floating-point arithmetic, ANSI/IEEE Std 754-1985, 0 1- (1985)
391. B.J. Bloom, T.L. Nicholson, J.R. Williams, S.L. Campbell, M. Bishof, X. Zhang, W. Zhang, S.L. Bromley, J. Ye, An optical lattice clock with accuracy and stability at the 10^{-18} level. *Nature* **506**, 71 (2014)
392. P.J. Denning, The locality principle. *Commun. ACM* **48**, 19 (2005)
393. *The top 500 list*, www.top500.org
394. R. van der Pas, E. Stotzer, C. Terboven, *Using OpenMP-The Next Step: Affinity, Accelerators, Tasking, and SIMD, Scientific and Engineering Computation* (MIT Press, Cambridge, MA, 2017)
395. L.S. Blackford, A. Petitet, R. Pozo, K. Remington, R.C. Whaley, J. Demmel, J. Dongarra, I. Duff, S. Hammarling, G. Henry, M. Heroux, L. Kaufman, A. Lumsdaine, An updated set of basic linear algebra subprograms (BLAS). *ACM Trans. Math. Softw.* **28**, 135–151 (2002)
396. W. Gropp, E. Lusk, A. Skjellum, *Using MPI: Portable Parallel Programming with the Message-Passing Interface, Scientific and Engineering Computation* (MIT Press, Cambridge, MA, 2014)
397. G. Amdahl, *Proceedings AFIPS* **30**, 483–485 (1967)
398. T. Sterling, *Beowulf Cluster Computing with Linux*, Scientific and Computational Engineering Series (Books24x7.com, 2002)
399. M. Scarpino, *Opencl in Action: How to Accelerate Graphics and Computation* (Manning, 2012)
400. A. Sheppard, *Programming GPU's* (O'Reilly Media, Incorporated, 2012)
401. A. Robbins, *Gdb pocket reference, Pocket Reference* (O'Reilly) (O'Reilly Media, 2005)
402. R. Stallman, C. Support, *Debugging with Gdb: the Gnu Source-Level Debugger* (Free Software Foundation, 1996)
403. F. Haake, *Quantum Signatures of Chaos, Springer Series in Synergetics* (Springer, Berlin, 2010)
404. E. Burke, B. Coyner, *Java Extreme Programming Cookbook, Cookbook Series* (O'Reilly, 2003)
405. S. Aaronson, *Quantum Computing Since Democritus* (Cambridge University Press, Cambridge, MA, 2013)

406. D.A. Spielman, S.-H. Teng, Smoothed analysis: An attempt to explain the behavior of algorithms in practice. *Commun. ACM* **52**, 76–84 (2009)
407. M. Bläser, B. Manthey, Smoothed complexity theory. *Math. Found. Comput. Sci., LNCS* 7464, **198** (2012)
408. D.A. Spielman, S.-H. Teng, Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *J. ACM* **51**, 385–463 (2001)

Index

A

Adiabatic quantum computation, 110
Adiabatic theorem, 110
Algorithmic complexity, 10
Anomalous dimension, 103
Auxiliary dimension, 60
Auxiliary indexes, 54, 60

C

Central charge, 106
Cluster mean field, 40
Conformal field theory, 106
Correlation length exponent, 103
Crank-Nicolson method, 30

D

DCRAB, 119
Degeneracy tensor, 84
Dimensionality (group), 80
Dynamical-scaling exponent, 104

E

Equivalent representations, 81
Euler method, 29

F

Fast Fourier Transform , 29
Free energy, 98, 100

G

Gauge symmetries, 86
Gaussian elimination, 10

Gaussian integration, 22
Group theory, 80

H

Homomorphic (group), 80

I

Index fusion, 52
Index splitting, 52
Inverse power method, 13
Irreducible representation, 81
Ising model, 38
Isometric condition, 62
Isomorphic (group), 80

K

Kibble-Zurek mechanism, 114

L

Lanczos diagonalization, 15
Landau-Zener crossing, 112
Locally purified tensor network, 69
Local observables, 74
LU decomposition, 11, 14

M

Matrix product operator, 56, 64
Matrix product state, 55, 60
Mean field, 38, 45
Multiscale entanglement renormalization ansatz, 69

N

Non-local observables, [74](#)

O

Order (group), [80](#)

Order parameter, [102](#)

P

Partition function, [100](#)

Physical indexes, [54](#), [60](#)

Polar decomposition, [54](#), [62](#)

Projected entangled pair states, [69](#)

Q

Quantum annealing, [110](#)

Quantum link formulation, [86](#), [87](#), [89](#)

Quantum speed limit, [121](#)

R

Reducible representation, [81](#)

Rényi entropies, [107](#)

Runge-Kutta method, [31](#)

S

Simpson's rule, [21](#)

Singular value decomposition, [52](#), [63](#), [66](#), [70](#),
[73](#)

Spontaneous magnetization, [102](#)

Strassen algorithm, [55](#)

Structural tensor, [84](#)

Structure factor, [103](#), [104](#)

Suzuki-Trotter decomposition, [69](#)

T

Tensor gauging, [54](#), [62](#), [69](#), [72–75](#)

Time-dependent DRMG, [70](#)

Time-dependent variational principle, [72](#)

Time evolving block decimation, [70](#)

Topological entropy, [108](#)

Transfer matrix, [31](#), [100](#)

Trapezoidal rule, [19](#)

U

Universality classes, [102](#)

V

Von Neumann entropy, [46](#), [75](#), [105](#), [106](#)