Geometry of Entanglement and Quantum Simulators

Eirik Ovrum

Thesis submitted for the degree of
Philosophiæ Doctor

Department of Physics
University of Oslo

May 2007

Abstract

This phd-thesis presents articles on the geometry of entanglement, and on quantum computer algorithms for simulating quantum systems. A thorough introduction to the geometry of entanglement is given, with two-dimensional cross sections through the space of hermitian matrices showing the boundary of the density matrices and of the positive partial transpose density matrices. We present an alternative proof of the sufficiency of Péres criterion for the case of two two-dimensional systems. We describe, and show numerical results, for an algorithm that finds the shortest distance from a given density matrix to the set of separable states. We also show cross sections through the hermitian matrices where we have used this algorithm to find the boundary of the separable states. We give a criteria for determining whether a state is an extreme point of the convex set of positive partial transpose density matrices, and show results from an algorithm finding such states randomly. This algorithm finds random bound entangled states and can be used to further understand the structure of the positive partial transpose states.

The second part is on algorithms for quantum computers to use for simulating quantum systems. We use the Jordan-Wigner transformation to create a compiler that takes any two-body fermionic Hamiltonian and outputs all qubit gates needed to simulate the time evolution of that Hamiltonian. Numerical simulations are included of a quantum computer using this compiler to find the eigenvalues of the Hubbard and the pairing models.

Acknowledgments

To come

List of papers

Paper I: Jon Magne Leinaas, Jan Myrheim and Eirik Ovrum,
Geometrical aspects of entanglement,
Phys. Rev. **A 74**, 012313 (2006)

Paper II: Geir Dahl, Jon Magne Leinaas, Jan Myrheim and Eirik Ovrum,
A tensor product matrix approximation problem in quantum physics,
Linear Algebra and its Applications **Volume 420**, Issues 2-3 , 15 January 2007, Pages 711-725

Paper III: Jon Magne Leinaas, Jan Myrheim and Eirik Ovrum,
Extreme points of the set of density matrices with positive partial transpose,
Submitted to PRL, arXiv:0704.3348v1 (2007)

Paper IV: Morten Hjorth-Jensen and Eirik Ovrum,
Quantum computation algorithm for many-body studies,
arXiv:0705.1928v1 (2007)

Contents

Part I

Introduction

Chapter 1

Introduction

Quantum mechanics is the theory that describes the smallest parts of nature. It gives amazingly accurate predictions and when it came, quantum mechanics revolutionized science. All old physics was called classical physics contrary to quantum physics. The quantum world behaves in ways that seem counter-intuitive to our eyes accustomed to the macroscopic. The two most striking features are the non-locality of quantum mechanics and the fact that quantum mechanics only tells us the probabilities of obtaining measurement values, not values with certainty as is possible in all classical theories.

Quantum mechanics is not a local theory, the state of two particles that are separated by a great distance, can be changed by measurements or operations on one of the particles, changing the possible outcome of measurements on the other particle immediately. At first glance this seems contradictory to our understanding of the world, especially in light of the theory of special relativity. Yet all attempts to find ways to send information faster than light have failed, and in that respect it appears that quantum mechanics respects special relativity. Is the world really non-local or is this just an anomaly of the theory? Quantum mechanics is used to calculate probabilities, it does not necessarily give an absolute answer. If we know the starting state and all interactions for a system, we can calculate the probabilities for the different measurement results for all time, but we cannot say that we will definitely measure a given value. There is an element of randomness in the quantum world that means even with perfect knowledge we cannot predict exactly the outcome of a measurement, but is that really how the world is? Is quantum mechanics just a calculational tool, and does there exist a deeper theory that predicts the outcome of all experiments with certainty? Entanglement is a key concept of quantum mechanics and at the heart of these philosophical questions.

Quantum mechanics is different from classical physics and in recent years these differences have been exploited to construct computational algorithms that are far better than their classical counterparts. Algorithms are recipes for solv-

ing problems, there are algorithms for example for adding two numbers and the special properties of quantum mechanics can for instance be used to factorize large numbers faster than any classical algorithm [24]. These information theoretical advances along with the superiority of using quantum systems to model other quantum systems, have led to a great interest in quantum computers, computers manipulating the smallest parts of nature to process information. In this new quantum information theory, entanglement is an important phenomenon and it needs to be better understood. We need ways to see if quantum systems are entangled, what kind of entanglement they have and how much of it there is.

My work in this area has been twofold. I have simulated quantum computers modeling quantum systems, and created a quantum compiler that can take any two-body fermionic Hamiltonian and output all the operations a quantum computer must perform to simulate the time evolution of that fermion system.

The other part of my work has been part of an effort to understand entanglement geometrically, for more on this approach see [5]. Familiar terms such as angles and distances, planes, curved lines or straight lines, have been used to describe quantum states. A numerical algorithm has been developed for finding the closest non-entangled state to a quantum state. An alternative and geometrically more accessible proof of the sufficiency of a positive partial transpose for separability in two two-level systems has been given. A numerical approach, with a direct geometrical explanation, has been used to find bound entangled extreme points of the convex set of positive partial transpose matrices. All these terms will be explained further on in the thesis.

1.1 The quantum world

Quantum mechanics was developed in the beginning of the previous century to answer questions that troubled physicists at the time. This first part is an overview of the history of quantum mechanics, showing the different aspects of nature that was revealed as the theory was understood: **superpositions of states**, **non-commuting variables** leading to **uncertainty relations** and **measurements**, with resulting **collapse of the state**. The features of quantum mechanics lead to philosophical questions and is seemingly at odds with our understanding of classical physics. The debate around these questions led to the concept of entanglement, which was understood to be at the heart of the differences between classical and quantum physics. Years after the concept of entanglement was introduced, John Bell [4] came up with a thought experiment that, when successfully performed 18 years later, showed that quantum mechanics was more than an uncannily successful calculational tool. Advances in technology have led to the rise of quantum computers and quantum information theory, where entanglement is a key concept

and a crucial resource, no longer just an item for philosophical debates.

The new theory

The first idea that led to quantum mechanics was the quantization of light and the energy levels of atoms. Planck solved the problem of the black body radiation spectrum by making an assumption that there were a set of oscillators that sent out radiation with discrete energy levels, $E = nhf$, where n are positive integers. Later Einstein explained the photo-electric effect by using the same idea and said that light was made up of particles, photons, with a discrete package of energy, the same as Planck's oscillators. Then Bohr gave a very good explanation of the spectrum of the hydrogen atom by assuming discrete energy levels,with the energy proportional to $1/n^2$, where again n was an integer. These ideas all showed the same thing, that these problems were solved by thinking of the smallest parts of nature as quanta, discrete packages, not continuous variables. The quantum is one of the main concepts in quantum mechanics, as the name shows.

In Newton's days light was thought to be made up of particles, but when Maxwell formulated his equations that explain everything about classical light, the view changed to light being waves in the electromagnetic field. When Einstein came up with the photons, where light again was particles, de Broglie thought that maybe all particles were waves as well, that the physically manifest objects we touch each day also are waves with a wavelength inversely proportional to the momentum. Waves follow wave equations and these are linear so that superpositions of solutions of the equations also are solutions. When we go swimming and two waves approach each other, we see constructive or destructive interference, the total wave when they are atop each other is a superposition of the other two waves. The Schrödinger equation governs the time evolution of all quantum mechanical systems, and like all linear differential equations it allows superpositions of solutions. This means that particles, once thought to be absolute real objects that either were or were not, can be in a superposition of different observable states.

This is seen indirectly in the double slit experiment, where electrons pass through two slits that are closer to each other than the de Broglie wavelength of the electrons. This results in a case where the electron state is a superposition of the state where the electron pass through the first slit and the state where the electron pass through the second slit. Still, every electron hits one and only one point on the detector wall on the other side of the slits. When it hits it behaves as a single particle, only when measuring the impact hits of many electrons do we see an interference pattern. This shows that the electron interferes with itself as a wave when it passes through the slits, it follows both paths at the same time, but when measured upon appears as a particle.

Heisenberg formulated quantum mechanics as a matrix algebra instead of the

partial differential equation approach of Schrödinger. Variables are represented by matrices which do not necessarily commute, and when they do not there is a relation between the uncertainties in measurements upon these variables. This uncertainty relation means it is impossible to measure exactly the position and momentum of a particle at the same time. It is not just difficult, for small enough systems quantum mechanics clearly states it is in principle impossible. This is a foreign concept for classical physics where everything can be determined exactly.

When measurements are done one never measures a superposition of values, unlike the superposition of waves in water and air. Either we find the electron there or not. Before a measurement the system is in a superposition of states and afterwards it collapses to one of the states allowed by the variable measured. What quantum mechanics can calculate is the probability for each of these outcomes, it cannot say which of the possible outcomes it will be. States exist which we cannot directly see, and our measurements directly influence the system we measure. Measurements do not need a conscious observer however, only the possibility of a measurement being performed is required for a state to collapse. What actually happens to a system measured upon, and what measurements really are, are hard questions that people have tried to answer with different interpretations of quantum mechanics.

1.1.1 Can quantum-mechanical description of physical reality be considered complete?

This was the title of the famous EPR particle by Einstein, Podolsky and Rosen from 1935 [9]. The authors used examples from quantum mechanics to debate whether it could be a complete theory of nature under what they assumed to be reasonable criteria for a sound description of reality.

They argued that in a complete theory there should be an element corresponding to each element of reality, and they further said that a sufficient condition for the reality of a physical quantity, was the possibility of predicting it with certainty, without disturbing the system. They gave an example of an entangled (the explanation will come later, at the time of this article it was not yet a term in use) system of two particles. One could either measure the position or the momentum of particle one, and because of the uncertainty relation that meant that the other quantity was not well defined for particle one. For this entangled state, measuring the position of particle one and causing the state to collapse, meant that the position of the second particle was also well defined, but not the momentum. If one instead measured the momentum of particle one, the momentum of particle two was well defined, but not the position. In the first case, they said, the position of particle two was an element of reality, in the second case the momentum was an

element of reality. However, since they also assumed locality, the measurement on particle one could not affect particle two if it was far away. This meant that both the position and the momentum were simultaneously elements of reality, in contradiction with the uncertainty relation of quantum mechanics.

Their definition of reality and locality led to this contradiction, and because of that they concluded that the state in quantum mechanics does not provide a complete description of nature, and ended their article by saying:

> While we have thus shown that the wave function does not provide a complete description of the physical reality, we left open the question of whether or not such a description exists. We believe, however, that such a theory is possible.

1.1.2 Entanglement

Schrödinger further illustrated the special and counter-intuitive properties of quantum mechanics in an article that followed EPR's, where he coined the term entanglement. Schrödinger came up with a thought experiment, Schrödinger's cat: A cat is placed inside a box with a bomb, the box separates the cat and the bomb completely from the rest of the world and the bomb is set to go off when a radioactive nucleus spontaneously decays, see fig. 1.1. After a time equal to the half-life of

Figure 1.1: The physical state is a superposition of the two observable states, but when we measure we either see a healthy cat or a cat blown to pieces. The cat and the bomb are said to be entangled.

the nucleus there is a probability one half that the bomb has gone off and probability one half that it has not. If they are completely separated from the outside world and there is no way for an observer to see if the bomb has gone off or not, the cat exists in a superposition of two states, it is neither alive nor dead. Yet it is not possible to directly test this, since the state will collapse to dead or alive when measured upon. Schrödinger said the cat and the bomb were now entangled.

1.1.3 Bell's answer

Do particles live in superpositions or is simply quantum mechanics an incomplete theory that gives us probabilities just because we do not know enough?

Bell came up with a beautiful argument and experiment in 1964 [4], as he had studied the two criteria put forth by EPR and was not satisfied with the Copenhagen answer to EPR's question. He used the entangled state known as a singlet, whose special features are more easily seen than the state used by EPR. The singlet consists of two two-level systems, for example two photons that can have two different values for their polarization in a given measurement, $+1$ or -1 with the corresponding quantum states $|+\rangle$ and $|-\rangle$. The singlet state of the two photons is $1/\sqrt{2}(|+\rangle \otimes |-\rangle - |-\rangle \otimes |+\rangle)$. When measuring the polarization of the photons in a given basis they will always have opposite values. If the first one is measured to be in the $|+\rangle$ state, that means the total state has collapsed to $|+\rangle \otimes |-\rangle$ and we know the second photon has polarization -1, and vice versa. When measuring in different angles they do not necessarily have opposite values for the polarization, but we can calculate the expectation values and the expectation values of the product of the two observables.

He came up with an experiment where one should measure the polarization of the photons in the singlet state at different angles. The expectation values of the polarization of the particles have correlations, that is that the product of the polarization expectation values is not the same as the expectation value of the product of the polarizations, $\langle AB\rangle \neq \langle A\rangle\langle B\rangle$. Measuring the correlation is the statistical way of determining whether or not two properties are dependent on each other. Bell took the EPR criteria of local realism and formulated the most general classical theory incorporating these principles. This theory is called local hidden variables. This is some unknown theory in which there are local variables following each particle, that describe deterministically what each measurement upon them will be. The quantum theory only gives us a probability distribution of the different outcomes, but this local hidden variable theory would tell us exactly what the outcomes would be if we knew the theory. This would have been the more complete theory EPR was seeking. It was designed so that the values of the observables were elements of reality as formulated by EPR, and the measurement results on one particle were independent of which angle the other photon's polarization was measured in. All correlations would come from the interaction between the particles when they were close at the beginning of the experiment.

Bell showed that for all possible local hidden variable theories, there would be an upper limit to the correlation of the polarization values, this is called Bell's inequality. Then he did the quantum mechanical calculation of the correlation and found that quantum mechanics gave an answer higher than this upper bound. Quantum mechanics and local realistic theories did not agree. In the words of

John Bell:

> In a theory in which parameters are added to quantum mechanics to determine the results of individual measurements, without changing the statistical predictions, there must be a mechanism whereby the setting of one measuring device can influence the reading of another instrument, however remote. Moreover, the signal involved must propagate instantaneously, so that such a theory could not be Lorentz invariant.

Which were true? Bell came up with the experiment, it was now up to someone else to test it and find the answer: Does nature break the Bell inequality of the local realistic hidden variable theories?

Aspect and the Bell experiments

In 1982 Aspect [3] performed the experiment and gave the answer: The inequality was indeed broken, and quantum mechanics' status as a valid description of nature was strengthened. This was a quantitative answer that showed that quantum mechanics describes nature in a way that is impossible for any local hidden variable theory.

There have been proposed other Bell-type experiments, for example the three particle GHZ experiment [11], and lots of groups the world over have been busy verifying the results from Aspect, and they all show that nature breaks the Bell inequalities. There are two loopholes [17] in the Bell-type experiments however, groups around the world are doing more and more accurate experiments to close these loopholes and prove beyond a doubt no local realistic hidden variable theory can describe nature better. The so-called detection loophole was closed in for example the experiment in this article [22].

Quantum mechanics is not a local theory, and the state of the two particles in the Bell experiment is not dependent on the distance between them. The particles keep their state, which is a superposition of measurable states, even across 144 kilometers in the latest experiments [28]. The two particles are measured so quickly that it is impossible to send a signal from one to the other at the speed of light, nonetheless the two particles show stronger correlations than any local hidden variable theory can explain.

In the words of Paul Kwiat [16]:

> Quantum mechanics, though one of the most successful and fundamental theories in physics, encompasses phenomena at odds with our intuition. Most notable is the phenomenon of entanglement, which

> violates the fundamental classical assumptions of locality and realism, which respectively state that objects which are far apart do not instantaneously interact and that a description of the universe can exist which will predict the outcome of any experiment.
>
> At our lab, using a source of entangled photons, we are able to make a series of measurements which manifestly violate these assumptions, and therefore validate the quantum mechanical worldview. (The catch is that some assumptions go into these tests, leaving loopholes for local-realistic theories. Groups around the world, including this one, are currently working towards a loophole-free test which would finally vindicate the reality of quantum non-locality.)

Fortunately we had sceptics like Bell, who delved deeper into the theory and found a quantitative way of resolving the EPR "paradox". Scepticism led to insight into nature. How far should one go in interpreting quantum mechanics? Is it just a tool that tells us what comes out when we know what goes in? Bell's work has shown us that quantum mechanics is not just an inadequate description which has to resort to probabilities. In Anton Zeilinger's opinion [1]:

> Objective randomness is probably the most important element of quantum physics.

1.2 Quantum technology

As physicists gain more and more control of the quantum, new technologies arise. Today one can see, classify and move individual atoms around with the Atom Force Microscope [7]. One can create quantum dots, quantum wires, quantum cones. We have rotating Bose-Einstein condensates, ultra-cold Fermi gases, ion traps, NMR machines and more applications of the smallest parts of nature where quantum mechanics is needed to explain the effects.

With the ability to control two-level quantum systems, the smallest systems there are, we have the ability to perform quantum computations. Information is physical, this was the truth that Claude Shannon realized when he founded information theory in 1948 [23], and the quantum bits are both measures of quantum information and actual quantum two-level systems used for computation. By having an array of quantum bits, qubits, and performing quantum operations and measurements on them, one can perform any calculation a classical computer can. Not only can a quantum computer do everything a classical computer can, it can also in principle outperform it significantly in some cases.

The boundary between the classical and quantum world

The correspondence principle states that systems governed by quantum mechanics will become classical when large enough. This is not a very quantitative statement and there are several questions. Is it possible to create a Schrödinger cat state with an actual cat? How large particles can show interference patterns in a double slit experiment?

The quantum effects on microscopic systems are seen in larger and larger systems as technology advances. In Vienna, see refs. [2] and [18], single particle interference experiments have now been done with $C_{60}F_{48}$ bucky balls, large ball-shaped molecules of 60 carbon atoms and 48 fluor atoms. The group in Vienna hopes to do the experiment with viruses one day, maybe they will be able to create a superposition of life? These are not really double slit experiments, the particles pass through a fine grating, but the principle is the same and you get interference patterns showing the wave nature of the particles.

Violation of Bell type inequalities have now been done with polarization entangled photons over a distance of 144 km [28]. Photons were sent from La Palma to Tenerife and both violate the inequalities and were used to generate a quantum cryptographic key. This experiment raises hopes that entangled photons can be used in global quantum communication using satellites.

Physicists gain more and more control over the quantum, and the imagined border between the quantum and the classical world is steadily pushed towards our classical world. Zeilinger in a statement [1] has said that the border between quantum physics and classical physics is only a matter of money. This statement shows a belief some physicists have that there is no principle that stops us from making Schrödinger's cat.

What makes classical objects behave classically and not in superpositions? The answer seems to be decoherence. The idea that measurements cause a state to collapse to an observable state is difficult, and is referred to as the measurement problem. A measurement of a variable here does not mean an active action from a conscious observer, it means that the system interacts with the environment in such a way that it would in principle be possible to measure the variable. This causes state collapse, or decoherence. Large objects have many internal degrees of freedom and there are many ways they can communicate to the environment which path they follow in the double slit experiment, whether the spin is up or down and so on. Decoherence is the loss of superposition due to transfer of information from the system to the environment. It is probably impossible to do the Schrödinger's cat experiment with a cat, but there are ideas for experiments observing macroscopic entanglement and only time will tell how far we get, see [29] for an example.

1.2.1 Qubits

Today the control over the micro- and nanoscopic is so great that actual qubits have been built and tested. Yet there are many difficulties to overcome before we have a functioning quantum computer. Decoherence is the main problem in making large systems of qubits, we want to be able to control all interactions between these smallest quantum systems and the rest of the world.

The first requirement for a qubit is that we can perform a universal set of operations on it, a small set of operations that when put together can approximate any unitary operation, which we can use to perform any calculation a computer can perform. In addition we have to be able to measure upon the qubit. Next we need to make a registry of qubits, all the qubits must be able to be addressed individually. We need to be able to perform the operations on them one at a time, and we need to be able to measure them one at a time. That is not enough however, we also need to be able to perform two qubit operations on any two of the qubits. When we can do this we need to scale the system, make it larger and larger so that we in the end can make a full fledged quantum computer.

One example of experimental qubits are the ion qubits, charged atoms that can be manipulated by magnetic fields, such as lasers, and can be moved around individually. When cooled the ions become effective two-level systems, in the sense that there are two energy levels, decoupled from the motion, that can be accessed. Lasers can be used to tune the two-level system yielding single qubit operations. With two ions next to each other, lasers with wavelengths longer than the distance between them can perform two qubit operations. To build a quantum register, a set of qubits which can be manipulated to perform quantum computation, we need several ions which can be addressed individually and in pairs. In fig. 1.2 we see four ions in an experimental quantum register. The ions can be moved en masse to the right and to the left, but since the ions are closer than the wavelength of the lasers, the ions that need to be addressed individually must be moved away from the others. Individual ions are then moved up the shafts to an operation area, here lasers can address single ions without disturbing the rest. To perform two qubit operations the two ions in question are moved into the operation area next to each other and then manipulated using lasers. This is an amazing control of the nanoscopic and our control of the smallest parts of nature will have to become even greater for us to be able to build a complete working quantum computer.

Quantum communication

A quantum computer can perform calculations, extract classical information and send this using existing information technology. Quantum communication however, is to send qubits to transfer quantum states, not just numbers. We can, for

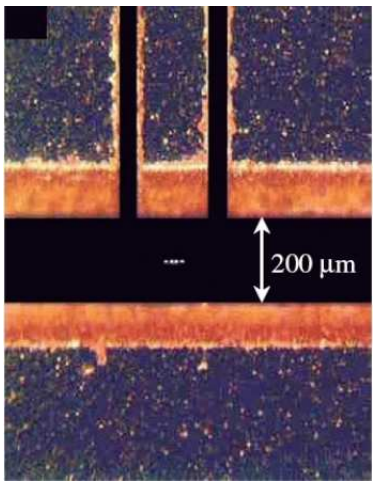


Figure 1.2: Here we see four ions trapped in a qubit register. Each one is a qubit.

example, encode a state on a set of ion qubits, perform some operations on them and then send the quantum state to another quantum computer. This we can do for example by having ion qubits interact with photonic qubits and send these through the air or in fibreoptic wires. Photons can be used as qubits since their polarization is a two-level quantum system. When they reach the other end, the photons interact with the ions there and transfer the quantum state using quantum teleportation. Quantum communication also opens for quantum cryptography, which is provably secure from eavesdropping. It is secure because it is possible to distribute cryptographic keys over an open channel, and if someone is listening one can detect this and discard the key [19].

Quantum communication uses entanglement in many forms as a resource, quantum teleportation is the best known example. Here two collaborators share a set of qubits, Alice and Bob each have one half of a set of qubit pairs in the singlet state. Then Alice can send a quantum state to Bob by performing operations on her own qubits only, and then sending Bob some classical information. Bob then uses that information to perform the proper measurements on his system which results in his qubits being in the state Alice wanted to teleport. The entanglement in the Bell pairs is lost after the teleportation is complete, and the resource needed to perform the task has been used.

Quantum calculations

By quantum calculations we mean calculating properties of quantum systems. To effectively control a quantum system one has to be able to make very good

predictions of the behavior of the system. In the same way as Newtonian mechanics must be used in a large scale to compute essential properties of a bridge before construction, quantum calculations are essential for quantum technology. As we gain more and more control over the smallest parts of nature, and as more and more of the basic parts of the quantum theory are understood, we need more and more computing power to simulate quantum systems. The dimensional curse means that in all general cases, simulating a quantum system on a classical computer is a hard task. By hard we mean that the computational complexity, i.e. number of operations and amount of memory needed, is an exponential function of the size of the system simulated. This curse can be lifted when we instead use a quantum system to perform the simulation. To simulate N two-level quantum systems on a classical computer we need 2^N bits, an exponential amount. On a quantum computer however, we only need N qubits.

The promise for an exponential gain in computing power when using qubits to perform our quantum calculations seems to be enormous. This can be done either with proper quantum computers or dedicated quantum simulators that are specifically built for quantum calculations. Simulating a general quantum system is an operation that demands an exponential amount of operations however. To effectively simulate a system, i.e. find operations on the qubits that emulate the time evolution operator, we need good transformations between the system being simulated and the actual qubits. Effective simulation here means that the amount of operations needed is a polynomial function of the amount of qubits used to represent the system. This is the problem we have worked on in article IV, where we implemented a transformation that enables a quantum computer to simulate any two-body fermionic Hamiltonian.

NMR quantum computing is the most advanced today. It is not a scalable system, because it is impossible to find thousands of atoms in one molecule with different spectra that physicists can access, and so there can never be a complete NMR quantum computer, but it is a great system for testing the ideas of quantum computing. The first proper quantum computation was done with an NMR machine, implementing Shor's algorithm to factorize 15 into five and three using six qubits [27]. In 2006 two groups, see [8] and [31], published results where they used NMR qubits to simulate the pairing Hamiltonian. This demonstration of a quantum simulator used the Jordan-Wigner transformation and the methods we used in our article IV.

Chapter 2

Entanglement

Where in the theory of quantum mechanics does entanglement come from? To answer this question we must examine its postulates and examine the mathematical structure of the spaces where representations of physical states live.

In this chapter we aim to show where entanglement comes from in the quantum theory. We also show how density matrices are used to represent physical states and explain their properties. The density matrices live in a real vector space, which means they have a geometry and we can use terms such as distances and angles, straight lines and curved lines, planes and cross sections. We can use this geometry for visualization and to learn more of quantum mechanics.

In this chapter we first list the postulates of quantum mechanics, the rules that form the basis for all we do. We describe how superpositions of states on multiple systems lead to entanglement. Then we describe how we introduce statistical uncertainties into quantum mechanics through density matrices. The main purpose of this chapter is to enable the reader to understand the geometric picture we have used in our articles.

Our work in this field has been part of an effort to understand entanglement and the difference between a positive partial transpose and separability. In our work we have described the set of density matrices and found an alternative proof of the sufficiency of a positive partial transpose for separability in the case of two two-dimensional systems. We have developed tools for visualizing the set of density matrices by two-dimensional cross sections through the set of hermitian matrices. We have created and implemented a numerical algorithm for determining the distance to the closest separable state for a general density matrix. We use this algorithm to determine whether a state is entangled and to find the boundary of the separable states in two-dimensional cross sections. We also have developed an algorithm for finding the extreme points of the Péres set. Using this algorithm we find bound entangled positive partial transpose states through a random walk on a convex set.

2.1 The postulates of quantum mechanics

Postulate 1 *Associated to any physical system is a unit vector in a Hilbert space (a complex inner product space, that is complete under its norm). This state vector $|\psi, t\rangle$, completely describes the system, and it obeys the Schrödinger equation*

$$i\hbar\frac{\partial}{\partial t}|\psi, t\rangle = \mathbf{H}|\psi, t\rangle \tag{2.1}$$

where $\mathbf{H}$ is the Hamiltonian operator.

Postulate 2 *To any physical observable F there is associated a linear hermitian operator $\mathbf{F}$ on the Hilbert space describing the physical system.*

The operator associated with a generalized coordinate and the operator associated with its corresponding momentum are subject to this commutation relation

$$[\mathbf{x}, \mathbf{p}] = i\hbar.$$

Postulate 3 *If the system is in the state $|\psi\rangle$, a measurement of F will yield one of the eigenvalues of $\mathbf{F}$*

$$\mathbf{F}|\phi_n\rangle = f_n|\phi_n\rangle$$

with probability $|\langle\phi_n|\psi\rangle|^2$. The state of the system after the measurement will be $|\phi_n\rangle$ as a result of the measurement.

As a consequence of this the expectation value of an observable F in a system in the state $|\psi\rangle$ is

$$\langle F\rangle = \langle\psi|\mathbf{F}|\psi\rangle. \tag{2.2}$$

Since the Schrödinger equation is linear, a superposition of solutions is also a solution, and therefore all states can be represented by a superposition of eigenstates of $\mathbf{F}$,

$$|\psi\rangle = \sum_i |\phi_i\rangle = |\phi_1\rangle + |\phi_2\rangle + \cdots$$

If $|\psi\rangle = |\phi_n\rangle$ then the expectation value is the eigenvalue $\langle\phi_n|\mathbf{F}|\phi_n\rangle = f_n$.

The postulates tell us that physical states are represented by normalized complex vectors, and these can be superpositions of other vectors. This way we can have physical states which are neither up nor down, but something in between. When the electron moves through a double slit it does not move through either the left slit or the right slit, it moves through both at the same time. It interferes with itself and the resulting probability distribution of measurements on the wall opposite shows an interference pattern similar to waves. Yet when we actually look at the electron, i.e. measure upon it, we can only see it move through one of the two slits.

Tensor products

When we have a system of two or more particles, or a particle with different quantum numbers, for example an atom with both an energy level and a spin quantum number, we have different Hilbert spaces for these subsystems. The total Hilbert space is the tensor product of all the subsystem Hilbert spaces, e.g. in the case of two electrons, A and B, the Hilbert space of the whole system is $\mathcal{H} = \mathcal{H}_\mathcal{A} \otimes \mathcal{H}_\mathcal{B}$. The dimensions of the two subsystems are n_A and n_B, and the total dimension is $n_A n_B$, and we say we have an $n_A \otimes n_B$ system. The demand that an operator on the whole space of all systems should be linear in each subsystem leads to the tensor product structure. Instead of having an operator bilinear in the two subsystems $O(|\psi_A\rangle, |\psi_B\rangle)$, we have a linear operator on the tensor product space $O(|\psi_A\rangle \otimes |\psi_B\rangle)$.

In the following we will only concern ourselves with systems which consist of two subsystems, so-called bipartite systems, and not with multipartite systems. Entanglement is a word used to describe relationships between systems, and it has no meaning for systems with only one Hilbert space.

Product vectors and the Schmidt decomposition

A product vector is a vector which can be written as a tensor product of a vector from system A with a vector from system B. A vector for which it is impossible to find a basis where it is a product of vectors from the subsystems is not a product vector, such vectors represent entangled states. The Schmidt decomposition is a way of finding whether or not a given vector is a product vector, it gives you the minimal number of product vectors you need in a superposition to represent your given vector.

A general vector in a bipartite system can be written in a product basis. When we have a basis for each subsystem, we have a product basis for the complete system, in which the basis-vectors are tensor products of the basis-vectors of each subsystem, $|v_{ij}\rangle = |i\rangle \otimes |j\rangle$. A vector $|v\rangle$ in $\mathcal{H}$ can then be written as

$$|v\rangle = \sum_i^{n_A} \sum_j^{n_B} v_{ij} |i\rangle \otimes |j\rangle. \tag{2.3}$$

A basis does not have to be a product basis, all you need is a complete set of linearly independent vectors which may or may not be product vectors. The matrix v of coefficients is a complex $n_A \times n_B$ matrix, and the singular value decomposition [10] tells us that it can be written as $v = WDU^\dagger$, where W is an $n_A \times n_A$ unitary matrix, $U^\dagger$ is an $n_B \times n_B$ unitary matrix and D is a real $n_A \times n_B$ matrix. The matrix D has the same dimensions as v with only positive or zero values on

the diagonal and all off-diagonal elements zero. We also use the singular value decomposition in article I, see section 2.5.1. This means

$$|v\rangle = \sum_{ij}\sum_{\alpha} W_{i\alpha} D_{\alpha\alpha} U^{\dagger}_{\alpha j} |i\rangle \otimes |j\rangle \tag{2.4}$$

$$= \sum_{\alpha} D_{\alpha\alpha} \left(\sum_{i} W_{i\alpha} |i\rangle \right) \otimes \left(\sum_{j} U^{\dagger}_{\alpha j} |j\rangle \right) \tag{2.5}$$

$$\equiv \sum_{\alpha} \lambda_{\alpha} |e^{A}_{\alpha}\rangle \otimes |e^{B}_{\alpha}\rangle. \tag{2.6}$$

The vectors $\{|e^{A}_{\alpha}\rangle\}$ and $\{|e^{B}_{\alpha}\rangle\}$ form orthonormal bases in their corresponding Hilbert spaces.

Entanglement

The expectation value of an operator $\mathbf{F}$ is defined in eq. 2.2 to be $\langle\psi|\mathbf{F}|\psi\rangle$. When we have a bipartite system we can have operators which are tensor products of operators on each subsystem, $\mathbf{F} = \mathbf{A} \otimes \mathbf{B}$, or a global operator which is not. If we have a product operator it represents observables in each subsystem, for example the spin of electron a and the spin of electron b. The expectation value of a product operator in a product state $|\psi\rangle = |\psi_A\rangle \otimes |\psi_B\rangle$ is

$$\langle \mathbf{A} \otimes \mathbf{B} \rangle = \left(\langle\psi_A| \otimes \langle\psi_B| \right) \mathbf{A} \otimes \mathbf{B} \left(|\psi_A\rangle \otimes |\psi_B\rangle \right)$$

$$= \langle\psi_A|\mathbf{A}|\psi_A\rangle \langle\psi_B|\mathbf{B}|\psi_B\rangle = \langle\mathbf{A}\rangle\langle\mathbf{B}\rangle. \tag{2.7}$$

If two systems are dependent on each other, the expectation value of the product of two observables may not be the same as the product of the expectation values of the observables, $\langle\mathbf{AB}\rangle \neq \langle\mathbf{A}\rangle\langle\mathbf{B}\rangle$. For product states there are no correlations between the observables on systems A and B, this is not the case when the Schmidt number is greater than one. The Schmidt number is the number of non-zero coefficients in the Schmidt decomposition eq. 2.6, it represents the minimal number of product vectors needed to represent the state. If it is greater than one, the state is not a product state and there will generally be correlations, $\langle\mathbf{AB}\rangle \neq \langle\mathbf{A}\rangle\langle\mathbf{B}\rangle$.

There can be correlations between expectation values in classical systems as well, but only when there is some uncertainty involved. When we have a state vector to describe the state we know all there is to know about the system, and yet we can have correlations between local observables. This is because the state is defined globally, but when we want to describe it only for one of the subsystems we do not know everything. We say the subsystems are entangled. A state

described by a vector which is not a product vector is entangled, in section 2.2 we give an example of this.

In the next section we will describe how we include uncertainty about the total state into the quantum formalism. When we are not sure what state vector represents the whole system we say we have classical or statistical correlations. Correlations that arise from the entanglement between subsystems are said to be quantum mechanical.

2.2 Density matrices

Quantum states are vectors in a Hilbert space, when we have such states we can in principle know all there is to know about the system. When we in addition add classical uncertainty we allow for statistical mixtures of these vectors to represent the physical states. When the state of the system is a vector $|\psi_k\rangle$ with probability p_k, we describe the system by a density matrix, which is a probability distribution over projections onto vectors in the Hilbert space,

$$\rho = \sum_k p_k |\psi_k\rangle\langle\psi_k|. \tag{2.8}$$

A projection operator is defined by $P^2 = P$. It is really a density **operator**, but it is common to just call it a density matrix, even when we mean the operator and not the matrix in a given basis. There is ambiguity in how one decomposes the density matrix, there can be many different sets of vectors $\{|\psi_k\rangle\}$ which can be used to make the same density matrix. The number of vectors in the sum in eq. 2.8 can be finite or infinite.

The trace of a matrix is the sum of the diagonal elements, from this we define the trace of an operator by taking its matrix elements in any basis and summing along the diagonal. The trace is invariant under unitary transformations because the trace is cyclic and as such all bases give the same result, $\mathrm{Tr}(\mathbf{U}\mathbf{A}\mathbf{U}^\dagger) = \mathrm{Tr}(\mathbf{A}\mathbf{U}^\dagger\mathbf{U}) = \mathrm{Tr}(\mathbf{A})$. The trace of an outer product of two vectors is then

$$\mathrm{Tr}\,|\psi\rangle\langle\phi| \equiv \sum_i \langle i|\psi\rangle\langle\phi|i\rangle = \sum_i \langle\phi|i\rangle\langle i|\psi\rangle = \langle\phi|\psi\rangle. \tag{2.9}$$

The first postulate of quantum mechanics states that the state vectors are normalized, $\langle\psi|\psi\rangle = 1$, which means that the trace of the density matrix is $\mathrm{Tr}(\rho) = \sum_k p_k \mathrm{Tr}(|\psi_k\rangle\langle\psi_k|) = \sum_k p_k \langle\psi_k|\psi_k\rangle = \sum_k p_k$. For this to be a probability distribution the weights p_k have to be positive and their sum has to be one. This means the trace of a density matrix has to be one,

$$\mathrm{Tr}(\rho) = 1. \tag{2.10}$$

The hermitian conjugate of a density matrix is

$$\rho^\dagger = \sum_k p_k (|\psi_k\rangle\langle\psi_k|)^\dagger = \sum_k p_k |\psi_k\rangle\langle\psi_k| = \rho \tag{2.11}$$

the matrix itself and therefore all density matrices are hermitian.

For any vector $|\psi\rangle$ we have

$$\langle\psi|\rho|\psi\rangle = \sum_k p_k \left|\langle\psi|\psi_k\rangle\right|^2 \geq 0, \tag{2.12}$$

which means all expectation values are zero or positive, and again all eigenvalues are zero or positive. Such a matrix is said to be positive semidefinite, which is denoted by $\rho \geq 0$.

What we have now are three criteria for density matrices which define the set geometrically. They are positive semidefinite hermitian matrices with trace one.

Pure and mixed states

We say that a given state represented by a density matrix is pure if there is only one term in the convex sum over projections onto states in the Hilbert space, see the definition in eq. 2.8. The density matrix is then a projection itself,

$$\rho = |\psi\rangle\langle\psi| \rightarrow \rho^2 = |\psi\rangle\langle\psi|\psi\rangle\langle\psi| = |\psi\rangle\langle\psi| = \rho.$$

For all density matrices the trace is equal to one, because the sum of the eigenvalues is one. The pure states only have one non-zero eigenvalue and it is one, $\rho = |\psi\rangle\langle\psi| \Rightarrow \rho|\psi\rangle = |\psi\rangle$. This means the sum of the square of the eigenvalues of a pure state is one,

$$\mathrm{Tr}(\rho) = \sum_k p_k = \mathrm{Tr}(\rho^2) = \sum_k p_k^2 = 1.$$

Since all the eigenvalues of density matrices also have to be positive, we see that the only possible solution to the above equation is if there is only one eigenvalue which is non-zero, which shows that only pure states have $\mathrm{Tr}(\rho^2) = 1$. This is also the maximal value of the trace of the squared density matrix and can be used as a measure of purity, called the degree of mixing. We say that if a density matrix is not pure, it is mixed, and the smaller the trace of the squared density matrix is, the more mixed and less pure the state is.

The minimal value of $\mathrm{Tr}(\rho^2)$ is found when all the eigenvalues are equal to $1/n$, where n is the dimension of the Hilbert space. This special state which is proportional to the identity matrix is called the maximally mixed state,

$$\rho = \widehat{\mathbf{1}}/n.$$

This density matrix has the special property that it can be written as a convex combination of all pure states with equal weights, and therefore all states are equally probable. This is therefore a state where we have minimal knowledge of the system, in other words the uncertainty is maximal.

Expectation values and reduced density matrices

All properties of a system can be found from the density matrix representing the state. The expectation value of an observable F in a system in a state defined by density matrix ρ is given by

$$\langle \mathbf{F} \rangle = \mathrm{Tr}(\rho \mathbf{F}) = \mathrm{Tr}\left(\sum_k p_k |\psi_k\rangle\langle\psi_k| \mathbf{F} \right) = \sum_k p_k \langle \psi_k | \mathbf{F} | \psi_k \rangle. \tag{2.13}$$

It is a statistical average of the expectation values in the different state vectors.

When the total system consist of two subsystems, $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B$, and we can only access system A, all information we have available is found in the reduced density matrix. The expectation values of a variable in system A are determined by the reduced density matrix for system A, $\langle \mathbf{F}_A \rangle = \mathrm{Tr}(\rho_A \mathbf{F}_A)$. The reduced density matrix ρ_A is found by tracing out the B system in a product basis,

$$\mathrm{Tr}_B \left(|a\rangle\langle b| \otimes |c\rangle\langle d| \right) = |a\rangle\langle b| \langle d|c\rangle, \tag{2.14}$$

$$\rho_A \equiv \mathrm{Tr}_B(\rho), \qquad \rho_B \equiv \mathrm{Tr}_A(\rho). \tag{2.15}$$

Entanglement and separability

We saw in section 2.1 that for pure states the question of whether a state is entangled is the same as asking if it is a product state. Separable means not entangled. A good picture to have in mind for understanding entanglement is a string of entangled or separable yarn. We say a state is separable if it is possible to write it as a probability distribution over product states,

$$\rho_S = \sum_k p_k \rho_k^A \otimes \rho_k^B. \tag{2.16}$$

The states ρ^A and ρ^B can be written as a convex sum of pure states themselves, and then ρ_S can be expressed as a convex sum of pure product states,

$$\rho_S = \sum_k p'_k |\psi_k\rangle\langle\psi_k| \otimes |\phi_k\rangle\langle\phi_k|. \tag{2.17}$$

The expectation value of the product of two local observables for a separable state is given by

$$\langle \mathbf{A} \otimes \mathbf{B} \rangle = \mathrm{Tr}(\mathbf{A} \otimes \mathbf{B} \rho_S) = \sum_k p_k \, \mathrm{Tr}(\mathbf{A} \rho_k^A) \, \mathrm{Tr}(\mathbf{B} \rho_k^B)$$

$$= \sum_k p_k \langle \mathbf{A}_k \rangle \langle \mathbf{B}_k \rangle. \tag{2.18}$$

This is the average of the product of the expectation values, the average over uncorrelated states. It is different from the product of the expectation values of operators $\mathbf{A}$ and $\mathbf{B}$ which is $\langle \mathbf{A} \rangle \langle \mathbf{B} \rangle = (\sum_k p_k \langle \mathbf{A}_k \rangle)(\sum_k p_k \langle \mathbf{B}_k \rangle)$. This is a probability distribution over states that we say have no quantum mechanical correlations between the subsystems, we call the correlations in a separable state statistical or classical. For a general mixed state it is difficult to distinguish between quantum mechanical correlations that arise from entanglement and statistical correlations that arise from global uncertainty.

Entanglement and the environment

A closed system is a system that does not interact with the world outside, such a system can be described by a vector in Hilbert space. When the system interacts with the environment we might have uncertainties arising from poorer measurements, or simply lack of knowledge. Such systems must be described using mixed density matrices.

The whole universe must be a closed system, or else there is something outside according to the postulates of quantum mechanics. When we divide our total system into system A and the environment B, the state of the whole thing should be described by a vector $|\psi\rangle$. If we have a mixed state in A we can always find another system B in which the total state is a vector. Then we say B purifies A since the total state is no longer mixed, but pure. If we only control system A and all our measurements are described by operators on $\mathcal{H}_A$, the expectation values we find are functions of the reduced density matrix $\rho_A = \mathrm{Tr}_B(|\psi\rangle\langle\psi|)$. If system A is properly separated from the environment, the total state for A and B should be a product state with no uncertainties or correlations. If they interact somehow, the two systems are entangled, that entanglement leads to a mixed reduced density matrix. All mixed density matrices can be viewed as reduced density matrices calculated from a state vector that "purifies" the density matrix. There is always some environment around our systems in which the total state is pure.

The singlet, an example of an entangled state

An entangled pure state is a state where we have all available global information. There is no uncertainty in the full state $|\psi\rangle$, and still we cannot tell exactly which state we have if we only have access to one subsystem. The best known example is the spin singlet, a maximally entangled state often used in Bell experiments, see section 1.1.3,

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle \otimes |1\rangle - |1\rangle \otimes |0\rangle). \tag{2.19}$$

The total density matrix for this state in the same basis is

$$\rho = |\psi\rangle\langle\psi| = \frac{1}{2}\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \tag{2.20}$$

The reduced density matrix for subsystem A is equal to the reduced density matrix of subsystem B,

$$\rho_A = \mathrm{Tr}_B \rho = \frac{\widehat{\mathbf{1}}}{2} = \rho_B. \tag{2.21}$$

The spin singlet is a state of two spin $1/2$ particles, upon measurement in the same basis, the two particles will always have spins pointing in opposite directions. The reduced density matrix is the maximally mixed state, which means we have minimal knowledge of the subsystem, there is a fifty-fifty chance of measuring spin up or spin down. We get exactly the same reduced density matrix from the mixed total state

$$\mathbf{B} = \frac{1}{2}\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \tag{2.22}$$

which represents the state where we know the spins are pointing in opposite directions, but we do not know which is up and which is down. When we measure spin up on one particle, we know the other one is pointing down in exactly the same way as for the singlet state. Yet the correlations in this mixed state are different from the pure entangled singlet state. The uncertainty coming from entanglement is not the same as just uncertainty.

2.3 Geometry of density matrices

Geometry can be used to picture how the set of density matrices looks, and one can use this insight to perhaps understand how one should quantify entanglement

or find proper entanglement criteria. The geometric view can also be important in the development of numerical methods for studying entanglement. Hermitian matrices form a real vector space as they can be added and multiplied with real scalars to construct other hermitian matrices. When we have a real vector space we can take two-dimensional cross sections to visualize parts of the space.

If we have an n-dimensional Hilbert space our state vectors have n complex dimensions, and we use $n \times n$ complex matrices to represent operators on that space. To find the dimension of the space of hermitian $n \times n$ matrices we count free parameters. We have n real parameters on the diagonal, and all the parameters on the upper half are just repeated on the lower half. On the upper half we have $1+2+\cdots+(n-1)$ complex numbers. In total we have $n+2(1+n-1)(n-1)/2 = n+n^2-n = n^2$ real parameters, which means that the real vector space of $n \times n$ hermitian matrices is n^2 dimensional. The density matrices have trace equal one, which is a linear constraint which means they are in a space one dimension lower than that of the hermitian matrices. The dimension of the set of density matrices is $n^2 - 1$. The density matrices themselves do not form a vector space, as the addition of two density matrices has trace equal two, not one.

The n^2-1 generators of $SU(n)$ along with the identity matrix form a basis for the $n \times n$ hermitian matrices. Each matrix basis element $\{\mathbf{E}_i\}$ can be represented by a vector $\vec{e}_i$ in the n^2 dimensional real vector space of hermitian matrices,

$$\mathbf{A} = \sum_i a_i \mathbf{E}_i \Leftrightarrow \vec{a} = \sum_i a_i \vec{e}_i. \tag{2.23}$$

Inner product and metric

We define an inner product for the hermitian matrices,

$$\langle \mathbf{A}, \mathbf{B} \rangle \equiv \text{Tr}(\mathbf{AB}). \tag{2.24}$$

By taking an orthonormal basis of hermitian matrices and expressing $\mathbf{A}$ and $\mathbf{B}$ as in eq. 2.23, and defining the inner product between the basis vectors $\vec{e}_i$ to be the same as the inner product between the basis matrices $\langle \vec{e}_i, \vec{e}_j \rangle = \text{Tr}(E_i E_j) = \delta_{ij}$, we get

$$\begin{aligned} \langle \mathbf{A}, \mathbf{B} \rangle &= \text{Tr}\left((\sum_i a_i \mathbf{E}_i)(\sum_j b_j \mathbf{E}_j) \right) \\ &= \sum_{ij} a_i b_j \, \text{Tr}(\mathbf{E}_i \mathbf{E}_j) = \langle \vec{a}, \vec{b} \rangle = \sum_i a_i b_i, \end{aligned} \tag{2.25}$$

This is the Euclidean inner product.

The inner product $\langle \mathbf{A}, \mathbf{B} \rangle = \text{Tr}(\mathbf{AB})$ leads to the natural norm $|\mathbf{A}| = \sqrt{\langle \mathbf{A}, \mathbf{A} \rangle} = \sqrt{\text{Tr}(\mathbf{A}^2)}$. This norm gives us the Euclidean metric on the real vector space of

hermitian matrices, it is called the Hilbert-Schmidt metric. The metric squared is simply the sum of the absolute square of all the matrix elements,

$$|\mathbf{A}|^2 = \text{Tr}(\mathbf{A}^2) = \sum_{ij} |A_{ij}|^2 . \tag{2.26}$$

This gives us distances between matrices,

$$|\mathbf{A}-\mathbf{B}| = \sqrt{\text{Tr}[(\mathbf{A}-\mathbf{B})^2]} = \sqrt{\sum_i |a_i - b_i|^2} = \sqrt{\sum_{ij} |A_{ij} - B_{ij}|^2}. \tag{2.27}$$

The Hilbert-Schmidt metric is the complex counterpart of the Frobenius metric for real matrices. For general complex matrices the Hilbert-Schmidt inner product is $\text{Tr}(\mathbf{A}^\dagger\mathbf{B})$, while the Frobenius inner product for real matrices is $\text{Tr}(\mathbf{A}^T\mathbf{B})$. Now we have distances and angles between density matrices and can begin to apply other geometrical terms.

The trace normalization

The trace equal one condition means that the density matrices are in a hyperplane in the space of hermitian matrices. This hyperplane is parallel to the trace equal zero hyperplane. It cuts through the axis in the direction of the identity matrix, $\widehat{\mathbf{1}}$, at the maximally mixed state, $\widehat{\mathbf{1}}/n$. This state is at the center of the density matrices. To see that the density matrices lie in a hyperplane we first notice that we can decompose any density matrix into a trace zero matrix and the maximally mixed state, $\rho = \widehat{\mathbf{1}}/n + \sigma$. Then the inner product $\langle \widehat{\mathbf{1}}, \sigma \rangle = \text{Tr}(\widehat{\mathbf{1}}\sigma) = \text{Tr}(\sigma) = 0$ shows that all trace equal zero matrices are normal to $\widehat{\mathbf{1}}$. All density matrices can now be formed by linear combinations of traceless matrices and $\widehat{\mathbf{1}}/n$ and we see that the density matrices lie in the hyperplane of trace equal one matrices.

The positive semidefinite cone

A positive semidefinite matrix is a matrix where all expectation values are greater than or equal to zero. When $\mathbf{A}$ is positive semidefinite we denote it by $\mathbf{A} \geq 0$,

$$\mathbf{A} \geq 0 \Rightarrow \langle\psi|\mathbf{A}|\psi\rangle \geq 0 \quad \forall \quad |\psi\rangle. \tag{2.28}$$

The condition that all expectation values be zero or greater means that all eigenvalues have to be zero or greater and vice versa. All positive semidefinite hermitian matrices reside in what is called the positive semidefinite cone. When $\text{Tr}(\mathbf{A}) = 0$ that means that the only possibility is $\mathbf{A} = 0$, the zero matrix. In other words in the trace equal zero hyperplane the positive semidefinite matrices is only a point.

For the hyperplanes $\mathrm{Tr}(\mathbf{A}) = a$ we find that the maximal eigenvalue is a for one eigenvalue and all other zeros. This means, when we use the Euclidean Hilbert-Schmidt metric for the Hermitian matrices, for a given hyperplane $\mathrm{Tr}(\mathbf{A}) = a$ there is a maximum distance from the axis in the direction of $\widehat{\mathbf{1}}$, beyond which the matrices are no longer positive semidefinite. It is called the positive semidefinite cone because for any semidefinite matrix σ, the matrix $a\sigma$ is also positive semidefinite for all positive a.

This is why it is called the positive semidefinite cone, even though it is not a symmetric cone equally wide in all directions.

The trace equal one hyperplane cuts through the positive semidefinite cone and the intersection gives us the density matrices. Every density matrix can be multiplied with a positive scalar a and that gives us a positive semidefinite hermitian matrix with trace a. If we relax the condition that the density matrices should have trace equal one, each of these rays $a\rho$ corresponds to a physical state.

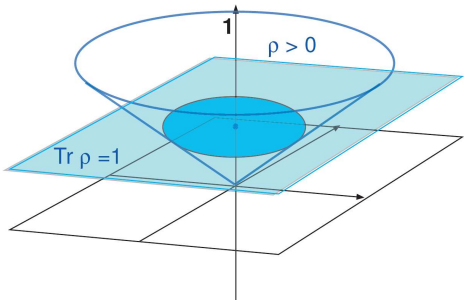


Figure 2.1: The space of Hermitian matrices. At the foot of the cone of positive semidefinite matrices is the zero matrix. The trace equal one hyperplane intersects the positive semidefinite cone at $\widehat{\mathbf{1}}/n$, yielding the set of density matrices.

Generating the set of density matrices

The elements $\{\mathbf{J}_a\}$ of a basis of $n \times n$ hermitian matrices are also the generators of the $n \times n$ unitary matrices, $\mathbf{U} = exp(i \sum_a \zeta_a \mathbf{J}_a)$, where ζ_a are real coefficients. The unitary transformation of a density matrix is another density matrix,

$$\tilde{\rho} = \mathbf{U}\rho\mathbf{U}^\dagger = \mathbf{U}\left(\sum_k p_k |\psi_k\rangle\langle\psi_k|\right)\mathbf{U}^\dagger$$

$$= \sum_k p_k \mathbf{U}|\psi_k\rangle\langle\psi_k|\mathbf{U}^\dagger = \sum_k p_k |\tilde{\psi}_k\rangle\langle\tilde{\psi}_k|. \qquad (2.29)$$

The eigenvalues of the density matrix are not changed by the unitary transformation which means the degree of mixing, $\mathrm{Tr}(\rho^2)$, is not changed either. All density matrices can be diagonalized by unitary transformations $\rho = \mathbf{U}\mathbf{D}\mathbf{U}^\dagger$, with $\mathbf{D}$ diagonal. This means that the set of diagonal density matrices in a basis can be rotated by unitary transformations into the whole set of density matrices. The diagonal matrices form an $n-1$ dimensional convex set, see section 2.4, by convex combinations of the diagonal pure states. The diagonal pure states are the diagonal states with only one non-zero entry. In the two-dimensional case all diagonal density matrices in one basis are parameterized by $x \in [0,1]$,

$$\mathbf{D}(x) = x \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + (1-x) \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}. \tag{2.30}$$

This simplex can then be rotated into a number of other dimensions to get the whole set of density matrices. A simplex is a generalized higher dimensional triangle. A 2 simplex is a triangle, a 3 simplex is a pyramid, or tetrahedron, and so on.

There are n^2-1 generators for the $SU(n)$ group of unitary transformations. The generators are the trace equal zero basis matrices for the space of hermitian matrices. Of these n^2-1 directions, $n-1$ correspond to matrices that commute with the diagonal matrices and do not rotate the simplex into other dimensions. For illustrative purposes let us consider the set of real 3×3 density matrices. In this case we have three pure diagonal states which form a triangle of all diagonal states, with $\widehat{\mathbf{1}}/3$ in the center,

$$\sigma_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \sigma_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \sigma_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{2.31}$$

For the real symmetric trace equal one 3×3 matrices, there are $2+3$ independent parameters, two on the diagonal and three off diagonal. The three non-diagonal basis matrices generate rotations of the triangle around the dotted lines in the triangle seen in fig. 2.2. Each rotation takes the triangle into a separate dimension creating the five dimensional set of 3×3 real density matrices. The σ_{23} matrix is in the middle of the line between σ_2 and σ_3,

$$\sigma_{23} = \frac{1}{2} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{2.32}$$

The unitary transformations

$$\mathbf{R}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & u_{22} & u_{23} \\ 0 & u_{32} & u_{33} \end{pmatrix}, \tag{2.33}$$

leave σ_1 and σ_{23} invariant, this means we rotate the triangle of the diagonal states around the line from σ_1 to σ_{23}. This cone is then rotated around the line from σ_2 to σ_{31} and so on. All the rotations must be performed on the whole object. After the first rotation we have a three dimensional cone, now this whole cone must be rotated into a fourth dimension, this four dimensional object must then be rotated into the fifth dimension.

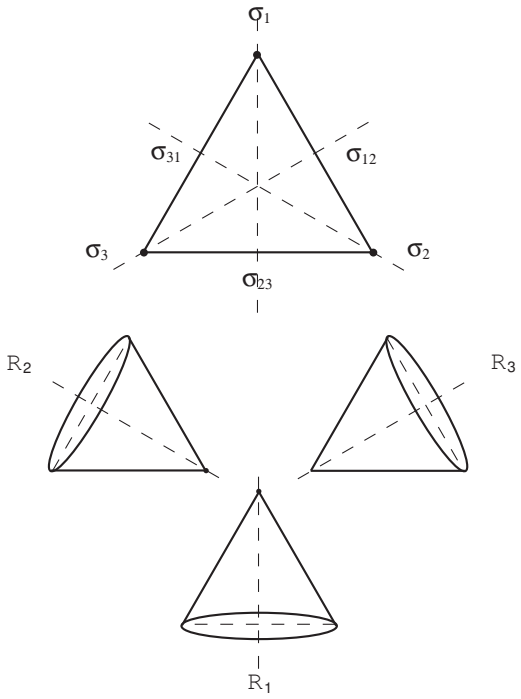


Figure 2.2: The 3×3 real density matrices. The triangle shows the diagonal states and the cones show which axes the diagonal states are rotated about to generate the whole five dimensional set of the $3 \otimes 3$ real density matrices.

In the case of two two-dimensional systems, two qubits, we have complex 4×4 matrices. The diagonal matrices now form a tetrahedron, a pyramid. This pyramid is then rotated into 12 other dimensions to generate the whole $n^2-1 = 15$ dimensional set of 4×4 density matrices.

2.4 Convex sets

In our work we have used convex optimization techniques to find the closest separable state to a density matrix. We have also developed an algorithm for finding

the extreme points of the convex set of positive partial transpose density matrices.

A set is convex if the line between any two points in the set consists only of points in the set. If x and y are points in the set then the point $\lambda x + (1 - \lambda)y$ is also in the set for all $\lambda \in [0, 1]$. A triangle is a convex set, a disc and a pyramid or tetrahedron are also convex sets, while the banana shape of a waxing or waning moon is not, see fig. 2.3.

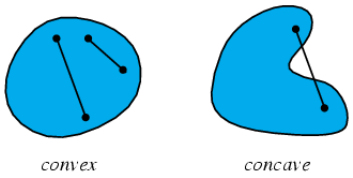


Figure 2.3: One convex and one concave set. Concave means not convex. [30]

Extreme points

The triangle is defined by its three corner points. The lines between the corners define the sides and lines between points on the sides fill out the whole convex set. Extreme points of a convex set are points which are in the set, but which do not lie on the line between any two other points in the set. If E is an extreme point then the only solution to the equation $E = \lambda x + (1 - \lambda y)$, with $x \neq y$, is for $\lambda = 0$ and $y = E$ or $\lambda = 1$ and $x = E$. A convex set is defined by its extreme points, the lines between the extreme points, and the lines between points on these again and so on, constitute all the points in the convex set.

Another equivalent way of defining a convex set is to use sets of inequalities which decide whether or not points are in the set. For example for the two-dimensional disc consisting of all points on and inside of the unit circle, the inequalities $x^2 + y^2 \leq 1$ for all x and y define the whole set. Even simpler, the four planes that each have one of the sides of a pyramid in them define the whole pyramid. These inequalities define hyperplanes for points on the surface of the convex set. The inequality in each point states that on one of the sides of the hyperplane there are no points that are in the convex set. All the hyperplanes together define the convex set.

Density matrices form a convex set, $\mathcal{D}$

The definition of a density matrix in eq. 2.8 shows us that the density matrices form a convex set, as the p_k are all positive and their sum is one, the density

matrix ρ is a convex combination of the pure states $|\psi_k\rangle\langle\psi_k|$. We call the set of density matrices $\mathcal{D}$.

From the definition we see that when there is only one pure state in the convex combination, $\rho = |\psi\rangle\langle\psi|$, we have one $p_k = 1$ and the rest are zero. When that is the case it is not possible to write ρ as a combination of any other density matrices, therefore the pure states $|\psi\rangle\langle\psi|$ are extreme points of $\mathcal{D}$.

The dimension of the set of pure states is found by looking at the free parameters in a pure state. A pure state is a projection onto a normalized complex vector in the Hilbert space: $\rho = |\psi\rangle\langle\psi|$ with $\langle\psi|\psi\rangle = 1$. In addition, as we see from the definition, any vector $e^{i\alpha}|\psi\rangle$ gives the same density matrix and the same physical state, the physical states are rays in the Hilbert space. When the Hilbert space has n complex dimensions there are $2n$ real variables in a vector. The normalization condition give one constraint, and the extraction of the phase factor to find the physical states gives one additional constraint. Which means the total dimension of the set of pure states is $2n - 2$, while the total dimension of the set of density matrices is $n^2 - 1$. All density matrices are formed as convex combinations of pure states, as they are a set of smaller dimension we know they must be hyperlines curving through the higher dimensional space so that lines between them span out the whole set of the density matrices. In the same way as a curved line in three-dimensional space carve out a three-dimensional convex set.

The other complementary way of defining the convex set is to find inequalities defining the boundaries. We know the density matrices are positive semidefinite, $\rho \geq 0$, which is the same as that all the eigenvalues have to be greater than or equal to zero. For each point on the boundary, each $n \times n$ density matrix on the boundary, we have n inequalities corresponding to the eigenvalues ρ_k of that matrix: $\rho_k \geq 0$ for $k = 1, 2, \ldots n$. For each point on the boundary of the convex set we can also define a plane, on the one side of which no points are density matrices.

Separable matrices form a convex set, $\mathcal{S}$

In the same way as the definition of the density matrices shows us that they are convex combinations of pure states, the definition of the separable matrices, eq. 2.17, show us that the separable density matrices form a convex set, $\mathcal{S}$, where the pure product states are the extreme points. Similar to $\mathcal{D}$, $\mathcal{S}$ is a convex hull of a lower dimensional set. The set of separable states has the same dimension as $\mathcal{D}$, n^2-1, as all states near enough to $\widehat{\mathbf{1}}/n$ are separable, while the set of pure product states has even lower dimension than the set of pure states. The pure states in an n dimensional system have $2n - 2$ free parameters. When we have the product of a pure state in the n_A dimensional system A with a pure state in the n_B dimensional system B, we have $2n_A - 2 + 2n_B - 2 = 2(n_A + n_B - 2)$ free parameters.

In this case, however, it is not possible to find sets of simple inequalities to define the borders of the set $\mathcal{S}$, even though we know all the extreme points of the set. In [13] it was shown that for large systems the separability problem is NP-hard. The separability problem is the problem of determining whether or not a density matrix is separable. That the problem is NP-hard means that it is not possible to find an algorithm with only polynomial complexity that solves the problem for any matrix. All algorithms for determining whether or not a large general density matrix is separable have exponential complexity.

2.5 Péres set and Péres criterion

The transposition is a positive map, in the sense that a transposed density matrix is also a density matrix, $\rho \geq 0 \Rightarrow \rho^T \geq 0$. The transposition must be defined in a given basis $\{|i\rangle\}$ so that the matrix elements of ρ are $\rho_{ij} = \langle i|\rho|j\rangle$, then $(\rho^T)_{ij} = \rho_{ji}$. The transposed of a 2×2 matrix is

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^T = \begin{pmatrix} a & c \\ b & d \end{pmatrix}. \tag{2.34}$$

The transpose is not, however, a completely positive map. When the Hilbert space is a tensor product of two Hilbert spaces, $\mathcal{H}_A$ with dimension n_A and $\mathcal{H}_B$ with dimension n_B, the dimension of the total space is $n_A n_B$. A map M is positive if $\rho' = M(\rho)$ is positive semidefinite, $\rho' \geq 0$. The map M is completely positive if the map $\widehat{\mathbf{1}} \otimes M$ is positive for all $\widehat{\mathbf{1}}$. If M operates on operators on Hilbert space B, then for all other Hilbert spaces A, the map $\widehat{\mathbf{1}} \otimes M$ should take any operator on the Hilbert space $\mathcal{H}_A \otimes \mathcal{H}_B$ to a positive semidefinite operator: $\rho' = (\widehat{\mathbf{1}} \otimes M)(\rho) \geq 0$. Completely positive maps are important in quantum mechanics, as all quantum operations on open systems must be represented by completely positive maps, see [19]. This is a reasonable physical criteria, as the result of a physical operation on a density matrix must be another density matrix, and there is always another system outside.

The partial transpose, $\widehat{\mathbf{1}} \otimes T \equiv P$, of a matrix is defined in a given product basis. The action on the matrix elements is that it transposes only one of the subsystems. A density matrix in a product basis is written as

$$\rho = \sum_{ij;kl} \rho_{ij;kl} (|a_i^A\rangle \otimes |e_j^B\rangle)(\langle b_k^A| \otimes \langle f_l^B|), \tag{2.35}$$

where i and k are indices corresponding to subsystem A, and j and l are indices for B. The matrix elements of the partial transpose of ρ are

$$(\rho^P)_{ij;kl} = \rho_{il;kj}, \tag{2.36}$$

the indices for the B system are interchanged. The density matrix ρ is an $n_A n_B \times n_A n_B$ matrix, but the matrix can be divided into an $n_A \times n_A$ block matrix, where each block is an $n_B \times n_B$ matrix. The action of the partial transpose is to transpose each $n_B \times n_B$ block. This is an example where n_A is two and $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ are $n_B \times n_B$ matrices,

$$(\widehat{\mathbf{1}} \otimes T)\left[\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix}\right] \equiv \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix}^P = \begin{pmatrix} \mathbf{A}^T & \mathbf{B}^T \\ \mathbf{C}^T & \mathbf{D}^T \end{pmatrix}. \tag{2.37}$$

Partial transpose and separability

The partial transpose of a separable density matrix is

$$\rho_S^P = \sum_k p_k \rho_k^A \otimes (\rho_k^B)^T, \tag{2.38}$$

and since the transpose of ρ_k^B is a density matrix, ρ_S^P is a density matrix. This means all separable matrices are positive under partial transposition,

$$\rho \in \mathcal{S} \Rightarrow \rho^P \in \mathcal{D}. \tag{2.39}$$

This is called the Péres criterion, for a density matrix to be separable, it must be positive under partial transposition.

2.5.1 Our geometric proof of the $2 \otimes 2$ sufficiency of the Péres criterion

In this section we review the proof of the sufficiency of the Péres criterion for the $2 \otimes 2$ and $2 \otimes 3$ case, and give an introduction to our alternative proof of the $2 \otimes 2$ case given in article I.

When we have a bipartite system $\mathcal{H} = \mathcal{H}_\mathcal{A} \otimes \mathcal{H}_\mathcal{B}$, where $\mathcal{H}_\mathcal{A}$ has dimension n_A and $\mathcal{H}_\mathcal{B}$ has dimension n_B, we say we have an $n_A \otimes n_B$ system. The Péres criterion is only a necessary condition for a general system, while for the case of $2 \otimes 2$ and $2 \otimes 3$ systems the criterion is also sufficient. That is, $\rho \in \mathcal{S} \Leftrightarrow \rho^P \in \mathcal{D}$ for $2 \otimes 2$ and $2 \otimes 3$ systems. If the partial transpose of a density matrix is positive for these dimensions, then the density matrix is separable. Why is the criterion only necessary and sufficient in these dimensions?

This was first proved using the Horodecki theorem [14], which states $\rho \in \mathcal{D}$ is separable if and only if the matrix $\rho' = (\widehat{\mathbf{1}} \otimes E)(\rho)$ is positive for all positive maps E. This is a necessary and sufficient condition for separability. A completely positive map would not give us any information using this theorem, because then

$\widehat{\mathbf{1}} \otimes E$ would always be a positive map. At first glance this theorem does not seem useful at all, how can one check all positive maps?

A theorem states, see [14], that in $2 \otimes 2$ and $2 \otimes 3$ systems, all positive maps can be expressed by

$$E = \Lambda_1 + \Lambda_2 T, \tag{2.40}$$

where Λ_1 and Λ_2 are completely positive maps and T is the transposition map, which is not completely positive. This is why it is only necessary to check whether a density matrix is positive under partial transposition to see if it is separable in the $2 \otimes 2$ and $2 \otimes 3$ systems.

In article I another proof was given for the sufficiency of the Péres criterion in $2 \otimes 2$ systems. By disregarding the trace equal one condition for the density matrices and looking at the rays in the positive semidefinite cone, one can prove this for the $2 \otimes 2$ case in a geometric and more accessible way.

The $2 \otimes 2$ system is composed of two two-dimensional systems. For one two-dimensional system, where we disregard the trace equal one condition, we have four basis matrices we can use to write any density matrix

$$\rho = \frac{1}{2}\left(\widehat{\mathbf{1}} + \mathbf{r}\boldsymbol{\sigma}\right) \rightarrow \frac{1}{2} x^\mu \sigma_\mu, \tag{2.41}$$

where $\sigma_\mu = [\widehat{\mathbf{1}}, \sigma_x, \sigma_y, \sigma_z]$ and $\mathbf{r}$ is a three dimensional real vector. If the four vector x^μ has a zero component x^0 different from one, the trace of the matrix ρ is no longer one. The relaxation of the trace equal one condition means we allow $x^0 \neq 1$.

The positivity condition $\rho \geq 0$ translates into

$$4\det(\rho) = (x^0)^2 - |\mathbf{r}|^2 = x^\mu x_\mu = g_{\mu\nu} x^\mu x^\nu \geq 0, \tag{2.42}$$

when we relax the trace condition. We use the language of special relativity to discuss the density matrices, with the four vector x^μ as either timelike inside the forward light cone or lightlike on the forward light cone. We know that the positivity of four vectors is preserved by Lorentz transformations, $SL(2,\mathcal{C})$ in this case,

$$\rho \rightarrow \tilde{\rho} = V \rho V^\dagger = \frac{1}{2} L^\nu_\mu x^\mu \sigma_\nu, \tag{2.43}$$

and we know we can reach any point on the interior of the light cone from any other point of the interior by Lorentz transformations. That is, we can use $SL(2,\mathcal{C})$ transformations to transform ρ into any 2×2 density matrix $\tilde{\rho}$.

In the case of the $2 \otimes 2$ system, the $SL(2,\mathcal{C}) \otimes SL(2,\mathcal{C})$ transformations $V = V_A \otimes V_B$ respect the product form of the separable density matrices, and

therefore conserve separability,

$$\rho' = V_A \otimes V_B \left(\sum_k p_k \rho_k^A \otimes \rho_k^B \right) V_A^\dagger \otimes V_B^\dagger$$
$$= \sum_k p_k \left(V_A \rho_k^A V_A^\dagger \right) \otimes \left(V_B \rho_k^B V_B^\dagger \right) \in \mathcal{S} \tag{2.44}$$

In article I it is proven that all density matrices in the $2 \otimes 2$ system can be transformed to the "standard" form

$$\tilde{\rho} = \frac{1}{4} \left(\widehat{\mathbf{1}} + \sum_{k=1}^{3} d_k \sigma_k \otimes \sigma_k \right). \tag{2.45}$$

This is accomplished by, among other things, using the singular value decomposition on the coefficient matrix of a general density matrix $\rho = c^{\mu\nu} \sigma_\mu \otimes \sigma_\nu$, this is also called a Schmidt decomposition of a matrix. As we see the standard form coefficients d_k span a three-dimensional space.

In the standard form we can identify the Péres set as the intersection between $\mathcal{D}$ and $\mathcal{D}^{\mathcal{P}}$, see fig. 4 in article I, and we find that the eight corners of the octahedron defining the Péres set are separable. Since all the extreme points of this convex set are separable, all points within are convex combinations of separable points and therefore are separable themselves. In this case we can see graphically that the Péres set and the separable density matrices are one and the same. And since the $SL(2,\mathcal{C}) \otimes SL(2,\mathcal{C})$ transformations preserve both the Péres set and separability, see article I, and can give us all the density matrices from the standard form, we know the Péres criterion is necessary and sufficient in the $2 \otimes 2$ system.

Positive partial transpose and entanglement

A density matrix which has a positive partial transpose and which is entangled has bound entanglement. This is not possible in $2 \otimes 2$ or $2 \otimes 3$ systems. This name comes from the fact that for it to be possible to distill entanglement from a state, it has to have a negative partial transpose, see [19]. There has not been found a negative partial transpose density matrix with bound entanglement, but neither has it been proven that they do not exist.

2.5.2 Our algorithm for finding the extreme points of $\mathcal{P}$

The Péres set, $\mathcal{P}$, is defined as the set of all density matrices which remain density matrices under partial transposition.

The partial transpose of a density matrix is a convex combination of partially transposed pure states,

$$\rho^P = \sum_k p_k (|\psi_k\rangle\langle\psi_k|)^P, \tag{2.46}$$

and we see this is also a convex set. This set, which is the partial transpose of $\mathcal{D}$, is called $\mathcal{D}^{\mathcal{P}}$. The Péres set is the intersection of these two sets,

$$\mathcal{P} \equiv \mathcal{D} \cap \mathcal{D}^{\mathcal{P}}. \tag{2.47}$$

The intersection of two convex sets is also a convex set, so $\mathcal{P}$ is a convex set.

We know from the Péres criterion that all separable density matrices are in the Péres set $\mathcal{S} \subset \mathcal{P}$, and we know that the matrices in the Péres set are density matrices, therefore

$$\mathcal{S} \subset \mathcal{P} \subset \mathcal{D}. \tag{2.48}$$

The extreme points of $\mathcal{S}$ are the pure product states, and they are also extreme points of $\mathcal{D}$. Since $\mathcal{P}$ is a subset of $\mathcal{D}$ the pure product states must also be extreme points of $\mathcal{P}$. We know, however, that for dimensions higher than six ($2 \otimes 3$) there must be other extreme points, or else the Péres set would be identical with the set of separable density matrices. The extreme points of $\mathcal{P}$ which are not pure product states are entangled, since they are extreme points they cannot be written as a convex combination of pure product states, other extreme points, which is the definition of separability. Another way to see that they must be entangled is to see that they would be extreme points in $\mathcal{S}$ as well if they were separable, and we know there are only pure product states that are extreme points in $\mathcal{S}$.

There are two basic ways of identifying a convex set, see section 2.4, through the set of extreme points, or through inequalities defining the boundaries of the set. For the set of density matrices, $\mathcal{D}$, we can use both methods, we know the pure states are the extreme points and we know the inequalities $\rho \geq 0$. For the set of separable states, $\mathcal{S}$, we have the extreme points, the pure product states, but have no inequalities defining the boundary. For the Péres set it is the other way around, we do not know all the extreme points, but we know the inequalities defining the boundary, namely $\rho \geq 0$ and $\rho^P \geq 0$.

In article III we have developed a numerical algorithm that moves in random directions from a point in $\mathcal{P}$ until it stops in an extreme point. To visualize how it works imagine a pyramid, a three dimensional convex set defined by the four corners, the extreme points. We start inside the pyramid and move in a random direction, then when we hit one of the sides we stop so we do not move outside the pyramid. Now we constrain ourselves to moving only on that side of the pyramid, a move in a random direction will now take us to one of three edges of that side. Constraining ourselves to that edge, we can either move forward or backward, and end up in one of the two corners of that edge. When we are in the corner there

is no way we can move that does not take us out of the pyramid or onto a higher dimensional side. See fig. 2.11 for a cross section involving extreme points of the Péres set, the explanation of the figure will come later.

2.6 Algorithm finding closest separable state

As entanglement is necessary for and used by processes in quantum computation, quantum teleportation and quantum cryptography, one can view it as a resource. When one has a resource one wants to know when you have it in a sample, you want to classify it and you want to know how much you have. Therefore it is necessary to develop efficient ways of determining when a state is entangled, i.e. not separable.

A metric is a way of determining distance between points in a set, there are several different metrics that can be used on the set of density matrices, not just the Hilbert-Schmidt metric, $|\mathbf{A}-\mathbf{B}|^2 = \mathrm{Tr}[(\mathbf{A}-\mathbf{B})^2]$. The metric lets us say that the distance between matrix $\mathbf{A}$ and $\mathbf{B}$ is $|\mathbf{A}-\mathbf{B}|$.

This leads to the geometric separability criterion: If one can construct a separable state that is close enough to ρ in a given metric, then ρ is separable,

$$\left|\rho - \sum_k p_k |\psi_k\rangle\langle\psi_k| \otimes |\phi_k\rangle\langle\phi_k|\right| < \delta \Rightarrow \rho \in \mathcal{S}. \tag{2.49}$$

In other words, if we find a separable state with distance zero to ρ, that state is ρ and ρ is obviously separable.

In article II we have constructed an algorithm to find the closest separable density matrix to a given state, with respect to the Hilbert-Schmidt metric. If that distance goes to zero we know the state is separable, if it converges to a finite value, we know it is entangled. The algorithm is based on a simple idea, which was then improved upon yielding a substantial gain in speed which made the algorithm usable for small systems. This is an example where geometrical insight led to a usable algorithm.

We start with a density matrix ρ, and want to find the convex combination of pure product states that is closest to ρ in the Hilbert-Schmidt metric. The first idea was to draw random pure product states and minimize the distance from ρ to the line between the pure product state and the previous closest separable state, see fig. 2.4. For the first iteration we say $\mathbf{S}_1 = \widehat{\mathbf{1}}/n$, which we know is separable. We then generate a random pure product state, $\mathbf{P}_1$, by creating two random normalized vectors and taking the tensor product. Then we minimize the distance squared from ρ to the line between $\mathbf{S}_1$ and $\mathbf{P}_1$. Since the points on that line are the convex combination between two separable points we know they are

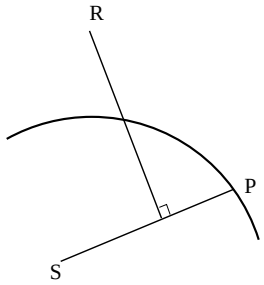


Figure 2.4: We want to find the shortest distance from R to the set of separable states. The true boundary is represented by the circle and S is a separable state, our best guess so far. We generate a random pure product state P, which we know also is separable, and find the point on the line between S and P which is closest to R. That is our new closest separable state. This first method shows the basic idea behind our algorithm.

separable themselves,

$$\min_{\lambda} |\rho - (\lambda \mathbf{S}_1 + (1-\lambda)\mathbf{P}_1)|^2 = \min_{\lambda} \mathrm{Tr}\left[(\rho - (\lambda \mathbf{S}_1 + (1-\lambda)\mathbf{P}_1))^2\right]. \quad (2.50)$$

The closest point is then $\mathbf{S}_2$ and we generate another random pure product state and find the closest point on the line between that state and $\mathbf{S}_2$ and so on. The distance to be minimized is a quadratic function of λ and is minimized analytically by differentiation with respect to λ. The reason this works is that the Hilbert-Schmidt metric has a simple quadratic form which enables us to quickly find the minimum on the line. Other metrics do not necessarily have a structure which allows for fast minimization on the line.

The two ways to improve upon this basic idea are to find better than random pure product states, and to minimize the distance from ρ to a larger convex set of these pure product states. The minimization is done by the conjugate gradient method [10] on a set of pure product states which vary throughout the program. Again this is possible because we use the Hilbert-Schmidt metric where the function to be minimized is $\mathrm{Tr}[(\rho - \sum_k \lambda_k \mathbf{P}_k)^2]$. This is a quadratic minimization problem in the variables $\{\lambda_k\}$ with constraints $\lambda_k \geq 0$ and $\sum_k \lambda_k = 1$.

To find more suitable pure product states we see that when we have the closest separable state, ρ_S, there are no pure product states ρ_P where the angle between $\rho - \rho_S$ and $\rho_P - \rho_S$ is less than $\pi/2$. Or, in other words, their inner product is always less than or equal to zero. If there were, one could take the line from ρ_S to ρ_P and we would find a closer point on that line, see fig. 2.5. The algorithm finds a local maxima of the inner product between $\rho - \rho_S$ and $\rho_P - \rho_S$ such that the pure product state found should give us the greatest decrease in distance to ρ.

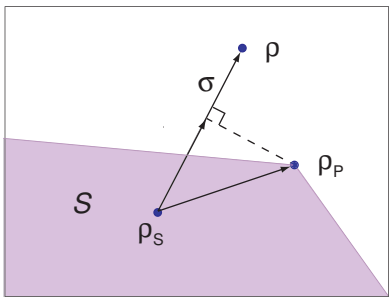


Figure 2.5: By choosing the pure product state ρ_P which maximizes the inner product between $\rho - \rho_S$ and $\rho_P - \rho_S$, we maximize the decrease in distance from ρ to our new closest separable state.

The algorithm is iterative and the distance it puts out is always decreasing, the distance found at each step is an upper bound on the true shortest distance. If the test state ρ is well within the boundary of $\mathcal{S}$, then the algorithm quickly converges to zero. If ρ is entangled and a good distance from $\mathcal{S}$ the algorithm converges to a finite value fast. In the case of density matrices on the border, the convergence is much slower, as it is increasingly difficult to find pure product states that give a substantial improvement.

2.7 Our visualizations

In the previous sections we have discussed various geometric properties of the density matrices. We have seen that they live in a real vector space, where we have an inner product which we use to find angles between vectors, and a metric to find distances between points. To visualize this we take two-dimensional cross sections through the space of hermitian matrices. In these cross sections we plot the boundary of the density matrices and the boundary of the partially transposed density matrices. $\mathcal{D}$ is defined by $\rho \geq 0$, this means that $\det(\rho) = 0$ at the boundary as the determinant of ρ is the product of the eigenvalues ρ_k: $\det(\rho) = \prod_k \rho_k$. Likewise, solutions of the equation $\det(\rho^P) = 0$ include the boundary of $\mathcal{D}^P$.

A plane is defined by three points in space, as long as they are not the same or on a line. We choose three density matrices and plot the two-dimensional plane which cuts through those three. In the plots below we always include the maximally mixed state, $\widehat{\mathbf{1}}/n$, as the origin. To find the x-direction we subtract $\rho_3 = \widehat{\mathbf{1}}/n$ from ρ_1 and get $\xi = \rho_1 - \widehat{\mathbf{1}}/n$ and then normalize to find the unit vector $\mathbf{I} = \xi/|\xi|$. Next we find $\mathbf{J}$ for the y-direction by using $\eta = \rho_1 - \widehat{\mathbf{1}}/n$ and $\mathbf{I}$, in such a way that

I and **J** are orthonormal. Now we have two unit vectors spanning out the vector space of the three density matrices we have chosen.We can now represent any hermitian matrix in this two-dimensional cross section by $\mathbf{A}(x,y) = x\mathbf{I} + y\mathbf{J} + \widehat{\mathbf{1}}/n$.

What we want to do is to plot all (x,y) points where $\det(\rho)$ or $\det(\rho^P)$ are zero. If we shift to polar coordinates, $x = r\cos\theta$ and $y = r\sin\theta$, and then choose a proper amount of θ values from zero to π, we must solve the equation

$$\det(\rho) = \left| r(\cos\theta\mathbf{I} + \sin\theta\mathbf{J}) + \frac{1}{n}\widehat{\mathbf{1}} \right| = 0 \tag{2.51}$$

to find the boundary of $\mathcal{D}$. By putting r on the outside we get

$$\det(\rho) = r\left| \cos\theta\mathbf{I} + \sin\theta\mathbf{J} - \left(\frac{-1}{nr}\widehat{\mathbf{1}}\right) \right| \tag{2.52}$$

$$\equiv r\left| \mathbf{B} - \lambda\widehat{\mathbf{1}} \right| = 0. \tag{2.53}$$

This is the eigenvalue equation for the matrix $\mathbf{B} = \cos\theta\mathbf{I} + \sin\theta\mathbf{J}$ with the eigenvalues $\lambda = -1/nr$. Finding the eigenvalues of $\mathbf{B}$ quickly gives us all the points r where $det(\rho) = 0$ for a given angle. Taking the partial transpose of $\mathbf{B}$ and finding the eigenvalues gives us the boundary of $\mathcal{D}^{\mathcal{P}}$.

When we choose three matrices for a cross section, and all of the states have zero determinant or zero determinant of the partial transpose, we cannot use this fastest procedure for plotting the boundaries. Instead we move in a radial direction from the center of our plot and diagonalize the matrices at each increment. Then when one of the eigenvalues of ρ or ρ^P become zero we plot the point. This technique can also be used for subspaces where all the density matrices have one or more eigenvalues equal zero. One simply checks whether an additional eigenvalue becomes zero and then plot the point.

The plots

In section 2.3 we saw that the space of density matrices is a hyperpyramid rotated into several new dimensions. There is no easy way to visualize this, but two-dimensional cross sections give a certain insight. In these cross sections we plot the lines where the determinant is zero in blue. The density matrices are then the points inside the blue line closest to the origin, $x = 0$ and $y = 0$, which is the maximally mixed state $\widehat{\mathbf{1}}/n$ in all these plots. The lines where the determinant of the partial transpose is zero are plotted in red.

When the three matrices commute the lines characterized by $\det(\rho) = 0$ will all be straight. When two matrices commute they can be diagonalized in the same basis, and two commuting matrices with one or more common eigenvalues equal

to zero will have a line between them that also has these eigenvalues equal to zero. As an example

$$x\mathbf{U}\begin{pmatrix}0.5 & 0 & 0\\ 0 & 0.5 & 0\\ 0 & 0 & 0\end{pmatrix}\mathbf{U}^\dagger + y\mathbf{U}\begin{pmatrix}0.9 & 0 & 0\\ 0 & 0.1 & 0\\ 0 & 0 & 0\end{pmatrix}\mathbf{U}^\dagger$$

$$= \mathbf{U}\begin{pmatrix}0.5x+0.9y & 0 & 0\\ 0 & 0.5x+0.1y & 0\\ 0 & 0 & 0\end{pmatrix}\mathbf{U}^\dagger. \tag{2.54}$$

Curved lines on the other hand mean non-commuting directions. This also goes for the partial transpose of the matrices, so that if the red lines, characterized by $\det(\rho^P) = 0$ are straight the partial transposes of the matrices spanning the cross section commute.

Pure states and distance to $\widehat{\mathbf{1}}/n$

When the dimension of the Hilbert space is n the distance from a pure state $|\psi\rangle\langle\psi|$ to the maximally mixed state is

$$\left||\psi\rangle\langle\psi| - \frac{\widehat{\mathbf{1}}}{n}\right| = \sqrt{\text{Tr}\left[(|\psi\rangle\langle\psi| - \frac{\widehat{\mathbf{1}}}{n})^2\right]}$$

$$= \sqrt{\text{Tr}\left[|\psi\rangle\langle\psi| - \frac{2}{n}|\psi\rangle\langle\psi| + \frac{\widehat{\mathbf{1}}}{n^2}\right]} = \sqrt{1-\frac{2}{n}+\frac{1}{n}}$$

$$= \sqrt{1-\frac{1}{n}} = \sqrt{\frac{n-1}{n}}. \tag{2.55}$$

All the pure states have the same distance from $\widehat{\mathbf{1}}/n$, this is the maximal distance for any density matrix since the pure states are the extreme points. For the $2\otimes 2$ case n is four and the distance to the pure states from the maximally mixed state is $\sqrt{3/4} = 0.866$, while for the $3\otimes 3$ case it is 0.943.

Two pure product states

The pure product states are separable, as is $\widehat{\mathbf{1}}/n$, so when we choose two commuting pure product states for our cross section we get straight lines and a complete overlap between the boundary of the density matrices and of the Péres set. The triangle in fig. 2.6 is a cross section through the pyramid of diagonal states in the set of 4×4 density matrices. The matrix $\mathbf{B}$ is the diagonal matrix with the first

entry equal to one and the rest zero, the density matrix corresponding to the pure product state $|0\rangle \otimes |0\rangle$. Matrix $\mathbf{A}$ has a one in entry $(2, 2)$, the second place on the diagonal, with the rest zero. This matrix corresponds to the pure product state $|0\rangle \otimes |1\rangle$.

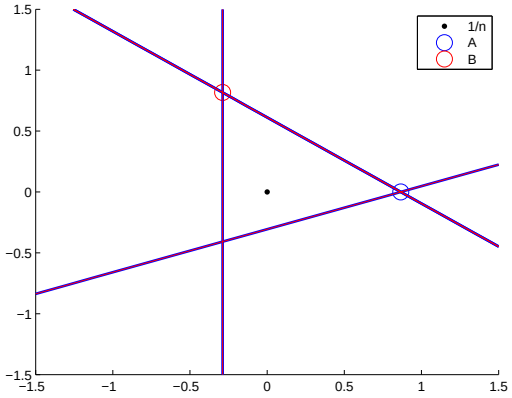


Figure 2.6: $\mathbf{A}$ and $\mathbf{B}$ are two commuting 4×4 pure product states. All points inside the triangle are mixed separable density matrices. Outside are the hermitian matrices which are not positive semidefinite and therefore are not density matrices.

The density matrices in this cross section are all the points inside the red and blue lines around the surrounding the origin, $\widehat{\mathbf{1}}/n$. They are all separable since this is the $2 \otimes 2$ case. The only pure states in this cross section are the points $\mathbf{A}$ and $\mathbf{B}$, all other states are closer to $\widehat{\mathbf{1}}$ than a pure state, see eq. 2.55.

Bell states

The Bell states are maximally entangled, which means they have the greatest distance to the set of separable states as well. In fig. 2.7 $\mathbf{A}$ is a Bell state $|\psi\rangle\langle\psi|$ called the singlet with $|\psi\rangle = 1/\sqrt{2}(|0\rangle \otimes |1\rangle - |1\rangle \otimes |0\rangle)$, and $\mathbf{B}$ is a pure product state corresponding to the $|0\rangle \otimes |0\rangle$ state.

In this plot the density matrices are all the points inside the blue triangle surrounding the origin, while the separable density matrices are the points inside the triangle which are also inside the red ellipse. We clearly see that the Bell state is farthest away from the separable states in this cross section. With these choices of

states the matrices $\mathbf{A}$ and $\mathbf{B}$ commute,

$$\mathbf{A} = \frac{1}{2}\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{2.56}$$

while their partial transposes do not

$$\begin{aligned} &\mathbf{A}^P\mathbf{B}^P - \mathbf{B}^P\mathbf{A}^P \\ &= \frac{1}{2}\begin{pmatrix} 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} - \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}\frac{1}{2}\begin{pmatrix} 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \end{pmatrix} \\ &= \frac{1}{2}\begin{pmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{pmatrix} \end{aligned} \tag{2.57}$$

That is why the blue lines are straight while the red ones have curved lines as well. The straight red line corresponds to the commuting subspace of the second and third rows and columns, and all points on the line have two zero eigenvalues in their partial transpose.

If we instead choose two commuting Bell states, the matrices and their partial transposes commute, see fig. 2.8.

This is a cross section through the tetrahedron in fig. 5 in article I. The red and blue parallelogram surrounding the origin is a cross section through the octahedron of separable states in fig. 5 in article I.

Typical cross section

The set of pure product states have dimension $2(n_A+n_B-2)$, as we saw in section 2.4. The set of separable states have full dimension n^2-1 and the boundary has one lower dimension, n^2-2. When we move in a random direction from the maximally mixed state at the center of the set of density matrices, we will never hit a pure state or a pure product state. This is because the boundaries of $\mathcal{D}$ and $\mathcal{S}$ have higher dimensions than the set of pure states and the set of pure product states. This is the same as sitting inside a triangle and moving in a random direction, we will always hit one of the sides, and never a corner, simply because a random direction means we can never aim that good.

In figures 2.9 and 2.10 we move in a random direction from $\widehat{\mathbf{1}}/n$ until we hit the boundary of the Péres set. That is, we move in a random direction in the space

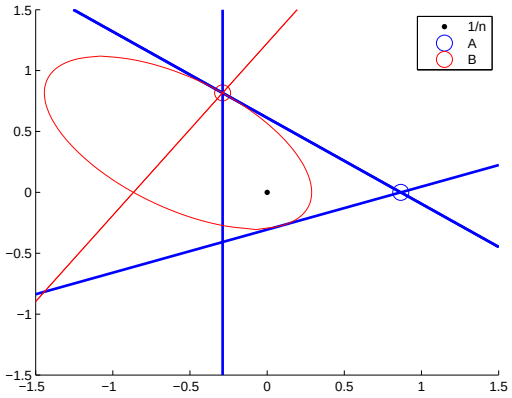


Figure 2.7: $\mathbf{A}$ is the singlet state and $\mathbf{B}$ is a pure product state. All points inside both the red ellipse and the blue triangle are separable. States in the triangle, but outside the ellipse, are entangled. Points outside the triangle are not density matrices.

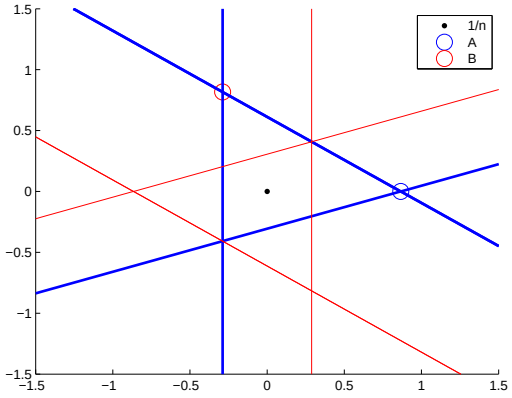


Figure 2.8: Two 4×4 Bell states, a cross section through the pyramide seen in fig. 5 in article I.

of hermitian matrices with trace one, until the matrix gets one eigenvalue equal to zero or its partial transpose gets one eigenvalue equal to zero. We do this two times getting two matrices on the boundary of $\mathcal{P}$. Along with $\widehat{\mathbf{1}}/n$ we have three matrices which we can use to make a two-dimensional cross section through $\mathcal{D}$ and $\mathcal{D}^P$. We have tested systems with $n_A = n_B = p$ for $p = 2, 3, 4, 5, 6, 7, 8, 9, 10$. In fig. 2.9 p is three, which means we have a $3 \otimes 3$ system. The black circle has

radius equal to the distance of the 9×9 pure states to $\widehat{\mathbf{1}}/9$, see eq. 2.55.

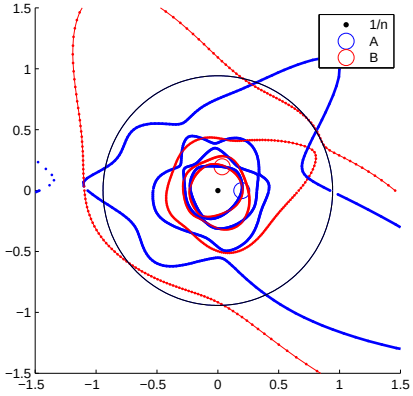


Figure 2.9: Two 9×9 random states on boundary of $\mathcal{P}$ with $\widehat{\mathbf{1}}/n$ at the origin. The black circle has radius equal to the distance to the pure states from $\widehat{\mathbf{1}}/n$, and illustrates that the boundary og $\mathcal{P}$ in general is much closer to $\widehat{\mathbf{1}}/n$ than to the pure states.

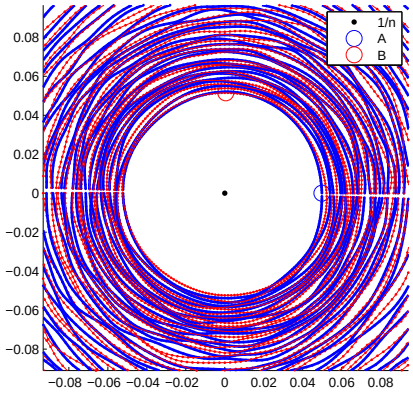


Figure 2.10: Two 100×100 random states on boundary of $\mathcal{P}$. The pure states are a distance 0.995 from $\widehat{\mathbf{1}}/n$, at the origin, while the distance to the boundary of $\mathcal{P}$ seems to be about 0.05.

In fig. 2.10 p is ten, the distance to the pure states from the origin is 0.995, while the approximate distance to the boundary of $\mathcal{P}$ in this cross section is 0.05.

Boundary of the separable states

In the case of the $3 \otimes 3$ and higher dimensional systems the boundary of the separable states is not the same as the boundary of the Péres set. We use our algorithm for finding the closest separable state in the Hilbert-Schmidt metric to find the boundary in a two dimensional cross section.

The algorithm gives an upper bound on the distance to the closest separable state, and the distance is always decreasing iteration for iteration. To find the boundary of the separable states in a two dimensional cross section we move from the boundary of $\mathcal{P}$ towards $\widehat{\mathbf{1}}/n$ radially, finding the boundary on each line separately. When we reach a point which we find to be separable we have located the boundary. For a point on the boundary, $\rho_{\delta\mathcal{P}}$, we parameterize the line between that point and $\widehat{\mathbf{1}}/n$ as $\rho(\lambda) = \lambda\rho_{\delta\mathcal{P}} + (1-\lambda)\widehat{\mathbf{1}}/n$. First we find the distance from $\rho(1)$ to the closest separable state, $d_0 = \min_{\sigma\in\mathcal{S}} |\rho(1) - \sigma|$. The separable state which is closest to $\rho_{\delta\mathcal{P}}$ will in general not lie in the two-dimensional cross section we have chosen, and the distance to the boundary of $\mathcal{S}$ in our plot will be greater than d_0. If this is not zero we move a distance d_0 towards $\widehat{\mathbf{1}}/n$ finding an appropriate λ_1. Then we find $\rho_1 = \lambda_1\rho_{\delta\mathcal{P}} + (1-\lambda_1)\widehat{\mathbf{1}}/n$. We now run the algorithm for ρ_1, finding an upper bound on the shortest distance to $\mathcal{S}$ and use this to move closer to $\widehat{\mathbf{1}}/n$ again. After some iterations we come to a point in our two-dimensional cross section where the distance to $\mathcal{S}$ is less than some bound, typically 10^{-5}. Either this point is separable or the boundary is so close that the difference is indiscernible in our plot.

In fig. 2.11 we have taken the Horodecki state, see [15] and article I, and found a positive partial transpose entangled extreme point of $\mathcal{P}$ by moving from this state. The Horodecki state was the first positive partial transpose entangled state to be found. The line through this extreme point and the Horodecki state end in another extreme point of $\mathcal{P}$ as the line hits the boundary of $\mathcal{P}$. The plot is then the cross section through the Horodecki state, these two extreme points and $\widehat{\mathbf{1}}/n$. The green star is the Horodecki state, while the two end points of the red line through the Horodecki state are the extreme points. The boundary of the Péres set gives us a light blue area of negative partial transpose entanglement. While our numerical method has been applied to the whole cross section and this gives us a dark blue area of entangled states within $\mathcal{P}$.

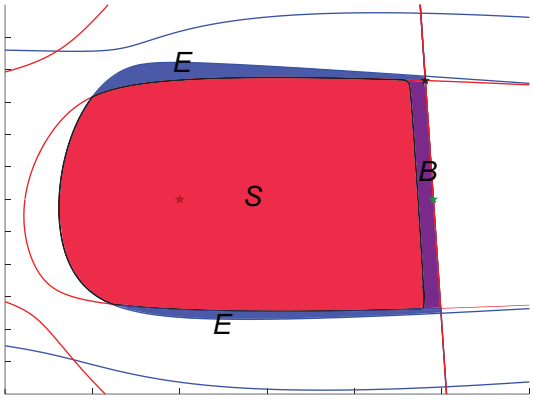


Figure 2.11: The numerical boundary of $\mathcal{S}$. The red area, S, show the separable states in this cross section. The E areas are the negative partial transpose entangled states. The B area in the cross section is the area of bound entangled states in the Péres set. The green star is the Horodecki state, the first positive partial transpose state found, and the two end points on the red line through the Horodecki state are extreme points of $\mathcal{P}$.

Chapter 3

Quantum computing

The basic unit of information in the standard model of quantum computation is the qubit, the quantum bit. Any suitable two-level quantum system can be a qubit, it is the smallest system there is with the least complex dynamics. The standard model of quantum computation is a model where two-level quantum systems are located at different points in space, and are manipulated by a small universal set of operations. These operations are called gates in the same fashion as operations on bits in classical computers are called gates.

Qubits are both abstract measures of information and physical objects. Actual physical qubits can be ions trapped in magnetic fields where lasers can access only two energy levels or the nuclear spins of some of the atoms in molecules accessed and manipulated by an NMR machine. Several other ideas have been proposed and some tested, see [19].

The computational basis

The computational basis for one qubit is $|0\rangle$ for the first state and $|1\rangle$ for the second, and for a set of qubits it is the tensor products of these basis states for each qubit. Below we write out the different basis states for a system of N qubits.

$$\begin{aligned} |0\rangle \equiv |00\cdots0\rangle &= |0\rangle \otimes |0\rangle \otimes \cdots \otimes |0\rangle \\ |1\rangle \equiv |00\cdots1\rangle &= |0\rangle \otimes |0\rangle \otimes \cdots \otimes |1\rangle \\ &\vdots \\ |2^N-1\rangle \equiv |11\cdots1\rangle &= |1\rangle \otimes |1\rangle \otimes \cdots \otimes |1\rangle \end{aligned} \tag{3.1}$$

This is a 2^N-dimensional system and we number the different basis states using binary numbers corresponding to the order in which they appear in the tensor

product. The different notations $|0\rangle|1\rangle$, $|01\rangle$ and $|0\rangle \otimes |1\rangle$ all mean the same. We can also represent the vectors as one-dimensional arrays,

$$a|0\rangle + b|1\rangle \equiv \begin{pmatrix} a \\ b \end{pmatrix}. \tag{3.2}$$

Qubit gates

Quantum computing is manipulating and measuring qubits in such a fashion that the measurement results solves the given problem. The quantum operations we need to be able to perform are a small set of elementary single qubit operations, or single qubit gates, and one universal two qubit gate, in our case we use the CNOT gate. To represent quantum computer algorithms graphically we use circuit diagrams. In a circuit diagram each qubit is represented by a line, and operations on the different qubits are represented by boxes. Below in fig. 3.1 we have an example of a quantum circuit where we show the time direction. The states $|a\rangle$

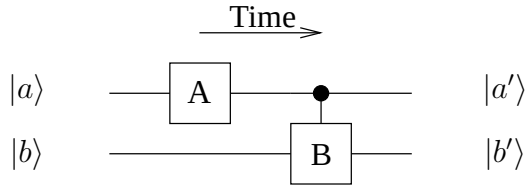


Figure 3.1: A quantum circuit

and $|b\rangle$ in the figure represent qubit states, one must remember that in general the total state will be a superposition of different qubit states.

A single qubit gate is an operation that only affects one physical qubit, e.g. one ion or one nuclear spin in a molecule. It is represented by a box on the line corresponding to the qubit in question.

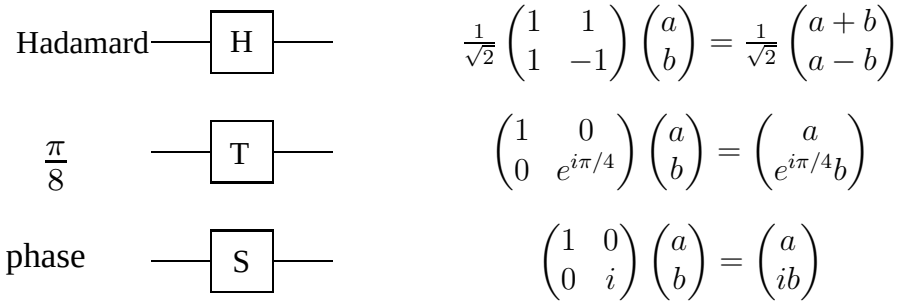
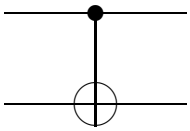


Figure 3.2: A set of three elementary single qubit gates

The matrices and vectors in fig. 3.2 show the action of the gates on a single qubit state. A two qubit gate is a box encompassing both the lines of the qubits it operates on. The $CNOT$ gate is a very important gate, it is an abbreviation of controlled not. It swaps the state of the second qubit if the first qubit is in the $|1\rangle$ state. In fig. 3.3 we show the quantum gate representing the $CNOT$ operation and a truth table, where A and B are the input basis states and A' and B' the output basis states.



A	B	A'	B'
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

Figure 3.3: $CNOT$ gate with truth table. The input basis states are in the A and B columns, while the output states for a given pair of inputs are shown in the A' and B' columns.

This is the effect of the $CNOT$ operation on basis states in the bra/ket notation,

$$
\begin{aligned}
|0\rangle \otimes |0\rangle &\rightarrow |0\rangle \otimes |0\rangle, \\
|0\rangle \otimes |1\rangle &\rightarrow |0\rangle \otimes |1\rangle, \\
|1\rangle \otimes |0\rangle &\rightarrow |1\rangle \otimes |1\rangle, \\
|1\rangle \otimes |1\rangle &\rightarrow |1\rangle \otimes |0\rangle.
\end{aligned}
$$

Finally we show the corresponding matrix representation ,

$$
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0
\end{pmatrix}. \tag{3.3}
$$

The three single qubit gates, the Hadamard gate, the phase gate and the $\pi/8$ gate, along with the two qubit CNOT gate can approximate any unitary operator on a set of qubits. They form a universal set and a physical realization of a quantum computer that can perform these operations on its qubits and measure them, can calculate anything any computer with a finite memory can.

All quantum operations, except the measurement, must be unitary. All unitary operations U are reversible, $UU^{\dagger} = \widehat{\mathbf{1}}$. This means that quantum computing is reversible, which is not the case for the classical computers used today, where some

gates have two bits input and only one output. Irreversible means it is impossible to go back and find the exact input state.

Bits vs. qubits

What are the chief differences between bits and qubits? First of all qubits are governed by quantum mechanics, this means that while bits are either 0 or 1, qubits can be in a superposition of both states, $|qubit\rangle = \alpha|0\rangle + \beta|1\rangle$. When we have a system of qubits we get entanglement between them because of superposition and the tensor product structure as explained in section 2.1.

Bits are only a measure in information theory. Qubits are both a measure in quantum information theory and physical objects upon which calculations are performed. Any fitting two-level quantum system can be a qubit.

3.1 Why quantum computers work

A universal Turing machine is a theoretical machine that can solve most algorithmic problems. That is the same as saying it can solve most problems that can be solved by following a recipe. A computer that we know, a pc for example, can in principle solve all problems a Turing machine can, except that the Turing machine has infinite memory. The Turing machine can model all computers, and to see if a proposed computer is universal, one must model it as a Turing machine. In 1980 Paul Benioff showed a model of a quantum mechanical Turing machine [6]. He showed that unitary operations on a quantum mechanical system gives a universal computer that can compute anything any Turing machine can.

Two-level unitary gates are universal

What is needed next is to find a way to implement any unitary operation on a system of qubits. The first step is to see that any unitary $n \times n$ matrix can be decomposed into at most $n(n-1)/2$ two-level unitary matrices. This is proven explicitly in [19] for the 3×3 unitary matrix and then using the same method it is easily proven for any unitary matrix. In article IV this result is very important as it shows that a straightforward general quantum simulator is not efficient.

A two-level unitary matrix is a matrix which acts non-trivially only on two or

fewer vector components,

$$\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a & \cdots & b & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & c & \cdots & d & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \vdots \\ \alpha_\xi \\ \vdots \\ \alpha_\eta \\ \vdots \\ \alpha_{t-1} \end{pmatrix} = \begin{pmatrix} \alpha_0 \\ \vdots \\ a\alpha_\xi + b\alpha_\eta \\ \vdots \\ c\alpha_\xi + d\alpha_\eta \\ \vdots \\ \alpha_{t-1} \end{pmatrix}, \tag{3.4}$$

with all elements on the diagonal being 1, except where a,b,c or d are on the diagonal. It is an identity matrix plus a matrix with only 4 components different from zero.

To prove that all unitary matrices can be factorized into two-level unitary matrices, we first show it for the 3×3 case,

$$U \equiv \begin{pmatrix} a & d & g \\ b & e & h \\ c & f & j \end{pmatrix}. \tag{3.5}$$

For a 3×3 matrix we need three two-level unitary matrices, U_1, U_2 and U_3, that fulfill

$$U_3 U_2 U_1 U = I. \tag{3.6}$$

If they do, then

$$\begin{aligned} U_1^\dagger U_2^\dagger U_3^\dagger U_3 U_2 U_1 U &= U_1^\dagger U_2^\dagger U_3^\dagger I \\ &= U_1^\dagger U_2^\dagger U_3^\dagger \\ &= U. \end{aligned} \tag{3.7}$$

Because the inverse of a two-level unitary matrix is also a two-level unitary matrix, we will have proved this universality for 3×3 matrices if we find U_1, U_2 and U_3. When we have done that, we use the same procedure to show it for the general case of an $n \times n$ matrix.

First we choose U_1 to be

$$\begin{pmatrix} \frac{a^*}{\sqrt{|a|^2+|b|^2}} & \frac{b^*}{\sqrt{|a|^2+|b|^2}} & 0 \\ \frac{b}{\sqrt{|a|^2+|b|^2}} & \frac{-a}{\sqrt{|a|^2+|b|^2}} & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{3.8}$$

Then we multiply U_1 with U and get

$$U_1 U = \begin{pmatrix} a' & d' & g' \\ 0 & e' & h' \\ c' & f' & j' \end{pmatrix}. \tag{3.9}$$

Here we have introduced new variables a' and so forth. What we need is that the first component of the second row is zero, as we can see from

$$\frac{b}{\sqrt{|a|^2 + |b|^2}} a + \frac{-a}{\sqrt{|a|^2 + |b|^2}} b + 0 = 0.$$

Applying a matrix U_2 of this form

$$U_2 \equiv \begin{pmatrix} \frac{a'^*}{\sqrt{|a'|^2+|c'|^2}} & 0 & \frac{c'^*}{\sqrt{|a'|^2+|c'|^2}} \\ 0 & 1 & 0 \\ \frac{c'}{\sqrt{|a'|^2+|c'|^2}} & 0 & \frac{-a'}{\sqrt{|a'|^2+|c'|^2}} \end{pmatrix} \tag{3.10}$$

to $U_1 U$, we find

$$U_2 U_1 U = \begin{pmatrix} 1 & d'' & g'' \\ 0 & e'' & h'' \\ 0 & f'' & j'' \end{pmatrix}. \tag{3.11}$$

$U_2 U_1 U$ is unitary since each of the factors are. This means that the hermitian conjugate is also the inverse and therefore the renamed factors d'' and g'' must be zero. If we now choose U_3 to be the hermitian conjugate of $U_2 U_1 U$,

$$U_3 \equiv (U_2 U_1 U)^\dagger = U^\dagger U_1^\dagger U_2^\dagger = \begin{pmatrix} 1 & 0 & 0 \\ 0 & e''^* & f''^* \\ 0 & h''^* & j''^* \end{pmatrix}, \tag{3.12}$$

then $U_3 U_2 U_1 U$ will be the identity matrix.

This proves it for the 3×3 case, for larger matrices we follow the same procedure to find lower level unitary matrices. If we have an $n \times n$ unitary matrix U, we can find a two-level unitary matrix, U_1, so that the second element in the first column of $U_1 U$ is zero. Using the technique above we then find U_2 so that the third element of the first row of $U_2 U_1 U$ is zero, and so on. When we have all zeros below the diagonal in the first column, we have used at most $n-1$ matrices. Now we have the case of eq. (3.12) where U is the product of $n-1$ two-level unitary matrices and one $(n-1)$-level unitary matrix A,

$$U' = \left[\prod_i U_i U\right] A \equiv \left[\prod_i U_i U\right] \begin{pmatrix} 1 & 0 \\ 0 & A^{(n-1)\times(n-1)} \end{pmatrix}^{n\times n}.$$

Using the same procedure we find $n-2$ two-level unitary matrices that multiplied with U' gives us an $(n-2)$-level unitary matrix. In the end we have U as a product of at most

$$k \leq \sum_{i=1}^{n}(n-1) = (n-1) + (n-2) + (n-3) + \cdots + 1 = n(n-1)/2 \quad (3.13)$$

two-level unitary matrices.

Two-level unitary matrices are factorized into CNOT and single qubit operations

Any two-level unitary matrix can then be implemented on a set of qubits by using single qubit operations and the CNOT gate as the only two qubit operation. The idea is to use gates to swap the coefficients of the two states affected so that they become the coefficients of two neighboring basis states, and then perform an appropriate controlled operation on the two qubits. When we swap back the coefficients of the target states have been changed correctly. CNOT is not the only universal two qubit gate, but the most commonly used. The number of operations needed to implement one two-level unitary matrix is one CNOT and on average N single qubit operations, where N is the number of qubits. This whole section is explained in more detail in [21]. The proof is not included here as our program only uses the two qubit gates and do not further factorize them into CNOT and single qubit gates, as different quantum computer realizations might use other elementary two qubit operations.

Single qubit operations decomposed into elementary operations

For classical computers any gate can be implemented on a set of bits using only a small subset of elementary gates. The same is true for quantum computers. It has been proven for some sets of single qubit operations, that a finite set of these elementary operations can approximate any single qubit unitary operation to arbitrary precision. In other words, one can approximate any single qubit operation one wants, using a small number of elementary operations. The Hadamard gate, the $\pi/8$ gate and the phase gate in fig. 3.2 is an example of such a set.

Universality

Benioff showed that unitary operators can be used to compute anything. Any unitary operator can now be exactly decomposed into CNOT gates and single qubit gates, and these single qubit gates can then be expressed through a small set of elementary operations. Everything that can be computed on a computer

with finite memory can be computed using CNOT and a small set of single qubit operations. This is the most important fact for quantum computing and the basis for the whole idea.

3.1.1 Complexity

For a system of N qubits, a general unitary matrix will be a $2^N \times 2^N$ matrix, which means that to implement a full unitary matrix directly would require at most $2^N(2^N - 1)/2$ two-level unitary matrices. This means that the complexity of implementing a full unitary matrix in terms of number of operations is exponential, i.e. the number of operations needed to simulate a full unitary matrix is an exponential function of the number of qubits. For quantum simulators this is important as we will come back to in section 3.2.1.

3.1.2 Exponential improvement

Some algorithms can in principle be solved much faster on quantum computers than any computer we have today. The two most famous quantum algorithms, which are exponentially faster than their classical counterparts, are Grover's search algorithm [12] and Shor's algorithm for factorizing primes [24]. Using Grover's search algorithm a quantum computer could find a number in a phone book with N entries using averagely $\sqrt{N}$ queries, while a classical computer would use averagely $N/2$ queries. An even more hailed algorithm is Shor's algorithm for finding the prime factors of an integer. The problem of finding the prime factors is a difficult task on classical computers, with an almost exponential time increase as the size of the number to factorize increases. Finding the number once you have the factors is easy however, it requires only multiplication. A quantum computer using Shor's algorithm would have as output an answer with a probability of being the right one, checking it does not take long, and averagely Shor's algorithm is much faster than the classical one. Not only is this academically very interesting, but would render one of the most used public encryption schemes in communication today useless if it suddenly became simple to factorize large numbers.

Why are quantum computers potentially more powerful than classical computers? The answers must lie in the basic properties of quantum mechanics, superposition of states and entanglement. Superposition can be used in what is called massive parallelization, for example to calculate $f(x)$ for a large set of x at the same time. Still we do not always get an improvement because of the necessity for measurements on the qubits and the collapse of the state vector. This means we can only read out one random $f(x)$ at the time, which counter the positive effects of the massive parallelization in many cases. However, clever schemes like the al-

gorithms of Shor and Grover can be constructed to make use of the superposition of states and entanglement between qubits in order to beat any classical algorithm.

3.2 Quantum simulator

The dimensionality problem is the major obstacle for numerical calculations in quantum mechanics. A state vector representing a state in a d-dimensional system has $2d$ real entries, while a matrix representing an operator on the system has $2d^2$ entries. Simulating N qubits means we need 2^{N+1} real numbers to represent the state and 2^{2N+1} real numbers to represent a matrix operating on the state. Unless we have clever algorithms in special cases, all simulation of quantum mechanical systems are exponential in complexity on classical computers.

On a quantum computer however, the complexity in storage is linear, for example we only need N qubits to simulate N spins in a spin $1/2$ chain. This is the basic idea behind the quantum simulator, which is either a full-fledged quantum computer or a specialized system performing quantum mechanical computations. The differences between classical physics and quantum physics make simulations difficult. By using quantum systems to simulate other quantum systems, some of these difficulties are removed. The complexity in time, or number of operations on the qubits, is not necessarily polynomial though, one needs clever algorithms to get the exponential speedup in time as well as in space.

The phase estimation algorithm is an algorithm for estimating the phases of the different computational basis states in a superposition. This can be used to find the eigenvalues of an Hamiltonian through the unitary time evolution operator, U. Other quantum computations that can be performed on a quantum computer includes computing relevant correlation functions of an arbitrary system with Hamiltonian H, see [25] and [26].

Finding the eigenvalues of a general Hamiltonian classically means diagonalizing a hermitian matrix. These eigenvalues can be found using the phase-estimation algorithm on a quantum computer after simulating the time evolution operator of the Hamiltonian. If we have a quantum register of qubits representing the state vector of the system being simulated, and these are in a random superposition of the eigenstates of the Hamiltonian, then the effect of the time evolution operator is to change the phases of the coefficients,

$$U|\psi\rangle = e^{-iH\Delta t}\sum_k c_k|k\rangle = \sum_k e^{-iE_k\Delta t}c_k|k\rangle. \tag{3.14}$$

We must estimate the phase of each eigenstate to find $\{E_k\}$.

To find the eigenvalues we use another register of qubits called work qubits, there are w of these and they are initialized in the state $1/\sqrt{2}(|0\rangle + |1\rangle)$. The

algorithm uses the time evolution operator w times with different timesteps, Δt. Each time the time evolution operator is applied it is used in a controlled operation with different work qubits as the control qubits. This way the phase information is stored in the work qubits. This information is then read out in the end by measuring upon the work qubits and analyzing the resulting probability distribution that we get after repeating the algorithm many times. For a thorough explanation see [21].

3.2.1 Limitations of the quantum simulator

The simulation of a system governed by a certain Hamiltonian requires implementation of the time evolution operator. One naive way of implementing the time evolution operator of a general Hamiltonian would be to create a qubit basis to describe a basis in which H is described as a matrix. The Hamiltonian is represented in a given basis $\{|i\rangle\}$ as $H = h_{ij}|i\rangle\langle j|$ by the matrix $\{h_{ij}\}$. Then one simply says that the computational basis equals the basis $\{|i\rangle\}$ in which H is expressed. Now one could find U as a product of two-level unitary matrices, express these as CNOTs and single qubit gates, and these again as the elementary operations. This approach is not good however, as the number of operations needed is an exponential function of the number of qubits as we saw in section 3.1.1. This happens because we try to implement a unitary operation that encompasses the whole computational basis, instead of addressing single qubits and pairs of qubits.

We have to find a transformation from the Hamiltonian to be simulated to the system governing the physical qubits. Simulating a Heisenberg spin chain on trapped ion qubits is not a problem since they essentially follow the same Hamiltonian, see section 3.2.1. While simulating a Bose-Einstein condensate using the same physical qubits is not that straightforward. In [20] the authors expressed clearly what is needed to efficiently simulate an Hamiltonian with a given system of qubits, namely an isomorphism between the algebras governing the two systems. That way one can create a qubit basis in such a way that the terms of the Hamiltonian themselves are transformed into operations on less than all the qubits, and the complexity in terms of number of operations will be polynomial. In [26] one can find a good introduction to the whole subject of simulating physical systems on a quantum computer.

The Heisenberg model

The Heisenberg model is a one dimensional chain of spin $1/2$ particles. The Heisenberg model is one variety of a class of spin chain models that are extremely useful in many areas of physics. In [21] the eigenvalues of the Heisenberg model

are found by simulating a quantum computer implementing the phase estimation algorithm. This was a specialized program of the same type used in article IV.

The Heisenberg model is a model of spins in a magnetic field, so that the individual spin directions give different energy to the total state. In addition the nearest neighbors influence each other. This results in an Hamiltonian with single spin terms and nearest neighbor interactions,

$$H = h\sum_k \sigma_k^z \pm J\sum_k \vec{\sigma}_k \cdot \vec{\sigma}_{k+1}. \tag{3.15}$$

The algebra for the spin $1/2$ particles is exactly the same as for the qubits, as they also are two-level quantum systems. This means the Hamiltonian is directly translated into a sum over single qubit and two qubit operations. To find the time evolution we must use a Trotter approximation, for example

$$U = e^{-iH\Delta t} = e^{-i(\sum_k H_k)\Delta t} = \prod_k e^{-iH_k\Delta t} + \mathcal{O}(\Delta t^2). \tag{3.16}$$

There are different ways to approximate U by products of exponentials of the different terms of the Hamiltonian.

Every two qubit unitary operation is represented by a 4×4 unitary matrix. This can now be decomposed into at most six two-level unitary matrices, see section 3.1. These unitary matrices can again be decomposed into the elementary operations of the actual quantum computer.

How many operations do we need to simulate the time evolution of the Heisenberg model? The sum in the Hamiltonian goes over all the qubits, from one to N, each of these terms is simulated by a given number of elementary operations. This means that the number of operations we need to simulate U for the Heisenberg model is on the order of N, we say the number of operations is $\mathcal{O}(N)$. We have an algorithm with the the number of operations linear in the number of qubits.

In the previous example we saw that the Hamiltonian was expressable as a sum over operations on one and two qubits. This means the different unitary matrices that must be decomposed into two-level unitary matrices are 4×4 matrices instead of $2^N \times 2^N$. To efficiently simulate a problem we must find a proper mapping between the algebra of the Hamiltonian to be simulated and the qubits. The straight forward method of using the whole U is not efficient.

Simulating fermions on a quantum computer

A set of standard qubits and a spin $1/2$ chain have the same algebra governing the systems, the $SU(2)$ algebra represented by the σ-matrices. The Jordan-Wigner

transformation is a transformation from fermionic annihilation and creation operators to the σ-matrices of a spin $1/2$ chain. There is an isomorphism between the two systems that means any a or $a^\dagger$ operator can be transformed into a tensor product of σ-matrices operating on a set of qubits. This was explored in [25], and in [20] an example of how to do this for single-particle fermionic operators was given, also see [26] for more information. This transformation ensure efficient, i.e. not exponential complexity, simulation of a fermionic system on a quantum computer. In article IV one can see how the annihilation and creation operators are tensor products of σ matrices on a string of qubits, this tensor product can then be implemented by a product of single- and two qubit operations. In the end we get an algorithm that takes any term in the fermionic Hamiltonian, $\prod_i a_i^\dagger \prod_j a_j$, to $\mathcal{O}(N)$ quantum gates, where N is the number of available quantum states and the number of qubits. A two-body fermionic Hamiltonian have N^4 such terms, and the total complexity is then $\mathcal{O}(N^5)$, while a complete n-body interaction requires $\mathcal{O}(N^{2n+1})$ operations.

Similar transformations must be found for other systems for them to efficiently simulated on a quantum computer. This was the main point in [25]. There is great hope for the quantum simulator, yet all is not done on the theoretical algorithmic side.

We have implemented a simulation of a quantum simulator that can take any two-body fermionic Hamiltonian and give its time evolution as a series of qubit gates. This proves that simulating two-body, three-body and higher forces is of polynomial computational complexity for fermionic systems, yet the entrance fee in terms of number of operations is high. This means that in order to get an efficient quantum simulator one would need qubits with a much higher decoherence time to operation time ratio than we see today. But when we have one that works for quite small problems, the polynomial complexity ensures that we do not need much improvement of the technology to reach the next level.

Chapter 4

Conclusion

This work has been part of the field of quantum information theory. The first part is on the foundational side with investigations of the properties of entanglement, while the second part is on the algorithmic side investigating the abilities of quantum computers to simulate quantum systems. Below is a short summary of the results from the four articles in this thesis, followed by a discussion with thoughts on how to continue this work.

Results

In article I we found an alternative proof for the sufficiency of Péres' criterion for separability in the case of two two-dimensional systems. In article I and II we developed and illustrated the use of a numerical algorithm for finding the closest separable state to a given density matrix in the Hilbert-Schmidt metric. Results from running the program show that the algorithm can be used for small systems. It is the only program available to determine whether or not a general state is entangled, even though other algorithms have been proposed. We also used this algorithm to find the boundary of the separable states in two-dimensional cross sections through the real vector space of hermitian matrices.

In article III we found a criteria for determining whether a given positive partial transpose state is an extreme point of the convex set of positive partial transpose states. We used this criteria in a numerical algorithm to randomly find such extreme points, both pure product states and bound entangled states. We found an upper bound on the rank and on the rank of the partial transpose of such extreme points.

In article IV we used the Jordan-Wigner transformation to develop a program that can take any two-body fermionic Hamiltonian and output the qubit gates a quantum computer needs to simulate the time evolution of that Hamiltonian. We then incorporated this into a quantum computer simulator, to show how this gen-

eral fermionic simulator finds the eigenvalues of the Hubbard and pairing models.

Discussion

The field of entanglement has grown immensely over the last years as an important part of quantum information theory. After the entanglement for pure bipartite states was quantified and understood, the entanglement of mixed states and of multipartite states was next in line. Entanglement for multipartite states is much more complex however, and there are many subtleties that might be exploited. We were interested in the general questions of finding a measure of entanglement for a mixed bipartite state and how one should determine whether it was separable or not. Our approach was to investigate the geometry of the separable states. One open problem was to find an algorithm that could see if a given state was separable or entangled, i.e. an algorithm that solved the separability problem. In [13] it was proven that the separability problem is NP-hard, that all algorithms that give an answer for a general matrix will have an exponential complexity after the system reaches a certain size. Our algorithm that finds the shortest distance from a density matrix to the set of separable states is not the only one proposed that can determine numerically if a state is entangled, but as far as we know the only one that has been implemented and tested. Our algorithm is of course not an effective algorithm, since that has been proven to be impossible, but it does work for small enough systems. In the future it might be important to have as effective as possible algorithms to determine if states are entangled, and this might be the basis for such an algorithm. Visualizing the boundary of the separable states in two-dimensional cross sections might be used to further understand how the boundary is generated, and so could be lead to advances in algorithms for determining separability.

The Péres criterion is necessary and sufficient for the two smallest bipartite systems, but for larger systems it is only necessary. It is still very effective for small systems though, in the sense that there is not much distance between the boundary of the positive partial transpose matrices and that of the separable matrices. Why is it so effective? The Péres set is important and our algorithm for finding its extreme points can be used to further understand the set. Today we know very little about the bound entangled extreme points of the Péres set, our work is a beginning and it might be possible to find their structure in a way similar to what we know of the extreme points of the set of separable states.

Entanglement is in any case very important and the geometrical approach can give us understanding that might lead to new algorithms or criteria that can be useful.

As we have discussed in article IV the quantum simulator has great potential for the future, but the work started by Somma, Ortiz, Gubernatis, Knill and Laflamme [25] must be continued to find transformations between systems we

wish to simulate on a quantum computer and the physical system of the qubits the computer is made up of. Our work was to take the Jordan-Wigner transformation, as they had, and to create a program that takes any two-body fermionic Hamiltonian into qubit gates for a quantum computer to implement. We showed how this could be done for higher order forces as well, and we showed that the complexity of the number of operations for each Hamiltonian term was linear in the number of qubits. This compiler might be the one to be implemented on an actual quantum simulator for simulating fermion systems. Other compilers for other systems, bosons and anyons for example [26] will have to be made in the future. The phase-estimation algorithm for quantum computers is a fundamentally new way of determining the eigenvalues of Hamiltonians. It is not an effective algorithm on a classical computer, but it is still a new method.

The field of quantum information will continue to grow over the next years and both the study of the geometry of entanglement and the simulation of quantum computers will be an important part of it.

Mathematical notation

- $\mathbf{A}$, $\mathbf{B}$, $\mathbf{E}_i$, $\mathbf{I}$, $\mathbf{J}$, ρ, σ, ξ, η: operators on an n-dimensional Hilbert space represented by $n \times n$ matrices. The matrix elements of matrix A are A_{ij}.
- λ: scalar
- $\vec{a}$, $\vec{b}$, $\vec{e_i}$: vectors in the n^2-dimensional real vector space of $n \times n$ hermitian matrices.
- $\mathbf{A} \geq 0$: matrix A is positive semidefinite, see section 2.3.
- $\mathbf{A}^T$: the transpose of the matrix $\mathbf{A}$, $(A^T)_{ij} = A_{ji}$.
- $\mathbf{A}^P$: the partial transpose of $\mathbf{A}$, $(A^P)_{ij;kl} = A_{il;kj}$.
- $\mathcal{D}$: the set of density matrices.
- $\mathcal{D}^P$: the set of partially transposed density matrices.
- $\mathcal{P}$: the set of positive partial transposed density matrices, the Péres set, $\mathcal{P} = \mathcal{D} \cap \mathcal{D}^P$. See section 2.5.
- $\mathcal{S}$: the set of separable density matrices, see section 2.2.
- $\widehat{\mathbf{1}}$: the identity matrix.

Bibliography

[1] M. Arndt, M. Aspelmeyer, H. J. Bernstein, R. Bertlmann, C. Brukner, J. P. Dowling, J. Eisert, A. Ekert, C. A. Fuchs, D. M. Greenberger, M. A. Horne, T. Jennewein, P. G. Kwiat, N. D. Mermin, J.-W. Pan, E. M. Rasel, H. Rauch, T. G. Rudolph, C. Salomon, A. V. Sergienko, J. Schmiedmayer, C. Simon, V. Vedral, P. Walther, G. Weihs, P. Zoller, and M. Zukowski, *Quantum physics from a to z,* quant-ph/0505187 (2005).

[2] M. Arndt, O. Nairz, J. Vos-Andreae, C. Keller, G. van der Zouw, and A. Zeilinger, *Wave-particle duality of c-60 molecules*, Nature **401** (1999).

[3] A. Aspect, P. Grangier, and G. Roger, *Experimental Realization of Einstein-Podolsky-Rosen-Bohm Gedankenexperiment: A New Violation of Bell's Inequalities*, Physical Review Letters **49** (1982), 91–94.

[4] J. S. Bell, *On the einstein-podolsky-rosen paradox*, Physics **1** (1964), 195–200.

[5] I Bengtsson and K. Zyczkowski, *Geometry of quantum states*, Cambridge university press, 2006.

[6] P. Benioff, *The computer as a physical system: A microscopic quantum mechanical hamiltonian model of of computers as represented by turing machines*, Journal of Statistical Physics **22** (1980), 563–590.

[7] G. Binnig, C. F. Quate, and C. Gerber, *Atomic force microscope*, Phys. Rev. Lett. **56** (1986), 930.

[8] K. Brown, R. Clark, and I. Chuang, *Limitations of quantum simulation examined by simulating a pairing hamiltonian using nuclear magnetic resonance*, Phys. rev. lett. **97** (2006), 050504.

[9] A. Einstein, B. Podolsky, and N. Rosen, *Can quantum-mechanical description of physical reality be considered complete?*, Physical Review **47** (1935), 777–780.

[10] G. H. Golub and C. F. Van Loan, *Matrix computations*, John Hopkins University Press, 1996.

[11] D. Greenberger, M. Horne, A. Shimony, and A. Zeilinger, *Bell theorem without inequalities*, American journal of physics **58** (1990).

[12] L. K. Grover, *Quantum mechanics helps in searching for a needle in a haystack*, Physical Review Letters **79** (1997), no. 2, 325.

[13] L. Gurvitz, *Classical complexity and quantum entanglement*, JOURNAL OF COMPUTER AND SYSTEM SCIENCES **69** (2004).

[14] M. Horodecki, P. Horodecki, and R. Horodecki, *Separability of mixed staes: necessary and sufficient conditions*, Physics Letters A **223** (1996), 1.

[15] P. Horodecki, *Separability criterion and inseparable mixed states with positive partial transposition*, Physics Letters A **232** (1997), 333.

[16] P. Kwiat, http://research.physics.uiuc.edu/QI/Photonics/nonlocality.html.

[17] P. Kwiat, P. Eberhard, and A. Steinbergand R. Chiao, *Proposal for a loophole-free bell inequality experiment*, Phys. rev. A **49** (1994).

[18] LHackermuller, K. Hornberger, B. Brezger, A. Zeilinger, and M. Arndt, *Decoherence of matter waves by thermal emission of radiation*, Nature **427** (2004).

[19] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information*, Cambridge University Press, Cambridge, 2000.

[20] G. Ortiz, J. E. Gubernatis, E. Knill, and R. Laflamme, *Simulating fermions on a quantum computer*, Computer physics communications **146** (2002), 302.

[21] E. Ovrum, *Quantum computing and many-body physics*, http://folk.uio.no/ovrum/thesis.pdf (2003).

[22] M. Rowe, D. Kielpinski, V. Meyer, C. Sackett, W. Itano, C. Monroe, and D. Wineland, *Experimental violation of a bell's inequality with efficient detection*, Nature **409** (2001), 791.

[23] C. Shannon, *A mathematical theory of communication*, Bell system technical journal **27** (1948).

[24] P. W. Shor, *Algorithms for quantum computation: discrete logarithms and factoring*, Proceedings of the 35th Annual Symposium on Fundamentals of Computer Science (Los Alamitos, CA), IEEE Press, 1994.

[25] R. Somma, G. Ortiz, J. E. Gubernatis, E. Knill, and R. Laflamme, *Simulating physical phenomena by quantum networks*, Phys. rev. A **65** (2002), 42323.

[26] R. Sommma, *Quantum computation, complexity, and many-body physics*, quant-ph/0512209 (2005).

[27] M. Steffen, L. Vandersypen, G. Breyta, C. Yannoni, M. Sherwood, and I. Chuang, *Experimental Realization of Shor's quantum factoring algorithm*, American Physical Society, Annual APS March Meeting, March 18 - 22, 2002 Indiana Convention Center; Indianapolis, Indiana Meeting ID: MAR02, abstract #T23.001, March 2002, p. 23001.

[28] R. Ursin, F. Tiefenbacher, T. Schmitt-Manderbach, H. Weier, T. Scheidl, M. Lindenthal, B. Blauensteiner, T. Jennewein, J. Perdigues, P. Trojek, B. Oemer, M. Fuerst, M. Meyenburg, J. Rarity, Z. Sodnik, C. Barbieri, H. Weinfurter, and A. Zeilinger, *Free-space distribution of entanglement and single photons over 144 km*, quant-ph/0607182 (2006).

[29] D. Vitali, S. Gigan, A. Ferreira, H. Böhm, P. Tombesi, A. Guerreiro, V. Vedral, A. Zeilinger, and M. Aspelmeyer, *Optomechanical entanglement between a movable mirror and a cavity field*, Phys. rev. lett **98** (2007), 030405.

[30] E. Weisstein, *Convex*, http://mathworld.wolfram.com/Convex.html.

[31] X. Yang, A. Wang, F. Xu, and J. Du, *Experimental simulation of a pairing hamiltonian on an nmr quantum computer*, Chemical physics letters **422** (2006), 20.

Part II

Papers

Paper I

Jon Magne Leinaas, Jan Myrheim and Eirik Ovrum,
Geometrical aspects of entanglement,
Phys. Rev. **A 74**, 012313 (2006)

Geometrical aspects of entanglement

Jon Magne Leinaas,[1] Jan Myrheim,[2] and Eirik Ovrum[1,3]
[1]*Department of Physics, University of Oslo, P.O. Box 1048 Blindern, 0316 Oslo, Norway*
[2]*Department of Physics, The Norwegian University of Science and Technology, 7491 Trondheim, Norway*
[3]*Centre of Mathematics for Applications, P.O. Box 1053 Blindern, 0316 Oslo, Norway*
(Received 22 May 2006; published 19 July 2006)

We study geometrical aspects of entanglement, with the Hilbert–Schmidt norm defining the metric on the set of density matrices. We focus first on the simplest case of two two-level systems and show that a "relativistic" formulation leads to a complete analysis of the question of separability. Our approach is based on Schmidt decomposition of density matrices for a composite system and nonunitary transformations to a standard form. The positivity of the density matrices is crucial for the method to work. A similar approach works to some extent in higher dimensions, but is a less powerful tool. We further present a numerical method for examining separability and illustrate the method by a numerical study of bound entanglement in a composite system of two three-level systems.

DOI: 10.1103/PhysRevA.74.012313 PACS number(s): 03.67.Mn, 02.40.Ft, 03.65.Ud

I. INTRODUCTION

Entanglement is considered to be one of the main signatures of quantum mechanics, and in recent years the study of different aspects of entanglement has received much attention. One approach has been to study the formal, geometrical characterization of entangled states as opposed to nonentangled or separable states [1–3]. In such a geometrical approach the Hilbert–Schmidt metric defines in many ways the natural metric on the space of physical states. This metric follows quite naturally from the Hilbert space norm, when the quantum description is extended from pure to mixed states, and it is a Euclidean metric on the set of density matrices. For a composite quantum system the separable states form a convex subset of the full convex set of density matrices, and one of the aims of the geometrical approach is to give a complete specification of this set and thereby of the nonseparable or entangled states.

The purpose of this paper is to examine some questions related to the geometrical description of entanglement. We focus primarily on the simplest composite systems consisting of two two-level systems (2×2 system) or two three-level systems (3×3 system), but examine also some questions relevant for higher dimensions. In the case of two two-level systems the separable states can be fully identified by use of the Peres criterion [4]. This criterion states that every separable density matrix is mapped into a positive semidefinite matrix by partial transposition, i.e., by a transposition relative to one of the subsystems. Since also Hermiticity and trace normalization is preserved under this operation, the partial transpose of a separable density matrix is a new density matrix.

A nonseparable density matrix, on the other hand, may or may not satisfy Peres' condition. This means that, in general, Peres' condition is necessary but not sufficient for separability. However, for the special case of a 2×2 system as well as for a 2×3 system the connection is stronger, and the Peres condition is both necessary and sufficient for a density matrix to be separable [5]. To identify the separable density matrices, and thereby the entangled states, is therefore in these cases relatively simple. In higher dimensions, in particular for the 3×3 system, that is not the case. Peres' condition is therefore not sufficient for separability, and the convex subset consisting of all the density matrices that satisfy this condition is larger than the set of separable matrices. This is part of the reason that the identification of the set of separable states in higher dimensions is a hard problem [6].

In the present paper we first give, in Sec. II, a brief introduction to the geometry of density matrices and separable states. As a next step, in Sec. III we focus on the geometry of the two-level system and discuss the natural extension of the Euclidean three-dimensional Hilbert–Schmidt metric to a four-dimensional indefinite Lorentz metric [7,8]. This indefinite metric is useful for the discussion of entanglement in the 2×2 system, where Lorentz transformations can be used to transform any density matrix to a diagonal standard form in a way which preserves separability and the Peres condition (Sec. IV). By using this standard form it is straightforward to demonstrate the known fact that any matrix that satisfies the Peres condition is also separable.

The transformation to the diagonal standard form is based on an extension of the Schmidt decomposition to the matrix space with indefinite metric. For general matrices such a decomposition cannot be done, but for density matrices it is possible due to the positivity condition. We show in Sec. V that the Schmidt decomposition in this form can be performed not only for the 2×2 system, but for bipartite systems of arbitrary dimensions. However, only for the 2×2 system the decomposition can be used to bring the matrices to a diagonal form where separability can be easily demonstrated. In higher dimensions the Schmidt decomposition gives only a partial simplification. This indicates that to study separability in higher dimensions one eventually has to rely on the use of numerical methods [9–13].

In Sec. VI we discuss a new numerical method to determine separability [14], and use the method to study entanglement in the 3×3 system. The method is based on a numerical estimation of the Hilbert–Schmidt distance from any chosen density matrix to the closest separable one. We focus particularly on states that are nonseparable but still satisfy the Peres condition. These are states that one usually associ-

ates with bound entanglement. A particular example of such states is the one-parameter set discussed by P. Horodecki [15] and extended by Bruss and Peres [16]. We study numerically the density matrices in the neighborhood of one particular Horodecki state and provide a map of the states in a two-dimensional subspace.

II. THE GEOMETRY OF DENSITY MATRICES

A. The convex set of density matrices

A density matrix ρ of a quantum system has the following properties:

$$\rho^\dagger = \rho \text{ Hermiticity},$$

$$\rho > 0 \text{ positivity},$$

$$\mathrm{Tr}\,\rho = 1 \text{ normalization}. \tag{1}$$

The matrices that satisfy these conditions form a convex set, and can be written in the form

$$\rho = \sum_k p_k |\psi_k\rangle\langle\psi_k| \tag{2}$$

with $|\psi_k\rangle$ as Hilbert space unit vectors and

$$p_k \geq 0, \quad \sum_k p_k = 1. \tag{3}$$

The coefficient p_k, for given k, can be interpreted as the probability that the quantum system is in the pure state $|\psi_k\rangle$. This interpretation depends, however, on the representation (2), which is by no means unique. In particular, the vectors $|\psi_k\rangle$ may be chosen to be orthonormal, then they are eigenvectors of ρ with eigenvalues p_k, and Eq. (2) is called the *spectral representation* of ρ.

The pure states, represented by the one-dimensional projections $|\psi\rangle\langle\psi|$, are the extremal points of the convex set of density matrices. That is, they generate all other density matrices, corresponding to mixed states, by convex combinations of the form (2), but cannot themselves be expressed as nontrivial convex combinations of other density matrices.

The Hermitian matrices form a real vector space with a natural scalar product, $\mathrm{Tr}(AB)$, which is bilinear in the two matrices A and B and is positive definite. From this scalar product a Euclidean metric is derived which is called the Hilbert–Schmidt (or Frobenius) metric on the matrix space,

$$ds^2 = \frac{1}{2}\,\mathrm{Tr}[(d\rho)^2]. \tag{4}$$

The scalar product between pure state density matrices $\rho_1 = |\psi\rangle\langle\psi|$ and $\rho_2 = |\phi\rangle\langle\phi|$ is

$$\mathrm{Tr}(\rho_1\rho_2) = |\langle\psi|\phi\rangle|^2. \tag{5}$$

For an infinitesimal displacement $|d\psi\rangle$ on the unit sphere in Hilbert space the displacement in the matrix space is

$$d\rho = |d\psi\rangle\langle\psi| + |\psi\rangle\langle d\psi| \tag{6}$$

and the Hilbert–Schmidt metric is

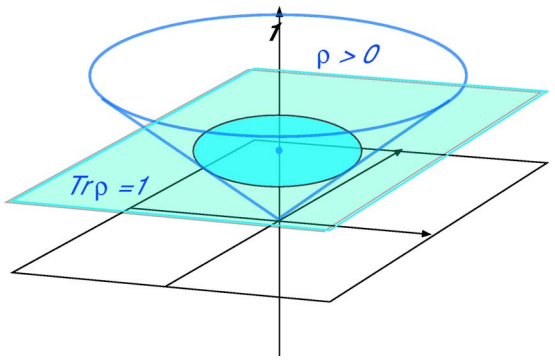


FIG. 1. (Color online) A schematic representation of the set of density matrices in the vector space of Hermitian matrices. The positive matrices form a cone about the axis defined by the unit matrix, and the normalization condition restricts the density matrices to a convex subset, here represented by the shaded circle.

$$ds^2 = \langle d\psi|d\psi\rangle - |\langle\psi|d\psi\rangle|^2, \tag{7}$$

where we have used that $\mathrm{Tr}(d\rho) = \langle\psi|d\psi\rangle + \langle d\psi|\psi\rangle = 0$. This may be interpreted as a metric on the complex projective space, called the Fubini–Study metric. It is derived from the Hilbert space metric and is a natural measure of distance between pure quantum states, in fact, ds is the infinitesimal angle between rays (one-dimensional subspaces) in Hilbert space. Since the Hilbert–Schmidt metric on density matrices is a direct extension of the Hilbert space metric, it is a natural metric for all, both pure and mixed, quantum states.

A complete set of basis vectors $\{J_a\}$, in the space of Hermitian matrices, can be chosen to satisfy the normalization condition

$$\mathrm{Tr}(J_a J_b) = \delta_{ab}. \tag{8}$$

For $n \times n$ matrices the dimension of the matrix space, and the number of basis vectors, is n^2. One basis vector, J_0, can be chosen to be proportional to the identity $\mathbb{I}$, then the other basis vectors are traceless matrices. A general density matrix can be expanded in the given basis as

$$\rho = \sum_a \xi_a J_a, \tag{9}$$

where the coefficients ξ_a are real, and the trace normalization of ρ fixes the value of ξ_0. With the chosen normalization of J_0 we have $\xi_0 = 1/\sqrt{n}$.

Due to the normalization, the density matrices are restricted to a hyperplane of dimension $n^2 - 1$, shifted in the direction of J_0 relative to a plane through the origin. The set of density matrices is further restricted by the positivity condition, so it forms a closed, convex set centered around the point $\rho_0 = J_0/\sqrt{n}$. This point corresponds to the *maximally mixed state*, which has the same probability $\langle\psi|\rho_0|\psi\rangle = 1/n$ for any pure state $|\psi\rangle$. The geometry is schematically shown in Fig. 1, where the set of density matrices is pictured as the interior of a circle. One should note that the normalization condition in a sense is trivial and can always be corrected for by a simple scale factor. In the discussion to follow we will find it sometimes convenient to give up this constraint. The

quantum states can then be viewed as rays in the matrix space, and the positivity condition restricts these to a convex sector (the cone in Fig. 1).

B. Unitary transformations

The Hermitian matrices J_a can be viewed as generators of unitary transformations,

$$U = \exp\left(i\sum_a \zeta_a J_a\right) \tag{10}$$

with real coefficients ζ_a, which act on the density matrices in the following way:

$$\rho \to \tilde{\rho} = U\rho U^\dagger. \tag{11}$$

If ρ is represented as in Eq. (2), then we see that

$$\tilde{\rho} = \sum_k p_k |\tilde{\psi}_k\rangle\langle\tilde{\psi}_k| \tag{12}$$

where $|\tilde{\psi}_k\rangle = U|\psi_k\rangle$. Thus the matrix transformation $\rho \to U\rho U^\dagger$ is induced by the vector transformation $|\psi\rangle \to U|\psi\rangle$. An immediate consequence of Eq. (12) is that the transformed density matrix $U\rho U^\dagger$ is positive.

Such unitary transformations respect both the trace and positivity conditions and therefore leave the set of density matrices invariant. Also the von Neumann entropy

$$S = -\mathrm{Tr}(\rho \ln \rho) \tag{13}$$

is unchanged, which means that the degree of mixing, as measured by the entropy, does not change.

Since the dimension of the set of density matrices grows quadratically with the Hilbert space dimension n the geometry rapidly gets difficult to visualize as n increases. However, the high degree of symmetry under unitary transformations simplifies the picture. The unitary transformations define an SU(n) subgroup of the rotations in the n^2-1 dimensional matrix space, and all density matrices can be obtained from the diagonal ones by these transformations. In this sense the geometry of the set of density matrices is determined by the geometry of the set of *diagonal* density matrices. The diagonal matrices form a convex set with a maximal set of n commuting pure states as extremal points. Geometrically, this set is a regular hyperpyramid, a *simplex*, of dimension $n-1$ with the pure states as corners. The geometrical object corresponding to the full set of density matrices is generated from this by the SU(n) transformations.

In Fig. 2 the set of diagonal density matrices is illustrated for $n=2$, 3, and 4, where in the first case the hyperpyramid has collapsed to a line segment, for $n=3$ it is an equilateral triangle and for $n=4$ it is a tetrahedron. For $n=2$, the SU(2) transformations generate from the line segment the three-dimensional *Bloch sphere* of density matrices. This case is special in the sense that the pure states form the complete surface of the set of density matrices. This does not happen in higher dimensions. In fact, the dimension of the set of pure states is $2n-2$, the dimension of SU(n)/[U(1)$\times$SU($n-1$)], because one given pure state has a U(1)$\times$SU($n-1$) invariance group. This dimension grows linearly with n, while the dimension of the surface, n^2-2, grows quadratically.

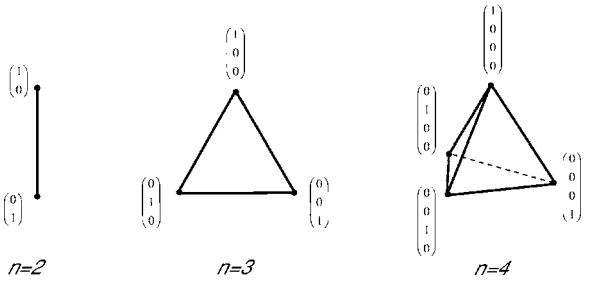


FIG. 2. Geometric representation of the diagonal density matrices for the three cases of Hilbert space dimension 2, 3, and 4. For general dimension n they define a hyperpyramid of dimension $n-1$.

The faces of the hyperpyramid of dimension $n-1$ are hyperpyramids of dimension $n-2$, corresponding to density matrices of the subspace orthogonal to the pure state of the opposite corner. Similarly, the hyperpyramid of dimension $n-2$ is bounded by hyperpyramids of dimension $n-3$, etc. This hierarchy is present also in the full set of density matrices, generated from the diagonal ones by SU(n) transformations. Thus, to each extremal point (pure state) the boundary surface opposite to it is a flat face corresponding to the set of density matrices of one lower dimension. In this way the boundary surface of the set of density matrices contains a hierarchy of sets of density matrices of lower dimensions.

The boundary of the set of density matrices is characterized by at least one of the eigenvalues of the density matrices being zero, since outside the boundary the positivity condition is broken. This means that at the boundary the density matrices satisfy the condition $\det\rho=0$, which is an algebraic equation for the coordinates of the boundary points. When n is not too large the equation can be solved numerically. This has been done in Fig. 6 where a two-dimensional section of the set of density matrices is shown. One should note that there will be solutions to the equation $\det\rho=0$ also outside the set of density matrices. The boundary of the set of density matrices can be identified as the *closed* surface, defined by $\det\rho=0$, that encloses the maximally mixed state and is closest to this point.

C. More general transformations

We shall later make use of the complex extension of the transformations (10) by allowing ζ_a to be complex. This means that the transformation group is extended from SU(n) to SL($n,\mathbb{C}$) (the normalization condition $\det U=1$, or $\zeta_0=0$, is trivial). Transformations of the form $\tilde{\rho}=V\rho V^\dagger$ do not respect the trace condition $\mathrm{Tr}\,\rho=1$ if V is nonunitary, but they do respect the positivity condition because they are still vector transformations of the form (12), with $|\tilde{\psi}_k\rangle = V|\psi_k\rangle$. This means that they leave the sector of non-normalized density matrices invariant. They no longer keep the entropy unchanged, however. Thus the larger group SL($n,\mathbb{C}$) connects a larger set of density matrices than the restricted group SU(n).

One further generalization is possible. In fact, even if we

allow V to be *antilinear*, the transformation $\tilde{\rho}=V\rho V^\dagger$ still preserves positivity because Eq. (12) still holds with $|\tilde{\psi}_k\rangle = V|\psi_k\rangle$. This point needs some elaboration.

An operator V is antilinear if

$$V(a|\psi\rangle + b|\phi\rangle) = a^* V|\psi\rangle + b^* V|\phi\rangle \tag{14}$$

for any vectors $|\psi\rangle$, $|\phi\rangle$ and complex numbers a, b. Let $\{|i\rangle\}$ be a set of orthonormal basis vectors, let $\psi_i=\langle i|\psi\rangle$, and write $|V\psi\rangle$ for the vector $V|\psi\rangle$. Then

$$|V\psi\rangle = V\sum_i \psi_i|i\rangle = \sum_{i,j} V_{ji}\psi_i^*|j\rangle \tag{15}$$

with $V_{ji}=\langle j|V|i\rangle$. The Hermitian conjugate $V^\dagger$ is defined in a basis independent way by the identity

$$\langle\psi|V^\dagger\phi\rangle \equiv \langle\phi|V\psi\rangle = \sum_{i,j}\phi_j^* V_{ji}\psi_i^* \tag{16}$$

or equivalently,

$$|V^\dagger\phi\rangle = V^\dagger|\phi\rangle = \sum_{i,j}\phi_j^* V_{ji}|i\rangle. \tag{17}$$

By definition, V is *antiunitary* if $V^\dagger=V^{-1}$.

Familiar relations valid for linear operators are not always valid for antilinear operators. For example, when V is antilinear and $|V\psi\rangle=V|\psi\rangle$, it is no longer true that $\langle V\psi|=\langle\psi|V^\dagger$. This relation cannot hold, simply because $\langle V\psi|$ is a linear functional on the Hilbert space, whereas $\langle\psi|V^\dagger$ is an antilinear functional. What is nevertheless true is that $V|\psi\rangle\langle\psi|V^\dagger = |V\psi\rangle\langle V\psi|$. In fact, both of these operators are linear, and they act on the vector $|\phi\rangle=\Sigma_j\phi_j|j\rangle$ as follows:

$$\begin{aligned} V|\psi\rangle\langle\psi|V^\dagger|\phi\rangle &= V\left(\sum_{i,j}\phi_j^* V_{ji}\psi_i^*|\psi\rangle\right) \\ &= \sum_{i,j}\psi_i V_{ji}^*\phi_j V|\psi\rangle \\ &= |V\psi\rangle\langle V\psi|\phi\rangle. \end{aligned} \tag{18}$$

As a consequence of this identity the form (12) is valid for the antiunitary transformations, and the positivity is thus preserved.

The transposition of matrices, $\rho\to\rho^T$, obviously preserves positivity, since it preserves the set of eigenvalues. This is not an SU(n) transformation of the form (11), as one can easily check. However, transposition of a Hermitian matrix is the same as complex conjugation of the matrix, and if we introduce the complex conjugation operator K, which is antilinear and antiunitary, we may write

$$\rho^T = K\rho K^\dagger. \tag{19}$$

Note that transposition is a basis dependent operation. The complex conjugation operator K is also basis dependent, it is defined to be antilinear and to leave the basis vectors invariant, $K|i\rangle=|i\rangle$. We see that $K^\dagger=K=K^{-1}$.

One may ask the general question, which are the transformations that preserve positivity of Hermitian matrices. If we consider an *invertible* linear transformation on the real vector space of matrices, then it has to be a one-to-one mapping of the extremal points of the convex set of positive matrices onto the extremal points. In other words, it is a one-to-one mapping of one-dimensional projections onto one-dimensional projections. In yet other words, it is an invertible vector transformation $|\psi\rangle\to V|\psi\rangle$, defined up to a phase factor, or more generally an arbitrary nonzero complex factor, for each pure state $|\psi\rangle$. One can show that these complex factors can be chosen in such a way that V becomes either linear or antilinear, and that the matrix transformation is $\rho\to V\rho V^\dagger$. However, we will not go into details about this point here.

D. Geometry and separability

We consider next a composite system with two subsystems A and B, of dimensions n_A and n_B, respectively. By definition, the *separable states* of the system are described by density matrices that can be written in the form

$$\rho = \sum_k p_k \rho_k^A \otimes \rho_k^B, \tag{20}$$

where ρ_k^A and ρ_k^B are density matrices of the two subsystems and p_k is a probability distribution over the set of product density matrices labeled by k. The separable states form a convex subset of the set of all density matrices of the composite system, with the *pure product states* $|\psi\rangle\langle\psi|$, where $|\psi\rangle=|\phi\rangle\otimes|\chi\rangle$, as extremal points. Our interest is to study the geometry of this set, and thereby the geometry of the set of entangled states, defined as the complement of the set of separable states within the full set of density matrices.

The Peres criterion [4] gives a necessary condition for a density matrix to be separable. Let us introduce orthonormal basis vectors $|i\rangle_A$ in $\mathcal{H}_A$ and $|j\rangle_B$ in $\mathcal{H}_B$, as well as the product vectors

$$|ij\rangle = |i\rangle_A \otimes |j\rangle_B. \tag{21}$$

We write the matrix elements of the density matrix ρ as

$$\rho_{ij;kl} = \langle ij|\rho|kl\rangle. \tag{22}$$

The partial transposition with respect to the B system is defined as the transformation

$$\rho\to\rho^P: \quad \rho^P_{ij;kl} \equiv \rho_{il;kj}. \tag{23}$$

This operation preserves the trace, but not necessarily the positivity of ρ. However, for separable states one can see from the expansion (20) that it preserves positivity, because it is just a transposition of the density matrices ρ_k^B of the subsystem B.

Thus the Peres criterion states that preservation of positivity under a partial transposition is a necessary condition for a density matrix ρ to be separable. Conversely, if the partial transpose ρ^P is *not* positive definite, it follows that ρ is nonseparable or entangled. The opposite is not true: if ρ^P is positive, the density matrix ρ is not necessarily separable.

It should be emphasized that the Peres condition, i.e., positivity of both ρ and ρ^P, is independent of the choice of basis vectors $|i\rangle_A$ and $|j\rangle_B$. In fact, a change of basis may result in another definition of the partial transpose ρ^P, which

differs from the first one by a unitary transformation, but this does not change the eigenvalue spectrum of ρ^P. The condition is also the same if transposition is defined with respect to subsystem A rather than B. This is obvious, since partial transposition with respect to the A subsystem is just the combined transformation $\rho \to (\rho^T)^P$.

Let us consider the Peres condition from a geometrical point of view. We first consider the transposition of matrices, $\rho \to \rho^T$, and note that it leaves the Hilbert–Schmidt metric invariant. Being its own inverse, transposition is an inversion in the space of density matrices, or a rotation if the number of inverted directions, $n(n-1)/2$ in an $n \times n$ Hermitian matrix, is even. Since ρ and ρ^T have the same set of eigenvalues, transposition preserves positivity and maps the set of density matrices onto itself. Thus the set of density matrices $\mathcal{D}$ is invariant under transposition as well as under unitary transformations.

Similarly, a partial transposition $\rho \to \rho^P$ preserves the metric and therefore also corresponds to an inversion or rotation in the space of matrices. On the other hand, it does not preserve the eigenvalues and therefore in general does not preserve positivity. This means that the set of density matrices, $\mathcal{D}$, is not invariant under partial transposition, but is mapped into an inverted or rotated copy $\mathcal{D}^P$. These two sets will partly overlap, in particular, they will overlap in a neighborhood around the maximally mixed state, since this particular state is invariant under partial transposition. We note that, even though partial transposition is basis dependent, the set of transposed matrices $\mathcal{D}^P$ does not depend on the chosen basis. Nor does it depend on whether partial transposition is defined with respect to subsystem A or B.

To sum up the situation, we consider the following three convex sets. $\mathcal{D}$ is the full set of density matrices of the composite system, while $\mathcal{P}=\mathcal{D}\cap\mathcal{D}^P$ is the subset of density matrices that satisfy the Peres condition, and $\mathcal{S}$ is the set of separable density matrices. In general we then have the following inclusions:

$$\mathcal{S} \subset \mathcal{P} \subset \mathcal{D}. \tag{24}$$

The Peres criterion is useful thanks to the remarkable fact that partial transposition does not preserve positivity. This fact is indeed remarkable for the following reason. We have seen that any linear or antilinear vector transformation $|\psi\rangle \to V|\psi\rangle$ will preserve the positivity of Hermitian matrices by the transformation $\rho \to V\rho V^\dagger$. It would seem that $V=\mathbb{I}_A \otimes K_B$, a complex conjugation on subsystem B, would be a vector transformation such that $\rho^P = V\rho V^\dagger$, and hence that partial transposition would preserve positivity. What is wrong with this argument is that there exists no such operator as $\mathbb{I}_A \otimes K_B$. To see why, choose a complex number c with $|c|=1$, and consider the transformation of a product vector $|\psi\rangle = |\phi\rangle \otimes |\chi\rangle$,

$$\begin{aligned}(\mathbb{I}_A \otimes K_B)|\psi\rangle &= (\mathbb{I}_A \otimes K_B)[(c|\phi\rangle) \otimes (c^*|\chi\rangle)] \\ &= c^2[|\phi\rangle \otimes (K|\chi\rangle)].\end{aligned} \tag{25}$$

The arbitrary phase factor c^2 invalidates the attempted definition.

The boundary of the congruent (or reflected) image $\mathcal{D}^P$ of $\mathcal{D}$ is determined by the condition $\det \rho^P = 0$ in the same way as the boundary of the set of density matrices $\mathcal{D}$ is determined by $\det \rho = 0$. As a consequence, to determine whether a density matrix ρ belongs to the set $\mathcal{P}$ is not a hard problem. One simply checks whether the determinants of $\hat{\rho}$ and $\hat{\rho}^P$ are both positive for every $\hat{\rho}$ on the line segment between ρ and the maximally mixed state ρ_0. However, to check whether a density matrix is separable and thus belongs to the subset $\mathcal{S}$ is in general not easy, even though the definition (20) of separability has a simple form. The exceptional cases are the systems of Hilbert space dimensions 2×2, 2×3, or 3×2, where $\mathcal{S}=\mathcal{P}$.

E. Schmidt decomposition and transformation to a standard form

A general density matrix of the composite system can be expanded as

$$\rho = \sum_{a=0}^{n_A^2-1} \sum_{b=0}^{n_B^2-1} \xi_{ab} J_a^A \otimes J_b^B, \tag{26}$$

where the coefficients ξ_{ab} are real, and J_a^A and J_b^B are orthonormal basis vectors of the two subsystems. We may use our convention that $J_0^A = \mathbb{I}_A/\sqrt{n_A}$ and $J_0^B = \mathbb{I}_B/\sqrt{n_B}$, and that J_k^A and J_k^B with $k>0$ are generators of $\mathrm{SU}(n_A)$ and $\mathrm{SU}(n_B)$.

A Schmidt decomposition is a diagonalization of the above expansion. By a suitable choice of basis vectors $\hat{J}_a^A$ and $\hat{J}_b^B$, depending on ρ, we may always write

$$\rho = \sum_{a=0}^{n_A^2-1} \hat{\xi}_a \hat{J}_a^A \otimes \hat{J}_a^B \tag{27}$$

assuming that $n_A \le n_B$. There exist many different such diagonal representations of a given ρ, in fact it is possible to impose various extra conditions on the new basis vectors. It is usual to impose an orthonormality condition, that the new basis vectors should be orthonormal with respect to some positive definite scalar product. Then the Schmidt decomposition of ρ is the same as the singular value decomposition [17] of the $n_A^2 \times n_B^2$ matrix ξ_{ab}. Below, we will introduce a Schmidt decomposition based on other types of extra conditions.

The usefulness of the representation (27) is limited by the fact that we expand in basis vectors depending on ρ. However, we may make a transformation of the form

$$\rho \to \tilde{\rho} = V\rho V^\dagger \tag{28}$$

where $V=V_A \otimes V_B$ is composed of transformations $V_A \in \mathrm{SL}(n_A, \mathbb{C})$ and $V_B \in \mathrm{SL}(n_B, \mathbb{C})$ that act independently on the two subsystems and transform the basis vectors $\hat{J}_a^A$ and $\hat{J}_b^B$ into $V_A \hat{J}_a^A V_A^\dagger$ and $V_B \hat{J}_b^B V_B^\dagger$. A transformation of this form obviously preserves the set $\mathcal{S}$ of separable states, since a sum of the form (20) is transformed into a sum of the same form. It also preserves the set $\mathcal{P}$ of density matrices satisfying the Peres condition. In fact, it preserves the positivity not only of ρ, but also of the partial transpose ρ^P, since

$$(\tilde{\rho})^P = (V_A \otimes V_B^*)\rho^P(V_A \otimes V_B^*)^\dagger. \tag{29}$$

Here V_B^* is the complex conjugate of V_B. What is not preserved by the transformation is the trace, but this can easily be corrected by introducing a normalization factor. Such transformations have been considered, e.g., by Cen *et al.* [18] and by Osterloh and Siewert [19] in the case of pure states.

As we will later show, it is possible to choose the transformation $V=V_A\otimes V_B$ in such a way that the transformed and normalized density matrix $\tilde{\rho}$ can be brought into the special form

$$\tilde{\rho} = \frac{1}{n_A n_B}\left(\mathbb{I} + \sum_{k=1}^{n_A^2-1} \tilde{\xi}_k \tilde{J}_k^A \otimes \tilde{J}_k^B\right), \tag{30}$$

with $\tilde{J}_k^A$ and $\tilde{J}_k^B$ as new sets of traceless orthonormal basis vectors. We have some freedom in choosing V_A and V_B because the form (30) is preserved by unitary transformations of the form $U=U_A\otimes U_B$.

In the case $n_A=n_B=2$ we may choose V_A and V_B so that $\tilde{J}_k^A$ and $\tilde{J}_k^B$ are fixed sets of basis vectors, independent of the density matrix ρ. In particular, we may use the standard Pauli matrices as basis vectors. In this way we define a special form of the density matrices $\tilde{\rho}$, which we refer to as the standard form. Any density matrix can be brought into this form by a transformation that preserves separability and the Peres condition. All matrices of the resulting standard form commute and can be simultaneously diagonalized. This makes it easy to prove the equality $\mathcal{S}=\mathcal{P}$, and thereby solve the separability problem. Although this result is well-known, the proof given here is simpler than the original proof.

The decomposition (30) is generally valid, but when either n_A, n_B, or both are larger than 2, it is impossible to choose both $\tilde{J}_k^A$ and $\tilde{J}_k^B$ to be independent of ρ. Simply by counting the number of parameters one easily demonstrates that this cannot be done in higher dimensions. Thus the product transformations $V_A\otimes V_B$ are specified by $2n_A^2+2n_B^2-4$ parameters, while the number of parameters in Eq. (30) is n_A^2-1, when the generators $\tilde{J}_k^A$ and $\tilde{J}_k^B$ are fixed. This gives a total number of parameters $3n_A^2+2n_B^2-5$, when $n_A\leq n_B$, compared to the number of parameters of the general density matrix, which is $n_A^2n_B^2-1$. Only for $n_A=n_B=2$ do these numbers match.

The mismatch in the number of parameters shows that the independent transformations V_A and V_B performed on the two subsystems are less efficient in simplifying the form of the density matrices in higher dimensions. In particular, it is impossible to transform all density matrices to a standard form of commuting matrices. Thus the question of separability is no longer trivially solved. Nevertheless, we consider the Schmidt decomposition to be interesting and important, if only because the number of dimensions in the problem is reduced. We expect this to be useful, even if, at the end, separability can only be determined by numerical methods.

III. THE TWO-LEVEL SYSTEM

The density matrices of a two-level system describe the states of a qubit and represent a simple, but important, special case. It is well-known that the normalized density matrices, expressed in terms of the Pauli matrices $\boldsymbol{\sigma}=(\sigma_1,\sigma_2,\sigma_3)$ as

$$\rho = \frac{1}{2}(\mathbb{I} + \mathbf{r}\cdot\boldsymbol{\sigma}), \tag{31}$$

can geometrically be pictured as the interior of a three-dimensional unit sphere, the *Bloch sphere*, with each point identified by a vector $\mathbf{r}$. The two-level case is special in that the pure states form the complete surface $|\mathbf{r}|=1$ of the set of density matrices. The diagonal 2×2 matrices, in any chosen basis, correspond to a line segment through the origin with the two pure basis states as end points.

The two-level system is special also in the sense that the Euclidean metric of the three dimensional space of density matrices can be extended in a natural way to an indefinite metric in four dimensions. The extension is analogous to the extension from the three-dimensional Euclidean metric to the four-dimensional Lorentz metric in special relativity. Since it is useful for the discussion of entanglement in the two qubit system, we shall briefly discuss it here.

We write the density matrix in relativistic notation as

$$\rho = \frac{1}{2}x^\mu\sigma_\mu, \tag{32}$$

where σ_0 is the identity matrix $\mathbb{I}$. The trace normalization condition is that $\mathrm{Tr}\,\rho=x^0=1$. We may relax this condition and retain only the positivity condition on ρ, which means that x^0 is positive and dominates the vector part $\mathbf{r}$ of the four-vector x^μ, as expressed by the covariant condition

$$4\det\rho = (x^0)^2 - |\mathbf{r}|^2 = x^\mu x_\mu = g_{\mu\nu}x^\mu x^\nu \geq 0. \tag{33}$$

In other words, the four-vector x^μ is restricted by the positivity condition to be either a timelike vector inside the forward light cone, or a lightlike vector on the forward light cone. The lightlike vectors correspond to pure states, and the timelike vectors to mixed states. As already discussed, all points on a given line through the origin represent the same normalized density matrix (see Fig. 3).

Positivity is conserved by matrix transformations of the form

$$\rho \to \tilde{\rho} = V\rho V^\dagger = \frac{1}{2}L_\mu^\nu x^\mu\sigma_\nu. \tag{34}$$

If we restrict V to be linear (not antilinear), invertible, and normalized by $\det V=1$, then it belongs to the group $\mathrm{SL}(2,\mathbb{C})$, and L is a continuous Lorentz transformation (continuous in the sense that it can be obtained as a product of small transformations). Thus preservation of positivity by the $\mathrm{SL}(2,\mathbb{C})$ transformations corresponds to preservation of the forward light cone by the continuous Lorentz transformations.

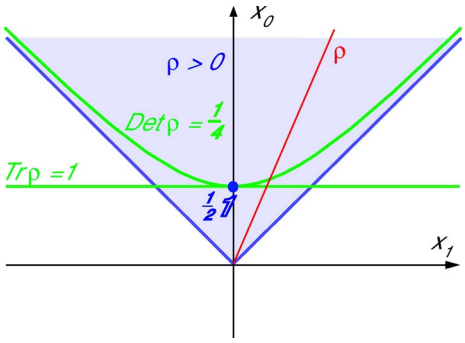


FIG. 3. (Color online) A relativistic view of the density matrices of a two-level system. The positive Hermitian matrices are represented by the forward light cone and a density matrix ρ by a timelike or lightlike ray (red line). The standard normalization $\mathrm{Tr}\,\rho = x^0 = 1$ breaks relativistic invariance (green horizontal line), but may be replaced by the relativistically invariant normalization $\det \rho = x^\mu x_\mu/4 = 1/4$ (green hyperbola).

In order to compare the indefinite Lorentz metric to the Euclidean scalar product $\mathrm{Tr}(AB)$ introduced earlier, we introduce the operation of space inversion,

$$\sigma_\mu = (\mathbb{I}, \boldsymbol{\sigma}) \to \bar{\sigma}_\mu \equiv (\mathbb{I}, -\boldsymbol{\sigma}). \tag{35}$$

It is obtained as a combination of matrix transposition, or equivalently complex conjugation, which for the standard Pauli matrices inverts the sign of $\sigma_y = \sigma_2$, and a rotation of π about the y axis. Thus for a general Hermitian 2×2 matrix A it acts as

$$A \to \bar{A} = RA^T R^\dagger \tag{36}$$

where

$$R = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}. \tag{37}$$

We may also write $A^T = KAK^\dagger$, where $K = K^\dagger$ is complex conjugation.

The Lorentz metric is now expressed by

$$\mathrm{Tr}(\bar{\sigma}_\mu \sigma_\nu) = 2g_{\mu\nu} \tag{38}$$

and the Lorentz invariant scalar product between two Hermitian matrices $A = a^\mu \sigma_\mu$ and $B = b^\mu \sigma_\mu$ is

$$\mathrm{Tr}(\bar{A}B) = 2a^\mu b_\mu. \tag{39}$$

The invariance of the scalar product (38) can be seen directly from the transformation properties under $\mathrm{SL}(2,\mathbb{C})$ transformations,

$$A \to A' = VAV^\dagger \Rightarrow \bar{A} \to \bar{A}' = V^{\dagger -1} \bar{A} V^{-1}. \tag{40}$$

Thus A and $\bar{A}$ transform under contragredient representations of $\mathrm{SL}(2,\mathbb{C})$.

The Lorentz transformed Pauli matrices

$$\tilde{\sigma}_\mu = V\sigma_\mu V^\dagger = L_\mu^\nu \sigma_\nu, \quad V \in \mathrm{SL}(2,\mathbb{C}) \tag{41}$$

satisfy the same metric condition (38) as the standard Pauli matrices σ_μ. Conversely, any set of matrices $\tilde{\sigma}_\mu$ satisfying relations of the form (38) are related to the Pauli matrices by a Lorentz transformation, which need not, however, be restricted to the continuous part of the Lorentz group, but may include space inversion or time reversal, or both.

It is clear from the relativistic representation that any density matrix can be reached from the maximally mixed state $\rho_0 = \mathbb{I}/2$ by a transformation corresponding to a boost. The Lorentz boosts generate from ρ_0 a three-dimensional hyperbolic surface (a "mass shell") where $x^\mu x_\mu = 1$. This surface will intersect any timelike line once and only once. Thus any mixed state is obtained from ρ_0 by a unique boost. However, the pure states, corresponding to lightlike vectors, can only be reached asymptotically, when the boost velocity approaches the speed of light, here set equal to 1. The form of the $\mathrm{SL}(2,\mathbb{C})$ transformation corresponding to a pure boost is

$$V(\boldsymbol{\xi}) = \exp\left(\frac{1}{2}\boldsymbol{\xi} \cdot \boldsymbol{\sigma}\right), \tag{42}$$

where the three-dimensional real vector $\boldsymbol{\xi}$ is the boost parameter, called rapidity. Since the boost matrices are Hermitian, a density matrix defined by a boost of the maximally mixed state will have the form

$$\rho = N(\xi)V(\boldsymbol{\xi})^2 = N(\xi)(\cosh \xi \mathbb{I} + \sinh \xi \hat{\boldsymbol{\xi}} \cdot \boldsymbol{\sigma}), \quad \hat{\boldsymbol{\xi}} = \frac{\boldsymbol{\xi}}{\xi}, \tag{43}$$

where $N(\xi)$ is a normalization factor determined by the trace normalization condition. The normalized density matrix is

$$\rho = \frac{1}{2}(\mathbb{I} + \tanh \xi \hat{\boldsymbol{\xi}} \cdot \boldsymbol{\sigma}). \tag{44}$$

Thus the boost parameter $\boldsymbol{\xi}$ gives a representation which is an alternative to the Bloch sphere representation. The relation between the parameters $\mathbf{r}$ and $\boldsymbol{\xi}$ is that

$$\mathbf{r} = \tanh \xi \hat{\boldsymbol{\xi}}, \tag{45}$$

which means that $\mathbf{r}$ can be identified as the velocity of the boost, in the relativistic picture, with $|\mathbf{r}| = 1$ corresponding to the speed of light. We note that the positivity condition gives no restriction on $\boldsymbol{\xi}$, and the extremal points, i.e., the pure states, are points at infinity in the $\boldsymbol{\xi}$ variable.

IV. ENTANGLEMENT IN THE 2×2 SYSTEM

We consider now in some detail entanglement between two two-level systems. We will show that with the use of nonunitary transformations the density matrices can be written in a standardized Schmidt decomposed form. In this form the question of separability is easily determined and the equality of the two sets $\mathcal{P}$ and $\mathcal{S}$ is readily demonstrated.

A. Schmidt decomposition by Lorentz transformations

We consider transformations $V = V_A \otimes V_B$ composed of $\mathrm{SL}(2,\mathbb{C})$ transformations acting independently on the two subsystems, and therefore respecting the product form (20) of the separable matrices. We will show that by means of

such transformations any density matrix of the composite system can be transformed to the form

$$\tilde{\rho} = \frac{1}{4}\left(\mathbb{I} + \sum_{k=1}^{3} d_k \sigma_k \otimes \sigma_k\right), \tag{46}$$

which we refer to as the standard form. Note that the real coefficients d_k must be allowed to take both positive and negative values.

We start by writing a general density matrix in the form

$$\rho = c^{\mu\nu}\sigma_\mu \otimes \sigma_\nu. \tag{47}$$

The transformation $V = V_A \otimes V_B$ produces independent Lorentz transformation L_A and L_B on the two subsystems,

$$\rho \to \tilde{\rho} = V\rho V^\dagger = \tilde{c}^{\mu\nu}\sigma_\mu \otimes \sigma_\nu \tag{48}$$

with

$$\tilde{c}^{\mu\nu} = L^\mu_{A\rho} L^\nu_{B\sigma} c^{\rho\sigma}. \tag{49}$$

The Schmidt decomposition consists in choosing L_A and L_B in such a way that $\tilde{c}^{\mu\nu}$ becomes diagonal. We will show that this is always possible when ρ is strictly positive.

Note that the standard Schmidt decomposition, also called the singular value decomposition, involves a compact group of rotations, leaving invariant a positive definite scalar product. The present case is different because it involves a noncompact group of Lorentz-transformations, leaving invariant an indefinite scalar product.

The positivity condition on ρ plays an essential part in the proof. It states that

$$\langle\psi|\rho|\psi\rangle \geq 0, \tag{50}$$

where $|\psi\rangle$ is an arbitrary state vector. Let us consider a density matrix ρ which is *strictly* positive so that Eq. (50) is satisfied with $>$ and not only with $\geq$, and let us restrict $|\psi\rangle$ to be of product form, $|\psi\rangle = |\phi\rangle \otimes |\chi\rangle$. With ρ expressed by Eq. (47), the positivity condition then implies

$$c^{\mu\nu} m_\mu n_\nu > 0 \tag{51}$$

with

$$m_\mu = \langle\phi|\sigma_\mu|\phi\rangle, \quad n_\nu = \langle\chi|\sigma_\nu|\chi\rangle. \tag{52}$$

These two four-vectors are on the forward light cone, in fact, it is easy to show that

$$m_\mu m^\mu = n_\mu n^\mu = 0, \quad m_0 = n_0 = 1. \tag{53}$$

We note that by varying the state vectors $|\phi\rangle$ and $|\chi\rangle$ all directions on the light cone can be reached. The inequality (51) holds for forward timelike vectors as well, because any such vector may be written as a linear combination of two forward lightlike vectors, with positive coefficients. We may actually write a stronger inequality

$$c^{\mu\nu} m_\mu n_\nu \geq C > 0 \tag{54}$$

valid for all timelike or lightlike vectors m and n with $m_0 = n_0 = 1$. In fact, this inequality holds because the set of such pairs of four-vectors (m,n) is compact.

Now define the function

$$f(m,n) = \frac{c^{\mu\nu} m_\mu n_\nu}{\sqrt{m^\rho m_\rho n^\sigma n_\sigma}}. \tag{55}$$

It is constant for m and n lying on two fixed, one-dimensional rays inside the forward light cone. It goes to infinity when either m or n becomes lightlike, because

$$f(m,n) \geq \frac{C m_0 n_0}{\sqrt{m^\rho m_\rho n^\sigma n_\sigma}}. \tag{56}$$

Using again a compactness argument, we conclude that there exist four-vectors $\bar{m}$ and $\bar{n}$ such that $f(\bar{m},\bar{n})$ is minimal. We may now choose the Lorentz transformations L_A and L_B such that

$$L^0_{A\mu} = \bar{m}_\mu, \quad L^0_{B\mu} = \bar{n}_\mu \tag{57}$$

assuming the normalization conditions $\bar{m}^\mu \bar{m}_\mu = \bar{n}^\mu \bar{n}_\mu = 1$. This defines L_A and L_B uniquely up to arbitrary three-dimensional rotations. Define

$$\tilde{f}(m,n) = \frac{\tilde{c}^{\mu\nu} m_\mu n_\nu}{\sqrt{m^\rho m_\rho n^\sigma n_\sigma}}. \tag{58}$$

Since $\tilde{f}(m,n) = f(\tilde{m},\tilde{n})$, with $\tilde{m}_\mu = L^\rho_{A\mu} m_\rho$ and $\tilde{n}_\mu = L^\rho_{B\mu} n_\rho$, it follows that $\tilde{f}(m,n)$ has a minimum at $m = n = (1,0,0,0)$. The condition for an extremum at $m = n = (1,0,0,0)$ is that $\tilde{c}^{0k} = \tilde{c}^{k0} = 0$ for $k = 1,2,3$, so that

$$\tilde{\rho} = \tilde{c}^{00}\mathbb{I} + \sum_{k=1}^{3}\sum_{l=1}^{3} \tilde{c}^{kl}\sigma_k \otimes \sigma_l. \tag{59}$$

The coefficient $\tilde{c}^{00}$ is the minimum value of $f(m,n)$, and hence positive.

The last term of Eq. (59) can be diagonalized by a standard Schmidt decomposition, and by a further normalization $\tilde{\rho}$ can be brought into the form (30). Finally, a unitary transformation $\tilde{\rho} \to U\tilde{\rho}U^\dagger$ of the product form $U = U_A \otimes U_B$ may be performed, where the unitary matrices U_A and U_B may be chosen so that $U_A \tilde{J}^A_k U^\dagger_A = \pm\sigma_k$ and $U_B \tilde{J}^B_k U^\dagger_B = \pm\sigma_k$. This is always possible because SU(2) transformations generate the full three-dimensional rotation group, excluding inversions. In this way we obtain the standard form (46).

Note that the standard form (46) of a given density matrix ρ is not unique because there exists a discrete subgroup of 24 unitary transformations that transform one matrix of this form into other matrices of the same form. This group includes all permutations of the three basis vectors $\sigma_k \otimes \sigma_k$, as well as simultaneous reversals of any two of the basis vectors. It is the full symmetry group of a regular tetrahedron. If we want to make the standard form unique we may, for example, impose the conditions $d_1 \geq d_2 \geq |d_3|$, allowing both positive and negative values of d_3.

B. Density matrices in the standard form

The density matrices of the standard form (46) define a convex subset of lower dimension than the full set of density matrices. It is a three-dimensional section of the 15-dimensional set of density matrices, consisting of commuting

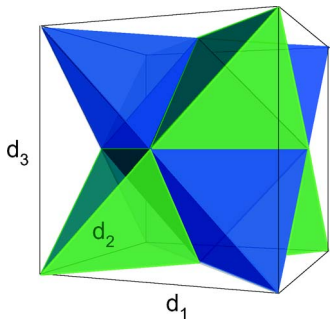


FIG. 4. (Color online) Geometric representation of the diagonal matrices spanned by orthogonal Bell states of the 2×2 system. The density matrices form a tetrahedron (green, its edges are diagonals of the faces of a cube) while the matrices obtained from this by a partial transposition define a mirror image (blue, its edges are the opposite diagonals). The separable states are defined by the intersection between the two tetrahedra and form an octahedron (in the center).

(simultaneously diagonalizable) matrices. The eigenvalues, as functions of the parameters d_k of Eq. (46), are

$$\lambda_{1,2} = \frac{1}{4}[1 \pm (d_1 - d_2) + d_3], \quad \lambda_{3,4} = \frac{1}{4}[1 \pm (d_1 + d_2) - d_3]. \tag{60}$$

The pure states (with eigenvalues $1,0,0,0$) that are the extremal points of the convex set of commuting matrices are specified by the conditions

$$|d_1| = |d_2| = |d_3| = 1, \quad d_1 d_2 d_3 = -1. \tag{61}$$

There are four such states, corresponding to the four corners of the tetrahedron of diagonal density matrices, and these are readily identified as four orthogonal Bell states (maximally entangled pure states).

We now consider the action of the partial transposition on the tetrahedron of diagonal density matrices. It preserves the standard form and transforms the coefficients as $d_1 \to d_1, d_2 \to -d_2, d_3 \to d_3$. Thus it produces a mirror image of the tetrahedron, by a reflection in the d_2 direction (see Fig. 4). The density matrices of standard form belonging to the set $\mathcal{P}$, i.e., satisfying the Peres condition that they remain positive after the partial transposition, form an octahedron which is the intersection of the two tetrahedra.

We will now show that for the density matrices of standard form the Peres condition is both necessary and sufficient for separability. What we have to show is that all the density matrices of the octahedron are separable. Since, in general, the separable matrices form a convex set, it is sufficient to show that the corners of the octahedron correspond to separable states.

The density matrices of the octahedron satisfy a single inequality

$$|d_1| + |d_2| + |d_3| \le 1 \tag{62}$$

and its six corners are $(d_1,d_2,d_3) = (\pm1,0,0),(0,\pm1,0),(0,0,\pm1)$, corresponding to the midpoints of the six edges of each of the two tetrahedra. The corners are separable by the identities

$$\frac{1}{4}(\mathbb{I} \pm \sigma_k \otimes \sigma_k) = \frac{1}{8}[(\mathbb{I} + \sigma_k) \otimes (\mathbb{I} \pm \sigma_k) + (\mathbb{I} - \sigma_k) \otimes (\mathbb{I} \mp \sigma_k)]. \tag{63}$$

This completes our proof that the Peres condition is both necessary and sufficient for separability of density matrices on the standard form. Furthermore, since any (nonsingular) density matrix can be obtained from a density matrix of standard form by a transformation that preserves both separability and the Peres condition, this reproduces the known result that for the 2×2 system the set of density matrices that remain positive after a partial transposition is identical to the set of separable density matrices.

With this we conclude the discussion of the two-qubit system. The main point has been to show the usefulness of applying the nonunitary Lorentz transformations in the discussion of separability. Also in higher dimensions such transformations can be applied in the form of $\mathrm{SL}(n,\mathbb{C})$ transformations, although not in precisely the same form as with two-level systems.

V. HIGHER DIMENSIONS

The relativistic formulation is specific for the two-level system, but some elements can be generalized to higher dimensions. We consider first a single system with Hilbert space dimension n. Again, if the trace condition is relaxed, the symmetry group $\mathrm{SU}(n)$ of the set of density matrices is extended to $\mathrm{SL}(n,\mathbb{C})$. The Hilbert–Schmidt metric is *not* invariant under this larger group, but the determinant is invariant for any n. However, it is only for $n=2$ that the determinant is quadratic and can be interpreted as defining an invariant indefinite metric.

The generalization to $n>2$ of the Lorentz boosts are the Hermitian matrices

$$V = V^\dagger, \quad \det V = 1 \tag{64}$$

and expressed in terms of the $\mathrm{SU}(n)$ generators $\mathbf{J}$ they have the form

$$V = \exp(\boldsymbol{\xi} \cdot \mathbf{J}) \tag{65}$$

with real group parameters $\boldsymbol{\xi}$. Here $\mathbf{J}$ and $\boldsymbol{\xi}$ are n^2-1 dimensional vectors. Any *strictly positive* density matrix ρ (with $\det\rho>0$) can be factorized in terms of Hermitian matrices (65) as

$$\rho = NV^2 = N\exp(2\boldsymbol{\xi} \cdot \mathbf{J}) \tag{66}$$

with the normalization factor

$$N = \frac{1}{\mathrm{Tr}\exp(2\boldsymbol{\xi} \cdot \mathbf{J})} = (\det\rho)^{1/n}. \tag{67}$$

Thus in the same way as for $n=2$, any strictly positive density matrix can be generated from the maximally mixed state by an $\mathrm{SL}(n,\mathbb{C})$ transformation of the form (65). The boundary matrices, however, which satisfy $\det\rho=0$, cannot be ex-

pressed in this way, they can be reached by Hermitian transformations only asymptotically, as $|\boldsymbol{\xi}| \to \infty$. In this limit $N \to 0$, and therefore $\mathrm{Tr}\, V^2 \to \infty$ for the non-normalized density matrix V^2.

Schmidt decomposition

We now consider a composite system, consisting of two subsystems of dimension n_A and n_B, and assume ρ to be a strictly positive density matrix on the Hilbert space $\mathcal{H}=\mathcal{H}_A \otimes \mathcal{H}_B$ of the composite system. The general expansion of ρ in terms of the SU(n) generators is given by Eq. (26). Our objective is to show that by a transformation of the form (28), with $V_A \in \mathrm{SL}(n_A, \mathbb{C})$ and $V_B \in \mathrm{SL}(n_B, \mathbb{C})$, followed by normalization, the density matrix can be transformed to the simpler form (30).

Let $\mathcal{D}_A$ and $\mathcal{D}_B$ be the sets of density matrices of the two subsystems A and B. The Cartesian product $\mathcal{D}_A \times \mathcal{D}_B$, consisting of all product density matrices $\rho_A \otimes \rho_B$ with normalization $\mathrm{Tr}\, \rho_A = \mathrm{Tr}\, \rho_B = 1$, is a compact set of matrices on the full Hilbert space $\mathcal{H}$. For the given density matrix ρ we define the following function of ρ_A and ρ_B, which does not depend on the normalizations of ρ_A and ρ_B,

$$f(\rho_A, \rho_B) = \frac{\mathrm{Tr}[\rho(\rho_A \otimes \rho_B)]}{(\det \rho_A)^{1/n_A} (\det \rho_B)^{1/n_B}}. \tag{68}$$

This function is well-defined on the interior of $\mathcal{D}_A \times \mathcal{D}_B$, where $\det \rho_A > 0$ and $\det \rho_B > 0$. Because ρ is assumed to be strictly positive, we have the strict inequality

$$\mathrm{Tr}[\rho(\rho_A \otimes \rho_B)] > 0 \tag{69}$$

and since $\mathcal{D}_A \times \mathcal{D}_B$ is compact, we have an even stronger inequality on $\mathcal{D}_A \times \mathcal{D}_B$,

$$\mathrm{Tr}[\rho(\rho_A \otimes \rho_B)] \geq C > 0 \tag{70}$$

with a lower bound C depending on ρ. It follows that $f \to \infty$ on the boundary of $\mathcal{D}_A \times \mathcal{D}_B$, where either $\det \rho_A = 0$ or $\det \rho_B = 0$. It follows further that f has a positive minimum on the interior of $\mathcal{D}_A \times \mathcal{D}_B$, with the minimum value attained for at least one product density matrix $\tau_A \otimes \tau_B$ with $\det \tau_A > 0$ and $\det \tau_B > 0$. For τ_A and τ_B we may use the representation (66), written as

$$\tau_A = T_A^\dagger T_A, \quad \tau_B = T_B^\dagger T_B \tag{71}$$

ignoring normalization factors. The matrices $T_A \in \mathrm{SL}(n_A, \mathbb{C})$ and $T_B \in \mathrm{SL}(n_B, \mathbb{C})$ may be chosen to be Hermitian, but they need not be, since they may be multiplied from the left by arbitrary unitary matrices. We further write $T = T_A \otimes T_B$, so that

$$\tau_A \otimes \tau_B = T^\dagger T. \tag{72}$$

Now define a transformed density matrix

$$\tilde{\rho} = T \rho T^\dagger \tag{73}$$

and define

$$\tilde{f}(\rho_A, \rho_B) = \frac{\mathrm{Tr}[\tilde{\rho}(\rho_A \otimes \rho_B)]}{(\det \rho_A)^{1/n_A} (\det \rho_B)^{1/n_B}} = f(T_A^\dagger \rho_A T_A, T_B^\dagger \rho_B T_B). \tag{74}$$

This transformed function $\tilde{f}$ has a minimum for

$$\rho_A \otimes \rho_B = (T^\dagger)^{-1} \tau_A \otimes \tau_B T^{-1} = \mathbb{I}_A \otimes \mathbb{I}_B = \mathbb{I}. \tag{75}$$

Since $\tilde{f}$ is stationary under infinitesimal variations about the minimum, it follows that

$$\mathrm{Tr}[\tilde{\rho} \delta(\rho_A \otimes \rho_B)] = 0 \tag{76}$$

for all infinitesimal variations

$$\delta(\rho_A \otimes \rho_B) = \delta\rho_A \otimes \mathbb{I}_B + \mathbb{I}_A \otimes \delta\rho_B \tag{77}$$

subject to the constraints $\det(\mathbb{I}_A + \delta\rho_A) = \det(\mathbb{I}_B + \delta\rho_B) = 1$, or equivalently,

$$\mathrm{Tr}(\delta\rho_A) = \mathrm{Tr}(\delta\rho_B) = 0. \tag{78}$$

The variations satisfying the constraints are the general linear combinations of the SU generators,

$$\delta\rho_A = \sum_i \delta c_i^A J_i^A, \quad \delta\rho_B = \sum_j \delta c_j^B J_j^B. \tag{79}$$

It follows that

$$\mathrm{Tr}(\tilde{\rho} J_i^A \otimes \mathbb{I}_B) = \mathrm{Tr}(\tilde{\rho} \mathbb{I}_A \otimes J_j^B) = 0 \tag{80}$$

for all SU(n_A) generators J_i^A and all SU(n_B) generators J_j^B. This means that the terms proportional to $J_i^A \otimes \mathbb{I}_B$ and $\mathbb{I}_A \otimes J_j^B$ vanish in the expansion for $\tilde{\rho}$, which therefore has the form

$$\tilde{\rho} = \frac{1}{n_A n_B} \left(\mathbb{I} + \sum_{k=1}^{n_A^2-1} \sum_{l=1}^{n_B^2-1} \xi_{kl} J_k^A \otimes J_l^B \right). \tag{81}$$

In order to write $\tilde{\rho}$ in the Schmidt decomposed form (30), we have to make a change of basis, from the fixed basis sets J_i^A and J_j^B to other orthonormal SU generators $\tilde{J}_i^A$ and $\tilde{J}_j^B$ depending on $\tilde{\rho}$. This final Schmidt decomposition involves a standard singular value decomposition of the matrix ξ_{kl} by orthogonal transformations. We may make further unitary transformations $U = U_A \otimes U_B$, but as already pointed out, this is in general not sufficient to obtain a standard form independent of ρ.

VI. NUMERICAL APPROACH TO THE STUDY OF SEPARABILITY

In higher dimensions the Peres condition is not sufficient to identify the separable states. In other words, there exist entangled states that remain positive after a partial transposition. This is known not only from general theoretical considerations [5], but also from explicit examples [15]. States of this type have been referred to as having *bound* entanglement. However, whereas it is a fairly simple task to check the Peres condition, it is in general difficult to identify the separable states [6].

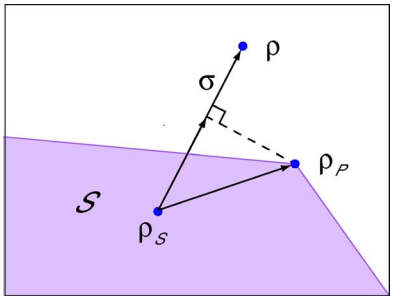


FIG. 5. (Color online) Schematic illustration of the method of finding the separable state closest to a test matrix ρ. The matrix ρ_s represents the best approximation to the closest separable state at a given iteration step, while ρ_p is the product matrix that maximizes the scalar product with the matrix $\sigma=\rho-\rho_s$. This matrix is used to improve ρ_s at the next iteration step. The shaded area $\mathcal{S}$ in the figure illustrates a section through the set of separable density matrices.

In this section we discuss a general numerical method for identifying separability, previously introduced in [14]. It is based on an iterative scheme for calculating the closest separable state and the distance to it, given an arbitrary density matrix (test state). The method can be used to test single density matrices for separability or to make a systematic search to identify the boundary of the set of separable density matrices. After giving an outline of the method we show how to apply the method in a numerical study of bound entanglement in a 3×3 system.

A. Outline of the method

Assume a test state ρ has been chosen. This may typically be close to the boundary of the set of states that satisfy the Peres condition. Let ρ_s be a separable state, an approximation in the iterative scheme to the closest separable state. We may start, for example, with $\rho_s=\mathbb{I}/n$, the maximally mixed state, or with any pure product state. The direction from ρ_s to ρ is denoted $\sigma=\rho-\rho_s$. In order to improve the estimate ρ_s we look for a pure product state ρ_p that maximizes the scalar product

$$s = \mathrm{Tr}[(\rho_p - \rho_s)\sigma] \tag{82}$$

or equivalently, maximizes $s'=\mathrm{Tr}(\rho_p\sigma)$ (see Fig. 5). If $s>0$, then it is possible to find a closer separable state ρ_s' by mixing in the product state ρ_p. This search for closer separable states is iterated, either until no pure product state ρ_p can be found such that $s>0$, which means that ρ_s is already the unique separable state closest to ρ, or until some other convergence criterion is satisfied.

There are two separate mathematical subproblems that have to be solved numerically in this scheme. The first problem is to find the pure product state maximizing the scalar product s'. The second problem is the so-called quadratic programming problem: given a finite number of pure product states, to find the convex combination of these which is closest to the given state ρ. Our approach to these two problems is described briefly below. We refer to Ref. [14] for more details.

To approach the first subproblem, note that a pure product state ρ_p has matrix elements of the form

$$\langle ij|\rho_p|kl\rangle = \phi_i\chi_j\phi_k^*\chi_l^*, \tag{83}$$

where $\Sigma_i|\phi_i|^2=\Sigma_j|\chi_j|^2=1$. We want to find complex coefficients ϕ_i and χ_j that maximize

$$s' = \mathrm{Tr}(\rho_p\sigma) = \sum_{i,j,k,l} \phi_i^*\chi_j^*\sigma_{ij;kl}\phi_k\chi_l. \tag{84}$$

The following iteration scheme turns out in practice to be an efficient numerical method. It may not necessarily give a global maximum, but at least it gives a useful local maximum that may depend on a randomly chosen starting point.

The method is based on the observation that the maximum value of s' is actually the maximal eigenvalue μ in the two linked eigenvalue problems

$$\sum_k A_{ik}\phi_k = \mu\phi_i, \quad \sum_l B_{jl}\chi_l = \mu\chi_j, \tag{85}$$

where

$$A_{ik} = \sum_{j,l} \chi_j^*\sigma_{ij;kl}\chi_l, \quad B_{jl} = \sum_{i,k} \phi_i^*\sigma_{ij;kl}\phi_k. \tag{86}$$

Thus we may start with any arbitrary unit vector $|\chi\rangle=\Sigma_j\chi_j|j\rangle_B\in\mathcal{H}_B$ and compute the Hermitian matrix A. We compute the unit vector $|\phi\rangle=\Sigma_i\phi_i|i\rangle_A\in\mathcal{H}_A$ as an eigenvector of A with maximal eigenvalue, and we use it to compute the Hermitian matrix B. Next, we compute a new unit vector $|\chi\rangle$ as an eigenvector of B with maximal eigenvalue, and we iterate the whole procedure.

This iteration scheme is guaranteed to produce a nondecreasing sequence of function values s', which must converge to a maximum value μ. This is at least a local maximum, and there corresponds to it at least one product vector $|\phi\rangle\otimes|\chi\rangle$ and product density matrix $\rho_p=(|\phi\rangle\langle\phi|)\otimes(|\chi\rangle\langle\chi|)$.

The above construction of ρ_p implies, if $s>0$, that there exist separable states

$$\rho_s' = (1-\lambda)\rho_s + \lambda\rho_p \tag{87}$$

with $0<\lambda\leq 1$, closer to ρ than ρ_s is. However, it turns out to be very inefficient to search only along the line segment from ρ_s to ρ_p for a better approximation to ρ. It is much more efficient to append the new ρ_p to a list of product states ρ_{pk} found in previous iterations, and then minimize

$$F = \mathrm{Tr}\left(\rho - \sum_k \lambda_k\rho_{pk}\right)^2 \tag{88}$$

which is a quadratic function of coefficients $\lambda_k\geq 0$ with $\Sigma_k\lambda_k=1$. We solve this quadratic programming problem by an adaptation of the conjugate gradient method, and we throw away a given product matrix ρ_{pk} if and only if the corresponding coefficient λ_k becomes zero when F is minimized. In practice, this means that we may construct altogether several hundred or even several thousand product states, but only a limited number of those, typically less than 100 in the cases we have studied, are actually included in the final approximation ρ_s.

B. Bound entanglement in the 3×3 system

For the 3×3 system (composed of two three-level systems) there are explicit examples of entangled states that remain positive under a partial transposition. This was first discussed by Horodecki [15] and then an extended set of states was found by Bruss and Peres [16]. We apply the method outlined above to density matrices limited to a two-dimensional planar section of the full set. The section is chosen to contain one of the Horodecki states and a Bell state in addition to the maximally mixed state, and the method is used to identify the boundary of the separable states in this two-dimensional plane.

Since the separable states $\mathcal{S}$ are contained in the set $\mathcal{P}$ of states that remain positive under partial transposition, we start the search for the boundary of $\mathcal{S}$ with a series of states located at the boundary of $\mathcal{P}$. This boundary is found by solving the algebraic equations $\det\rho=0$ and $\det\rho^P=0$. For each chosen state we find the distance to the closest separable state and change the test state ρ on a straight line between this point and the maximally mixed state, in a step of length equal to the evaluated distance. In a small number of steps the intersection of the straight line and the boundary of the separable states is found within a small error, typically chosen to be 10^{-6}. (The distance from the maximally mixed state to the pure states in this case is $d=2/3$.)

In Fig. 6 we show a plot of the results of the calculations. The numerically determined points on the border of the set of separable states $\mathcal{S}$ are indicated by black dots, while the border of the set of states $\mathcal{P}$ that satisfy Peres' condition, determined by solving the algebraic equations, is shown as lines which cross at the position of the Horodecki state. One should note that the states with bound entanglement in the corner of the set $\mathcal{P}$ cover a rather small area.

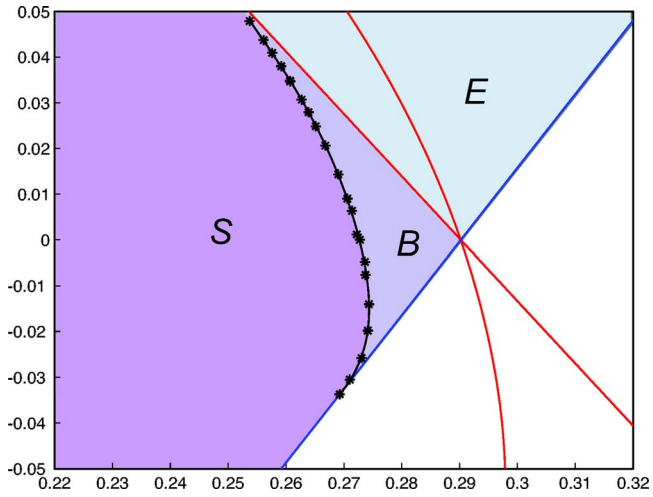


FIG. 6. (Color online) The boundary of the set of separable states in a two-dimensional section, as determined numerically. $\mathcal{S}$ denotes the set of separable states, $\mathcal{B}$ the states with bound entanglement, and $\mathcal{E}$ the entangled states violating the Peres condition. The straight line from lower left to upper right (in blue) is determined by the algebraic equation $\det\rho=0$ and gives the boundary of the full set of density matrices. The two curves from upper left to lower right (in red) are determined by the equation $\det\rho^P=0$, in particular, the red straight line gives the boundary of the set of states that satisfy the Peres condition. The black dots are the numerically determined points on the boundary of the set of separable states. The coordinate values in the plot are a factor of $\sqrt{2}$ too large according to the definition of distance in the text.

VII. CONCLUSIONS

To summarize, we have in this paper focused on some basic questions concerning the geometry of separability. The simplest case of 2×2 matrices has been used to demonstrate the usefulness of relaxing the normalization requirement $\mathrm{Tr}\,\rho=1$. Thus if this condition is replaced by $\det\rho=1$, a relativistic description with a Minkowski metric can be used, where all (nonpure) states can be connected by Lorentz transformations. For a composite system consisting of two two-level systems, independent Lorentz transformations performed on the two subsystems can be used to diagonalize an arbitrary density matrix in a way that respects separability. We have used this diagonalization to demonstrate the known fact that the Peres condition and the separability condition are equivalent in this case.

Although the diagonalization with Lorentz transformations is restricted to the composite 2×2 system, we have shown that the generalized form of the Schmidt decomposition used in this diagonalization can be extended to higher dimensions. The decomposition involves the use of nonunitary $\mathrm{SL}(n,\mathbb{C})$ transformations for the two subsystems. Although a full diagonalization is not obtained in this way, we suggest that the Schmidt decomposed form may be of interest in the study of separability and bound entanglement.

A third part of the paper has been focused on the use of a numerical method to study separability. This method exploits the fact that the set of separable states is convex, and is based on an iterative scheme to find the closest separable state for arbitrary density matrices. We have demonstrated the use of this method in a numerical study of bound entanglement in the case of a 3×3 system. A further study of separability with this method is under way.

ACKNOWLEDGMENTS

We acknowledge the support of NorForsk under the Nordic network program *Low-dimensional physics: The theoretical basis of nanotechnology*. One of us (J.M.L.) would like to thank Professor B. Janko at the Institute for Theoretical Sciences, a joint institute of University of Notre Dame and the Argonne National Laboratory, for the support and hospitality during a visit.

[1] M. Kus and K. Zyczkowski, Phys. Rev. A **63**, 032307 (2001).
[2] F. Verstraete, J. Dehaene, and B. De Moor, J. Mod. Opt. **49**, 1277 (2002).
[3] A. O. Pittenger and M. H. Rubin, Phys. Rev. A **67**, 012327 (2003).
[4] A. Peres, Phys. Rev. Lett. **77**, 1413 (1996).
[5] M. Horodecki, P. Horodecki, and R. Horodecki, Phys. Lett. A **223**, 1 (1996).
[6] L. Gurvits, in *Proceedings of the Thirty-Fifth ACM Symposium on Theory of Computing* (ACM, New York, 2003), pp. 10-19.
[7] P. Arrighi and C. Patricot, Phys. Rev. A **68**, 042310 (2003).
[8] P. Arrighi and C. Patricot, J. Phys. A **36**, L287 (2003).
[9] L. M. Ioannou, B. C. Travaglione, D. C. Cheung, and A. K. Ekert, Phys. Rev. A **70**, 060303(R) (2004).
[10] F. Hulpke and D. Bruss, J. Phys. A **38**, 5573 (2005).
[11] P. Horodecki, M. Lewenstein, G. Vidal, and I. Cirac, Phys. Rev. A **62**, 032310 (2000).
[12] A. C. Doherty, P. A. Parrilo, and F. M. Spedalieri, Phys. Rev. Lett. **88**, 187904 (2002).
[13] P. Badziąg, P. Horodecki, and R. Horodecki, e-print quant-ph/0504041.
[14] G. Dahl, J. M. Leinaas, J. Myrheim, and E. Ovrum, Linear Algebra and its Applications (to be published).
[15] P. Horodecki, Phys. Lett. A **232**, 333 (1997).
[16] D. Bruss and A. Peres, Phys. Rev. A **61**, 030301(R) (2000).
[17] G. H. Golub and C. F. Van Loan, *Matrix Computations* (John Hopkins University Press, Baltimore, MD, 1996).
[18] Li-Xiang Cen, Xin-Qi Li, and Yi Jing Yan, J. Phys. A **36**, 12267 (2003).
[19] A. Osterloh and J. Siewert, Phys. Rev. A **72**, 012337 (2005).

Paper II

Geir Dahl, Jon Magne Leinaas, Jan Myrheim and Eirik Ovrum,
A tensor product matrix approximation problem in quantum physics,
Linear Algebra and its Applications **Volume 420**, Issues 2-3 , 15 January 2007,
Pages 711-725

Available online at www.sciencedirect.com
ScienceDirect
Linear Algebra and its Applications 420 (2007) 711–725
LINEAR ALGEBRA AND ITS APPLICATIONS
www.elsevier.com/locate/laa

A tensor product matrix approximation problem in quantum physics

Geir Dahl [a,*], Jon Magne Leinaas [b], Jan Myrheim [c], Eirik Ovrum [d]

[a] *Center of Mathematics for Applications, Department of Informatics, University of Oslo, P.O. Box 1053, Blindern, NO-0316 Oslo, Norway*
[b] *Department of Physics, University of Oslo, P.O. Box 1048, Blindern, NO-0316 Oslo, Norway*
[c] *Department of Physics, Norwegian University of Science and Technology, NO-7491 Trondheim, Norway*
[d] *Center of Mathematics for Applications, Department of Physics, University of Oslo, P.O. Box 1048, Blindern, NO-0316 Oslo, Norway*

Received 11 May 2006; accepted 31 August 2006
Available online 17 October 2006
Submitted by R.A. Brualdi

Abstract

We consider a matrix approximation problem arising in the study of *entanglement* in quantum physics. This notion represents a certain type of correlations between subsystems in a composite quantum system. The states of a system are described by a density matrix, which is a positive semidefinite matrix with trace one. The goal is to approximate such a given density matrix by a so-called separable density matrix, and the distance between these matrices gives information about the degree of entanglement in the system. Separability here is expressed in terms of tensor products. We discuss this approximation problem for a composite system with two subsystems and show that it can be written as a convex optimization problem with special structure. We investigate related convex sets, and suggest an algorithm for this approximation problem which exploits the tensor product structure in certain subproblems. Finally some computational results and experiences are presented.

AMS classification: 15A90; 15A69; 52A41; 90C25

Keywords: Density matrix; Entanglement; Tensor product; Matrix approximation; Positive semidefinite; Convex sets

* Corresponding author.
E-mail addresses: geird@math.uio.no (G. Dahl), j.m.leinaas@fys.uio.no (J.M. Leinaas), jan.myrheim@ntnu.no (J. Myrheim), ovrum@fys.uio.no (E. Ovrum).

0024-3795/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.laa.2006.08.026

1. Introduction

A current problem in physics is to give a precise characterization of *entanglement* in a quantum system. This describes types of correlations between subsystems of the full quantum system that go beyond the statistical correlations that can be found in a classical composite system. The interest is motivated by ideas about utilizing entanglement to perform communication and computation in qualitatively new ways.

Although a general quantitative definition of the degree of entanglement of a composite system does not exist, there is a generally accepted definition that distinguishes between quantum states with and without entanglement. The non-entangled states are referred to as *separable states*, and they are considered to only contain classical correlations between the subsystems. In addition, for some special cases a generally accepted quantitative measure of entanglement exists.

The standard mathematical formulation of a composite quantum system is in terms of *density matrices*. These describe the quantum states of either the full system or one of its subsystems, and correspond to hermitian, positive semidefinite operators that act on a complex vector space, either finite or infinite dimensional. The density matrices also satisfy a normalization condition so that they form a compact convex set in the vector space of hermitian matrices. For a composite system the separable states form a subset of the density matrices that is also compact and convex.

The physical interpretation of a density matrix is that it contains information about the statistical distribution over measurable values for any observable of the quantum system. Such an observable is represented by a hermitian matrix A, with the eigenvalues corresponding to the possible outcomes of a measurement. In particular, for a given quantum state represented by a density matrix ρ, the expectation value of the observable A is defined by the trace of the product, $\mathrm{tr}(A\rho)$. A density matrix that is a projection on a single vector is referred to as representing a *pure state*, while other density matrices are representing *mixed states*. A mixed state can be interpreted as corresponding to a statistical distribution over pure states, and in this sense includes both quantum uncertainty (through the pure states) and classical uncertainty (through the statistical distribution).

To identify whether a state, i.e., a density matrix, is entangled or not is for a general quantum state considered to be a hard problem [5]. However, some general results exist concerning the separability problem, in particular a simple sufficient condition for entanglement has been found [12], and also general schemes to test separability numerically have been suggested [3,9]. Other results refer to more special cases, for matrices of low dimensions or for restricted subsets of density matrices [8]. There have also been attempts to more general approaches to the identification of separability by use of the natural geometrical structure of the problem, where the Euclidean metric (Hilbert–Schmidt or Frobenius norm) of the hermitian matrices can be used to give a geometrical characterization of the set of separable matrices, see [13,14] and references therein.

In the present paper we focus on the geometrical description of separability and consider the problem of finding the closest separable state (in the Frobenius norm) to any given state, either entangled or non-entangled. We consider the case of a quantum system with two subsystems, where the full vector space is the tensor product of the vector spaces of the subsystems, and where the vector spaces are finite dimensional. The distance to the closest separable state can be used as a measure of the degree of entanglement of the chosen state, although there are certain conditions a *good* measure of entanglement should satisfy, that are not obviously satisfied by the Euclidean norm, see [11]. However, beyond this question it seems that an efficient method to identify the nearest separable state should be useful in order to identify the boundary between the separable and entangled states. Obviously, if a density matrix is separable the distance to the closest separable state is zero, hence our method is a numerical test for separability.

In Section 2 we present the mathematical framework for our investigations. In Sections 4–6 we consider a systematic, iterative method which approximates the closest separable state (the DA algorithm). This is a two step algorithm, where the first step is to optimize the convex combination of a set of tensor product matrices. The second step is to find a best new tensor product matrix to add to the existing set. These two steps are implemented iteratively so that the expansion coefficients are optimized each time a new product matrix is included and a new optimal product matrix is found after optimization of the previous set. We give some numerical examples where the algorithm is used to find the closest separable state and to efficiently identify the boundary between the separable and entangled states.

In order to be specific, and to simplify the formulation of the theory, we shall consider in this paper mostly real matrices, with the density matrices restricted to be symmetric. We want to emphasize, however, that our algorithms can be immediately applied to complex matrices, which is the relevant case for quantum theory. The only modification needed then is that transposition of real matrices is generalized to hermitian conjugation of complex matrices, and the class of real symmetric matrices is generalized to complex hermitian matrices. In Section 7 we present an example of application to quantum theory, with complex density matrices.

We may remark here that there is a qualitative difference between the separability problems for real and complex matrices. In fact, the set of separable density matrices has the same dimension as the full set of density matrices in the complex case, but has lower dimension in the real case. The last result follows from the Peres separability criterion [12], since a separable matrix must be symmetric under partial transposition if only real matrices are involved. Thus, among all real density matrices that are separable as members of the set of complex density matrices, only a subset of lower dimension are separable in terms of real matrices only.

2. Formulation of the problem

We introduce first the mathematical framework needed to formulate the problem to be investigated. This includes a summary of the properties of density matrices and separable density matrices as expressed through the definitions and theorems given below. We end the section by formulating the problem to be investigated, referring to this as the density approximation problem.

We discuss here in detail only the case of real vectors and real matrices, because this simplifies our discussion, and because the generalization to the complex case is quite straightforward, based on the close correspondence between transposition of real matrices, on the one hand, and hermitian conjugation of complex matrices, on the other hand. In particular, the set of real symmetric matrices is expanded to the set of complex hermitian matrices, which is still a real vector space with the same positive definite real inner product $\mathrm{tr}(AB)$ (see below).

To explain the notation used, we consider the usual Euclidean space $\mathbb{R}^n$ of real vectors of length n equipped with the standard inner product and the Euclidean norm denoted by $\|\cdot\|$. Vectors are treated as column vectors, and the transpose of a vector x is denoted by x^{T}. The convex hull of a set S is the intersection of all convex sets containing S, and it is denoted by $\mathrm{conv}(S)$. Some recommended references on convexity are [15,1]. We let I denote the identity matrix (of order n, where n should be clear from the context). The unit ball in $\mathbb{R}^n$ is $B_n = \{x \in \mathbb{R}^n : \|x\| = 1\}$.

Let $\mathscr{S}^n$ denote the linear space consisting of all the real symmetric $n \times n$ matrices. This space has dimension $n(n+1)/2$. In $\mathscr{S}^n$ we use the standard inner product

$$\langle A, B\rangle = \mathrm{tr}(AB) = \sum_{i,j} a_{ij} b_{ij} \quad (A, B \in \mathscr{S}^n).$$

Here $\mathrm{tr}(C) = \sum_{i=1}^{n} c_{ii}$ denotes the trace of a matrix C. The associated matrix norm is the Frobenius norm $\|A\|_{\mathrm{F}} = (\sum_{i,j} a_{ij}^2)^{1/2}$ and we use this norm for measuring distance in the matrix approximation problem of interest. A matrix $A \in \mathcal{S}^n$ is positive semidefinite provided that $x^{\mathrm{T}}Ax \geqslant 0$ for all $x \in \mathbb{R}^n$, and this will be denoted by $A \succeq 0$. We define the *positive semidefinite cone*

$$\mathcal{S}^n_+ = \{A \in \mathcal{S}^n : A \succeq 0\}$$

as the set of all symmetric positive semidefinite matrices of order n. $\mathcal{S}^n_+$ is a full-dimensional closed convex cone in $\mathcal{S}^n$, so $\lambda_1 A_1 + \lambda_2 A_2 \in \mathcal{S}^n_+$ whenever $A_1, A_2 \in \mathcal{S}^n_+$ and $\lambda_1, \lambda_2 \geqslant 0$. For more about positive semidefinite matrices we refer to the excellent book in matrix theory [6]. A *density matrix* is a matrix in $\mathcal{S}^n_+$ with trace 1. We let $\mathcal{T}^n_+$ denote the set of all density matrices of order n, so

$$\mathcal{T}^n_+ = \{A \in \mathcal{S}^n_+ : \mathrm{tr}(A) = 1\}.$$

The set $\mathcal{T}^n_+$ is convex and we can determine its extreme points. Recall that a point x in a convex set C is called an extreme point when there is no pair of distinct points $x_1, x_2 \in C$ with $x = (1/2)x_1 + (1/2)x_2$.

Theorem 2.1. *The set $\mathcal{T}^n_+$ of density matrices satisfies the following*:

(i) *$\mathcal{T}^n_+$ is the intersection of the positive semidefinite cone $\mathcal{S}^n_+$ and the hyperplane $H = \{A \in \mathcal{S}^n : \langle A, I\rangle = 1\}$.*
(ii) *$\mathcal{T}^n_+$ is a compact convex set of dimension $n(n+1)/2 - 1$.*
(iii) *The extreme points of $\mathcal{T}^n_+$ are the symmetric rank one matrices $A = xx^{\mathrm{T}}$ where $x \in \mathbb{R}^n$ satisfies $\|x\| = 1$. Therefore*

$$\mathcal{T}^n_+ = \mathrm{conv}\{xx^{\mathrm{T}} : x \in \mathbb{R}^n,\ \|x\| = 1\}.$$

Proof. Property (i) follows from the definition of $\mathcal{T}^n_+$ as $\langle A, I\rangle = \sum_i a_{ii} = \mathrm{tr}(A)$. So H is the solution set of a single linear equation in the space $\mathcal{S}^n$ and therefore H is a hyperplane. Thus, $\mathcal{T}^n_+$ is the intersection of two closed convex sets, and this implies that $\mathcal{T}^n_+$ is also closed and convex. Moreover, $\mathcal{T}^n_+$ is bounded as one can prove that each $A \in \mathcal{T}^n_+$ satisfies $-1 \leqslant a_{ij} \leqslant 1$ for $1 \leqslant i,\ j \leqslant n$. (This follows from the facts that $\mathrm{tr}(A) = 1$, $e_i^{\mathrm{T}}Ae_i \geqslant 0$ and $(e_i + e_j)^{\mathrm{T}}A(e_i + e_j) \geqslant 0$ where e_i denotes the ith unit vector in $\mathbb{R}^n$.) Since $\mathcal{T}^n_+$ lies in the hyperplane H, $\mathcal{T}^n_+$ has dimension at most $\dim(\mathcal{S}^n) - 1 = n(n+1)/2 - 1$. Consider the matrices $(1/2)(e_i + e_j)(e_i + e_j)^{\mathrm{T}}$ $(1 \leqslant i < j \leqslant n)$ and $e_i e_i^{\mathrm{T}}$ $(1 \leqslant i \leqslant n)$. One can check that these $n(n+1)/2$ matrices are affinely independent (in the space $\mathcal{S}^n$) and they all lie in $\mathcal{T}^n_+$. It follows that $\dim(\mathcal{T}^n_+) = n(n+1)/2 - 1$. This proves Property (ii).

It remains to determine the extreme points of $\mathcal{T}^n_+$. Let $A \in \mathcal{T}^n_+$. Since A is real and symmetric it has a spectral decomposition

$$A = VDV^{\mathrm{T}},$$

where V is a real orthogonal $n \times n$ matrix and D is a diagonal matrix with the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ of A on the diagonal. By partitioning V by its columns $v_1, v_2, \ldots, v_n$, where v_i is the eigenvector corresponding to λ_i, we get

$$A = \sum_{i=1}^{n} \lambda_i v_i v_i^{\mathrm{T}}. \tag{1}$$

Since A is positive semidefinite, all the eigenvalues are non-negative. Moreover, $\sum_i \lambda_i = \mathrm{tr}(A) = 1$. Therefore, the decomposition (1) actually represents A as a convex combination of rank one matrices $v_i v_i^{\mathrm{T}}$ where $\|v_i\| = 1$ (as V is orthogonal). From this it follows that all extreme points of $\mathscr{T}^n$ are rank one matrices xx^{T} where $\|x\| = 1$. One can also verify that *all* matrices of this kind are indeed extreme points, but we omit the details here. Finally, from convexity theory (see [15]) the Krein–Milman theorem says that a compact convex set is the convex hull of its extreme points which gives the final property in the theorem. □

The theorem shows that the extreme points of the set of density matrices $\mathscr{T}_+^n$ are the symmetric rank one matrices $A_x = xx^{\mathrm{T}}$ where $\|x\| = 1$. Such a matrix A_x is an orthogonal projector (so $A_x^2 = A_x$, and A_x is symmetric) and $A_x y$ is the orthogonal projection of a vector y onto the line spanned by x.

Remark. The spectral decomposition (1) is interesting in this context. First, it shows that any matrix $A \in \mathscr{T}_+^n$ may be written as a convex combination of at most n extreme points. This improves upon a direct application of Carathéodory's theorem (see [15]) which says that A may be represented using at most $\dim(\mathscr{T}^n) + 1 = n(n+1)/2$ extreme points. Secondly, (1) shows how we can decompose A as a convex combination of its extreme points by calculation eigenvectors and corresponding eigenvalues of A. Finally, we remark that the argument above based on the spectral decomposition also shows the well-known fact that the extreme rays of the positive semidefinite cone correspond to the symmetric rank one matrices.

We now proceed and introduce a certain subset of $\mathscr{T}_+^n$ which will be of main interest below. Recall that if $A \in \mathbb{R}^{p\times p}$ and $B \in \mathbb{R}^{q\times q}$ then the tensor product $A \otimes B$ is the square matrix of order pq given by its (i, j)th block $a_{ij}B$ $(1 \leqslant i,\ j \leqslant p)$. A general reference on tensor products is [7].

For the rest of this section we fix two positive numbers p and q and let $n = pq$. We call a matrix $A \in \mathbb{R}^{n\times n}$ *separable* if A can be written as a convex combination

$$A = \sum_{j=1}^{N} \lambda_j B_j \otimes C_j$$

for some positive integer N, matrices $B_j \in \mathscr{T}_+^p$, $C_j \in \mathscr{T}_+^q$ $(j \leqslant N)$ and non-negative numbers λ_j $(j \leqslant N)$ with $\sum_{j=1}^N \lambda_j = 1$. Let $\mathscr{T}_+^{n,\otimes}$ denote the set of all separable matrices of order n. Note that $n = pq$, but p and q are suppressed in our notation. For sets U and W of matrices we let $U \otimes W$ denote the set of matrices that can be written as the tensor product of a matrix in U and a matrix in W. The following theorem summarizes important properties of $\mathscr{T}_+^{n,\otimes}$.

Theorem 2.2. *The set $\mathscr{T}_+^{n,\otimes}$ of separable matrices satisfies the following*:

(i) $\mathscr{T}_+^{n,\otimes} \subseteq \mathscr{T}_+^n$.
(ii) *$\mathscr{T}_+^{n,\otimes}$ is a compact convex set and $\mathscr{T}_+^{n,\otimes} = \mathrm{conv}(\mathscr{T}_+^p \otimes \mathscr{T}_+^q)$.*
(iii) *The extreme points of $\mathscr{T}_+^{n,\otimes}$ are the symmetric rank one matrices*

$$A = (x \otimes y)(x \otimes y)^{\mathrm{T}},$$

where $x \in \mathbb{R}^p$ and $y \in \mathbb{R}^q$ both have Euclidean length one. So

$$\mathscr{T}_+^{n,\otimes} = \mathrm{conv}\{(x \otimes y)(x \otimes y)^{\mathrm{T}} : x \in B_p,\ y \in B_q\}.$$

Proof. (i) Let $B \in \mathcal{T}_+^p$ and $C \in \mathcal{T}_+^q$, and let $A = B \otimes C$. Then, $A^{\mathrm{T}} = (B \otimes C)^{\mathrm{T}} = B^{\mathrm{T}} \otimes C^{\mathrm{T}} = B \otimes C = A$, so A is symmetric. Moreover, A is positive semidefinite as both B and C are positive semidefinite (actually, the eigenvalues of A are the products of eigenvalues of B and eigenvalues of C, see e.g. [7]). Finally, $\mathrm{tr}(A) = \mathrm{tr}(B)\mathrm{tr}(C) = 1$, so $A \in \mathcal{T}_+^n$. Since a separable matrix is a convex combination of such matrices, and $\mathcal{T}_+^n$ is convex, it follows that every separable matrix lies in $\mathcal{T}_+^n$.

(ii) By definition the set $\mathcal{T}_+^{n,\otimes}$ is the set of all convex combinations of matrices in $\mathcal{T}_+^p \otimes \mathcal{T}_+^q$. From a basic result in convexity (see e.g. [15]) this means that $\mathcal{T}_+^{n,\otimes}$ coincides with the convex hull of $\mathcal{T}_+^p \otimes \mathcal{T}_+^q$. Consider now the function $g : \mathcal{T}^p \times \mathcal{T}^q \to \mathcal{S}^n$ given by $g(B, C) = B \otimes C$ where $B \in \mathcal{S}^p$ and $C \in \mathcal{S}^q$. Then $g(\mathcal{T}_+^p \times \mathcal{T}_+^q) = \mathcal{T}_+^p \otimes \mathcal{T}_+^q$ and the function g is continuous. Therefore $\mathcal{T}_+^{n,\otimes}$ is compact as $\mathcal{T}_+^p \times \mathcal{T}_+^q$ is compact (and the convex hull of a compact set is again compact, [15]).

(iii) Let A be an extreme point of $\mathcal{T}_+^{n,\otimes}$. Then $A = B \otimes C$ for some $B \in \mathcal{T}_+^p$ and $C \in \mathcal{T}_+^q$ (for a convex combination of more than one such matrix is clearly not an extreme point). From Theorem 2.1 we have that $B = \sum_{j=1}^m \lambda_j x_j x_j^{\mathrm{T}}$ and $C = \sum_{k=1}^r \mu_j y_j y_j^{\mathrm{T}}$ for suitable vectors $x_j \in \mathbb{R}^p$ and $y_k \in \mathbb{R}^q$ with $\|x_j\| = \|y_k\| = 1$, and non-negative numbers λ_j $(j \leqslant m)$ and μ_k $(k \leqslant r)$ with $\sum_j \lambda_j = \sum_k \mu_k = 1$. Using basic algebraic rules for tensor products we then calculate

$$A = B \otimes C = \left(\sum_{j=1}^m \lambda_j x_j x_j^{\mathrm{T}}\right) \otimes \sum_{k=1}^r \mu_k y_k y_k^{\mathrm{T}} = \sum_{j,k} \lambda_j \mu_k \left(x_j x_j^{\mathrm{T}}\right) \otimes \left(y_k y_k^{\mathrm{T}}\right)$$
$$= \sum_{j,k} \lambda_j \mu_k (x_j \otimes y_k)\left(x_j^{\mathrm{T}} \otimes y_k^{\mathrm{T}}\right) = \sum_{j,k} \lambda_j \mu_k (x_j \otimes y_k)(x_j \otimes y_k)^{\mathrm{T}}.$$

Since $\sum_{j,k} \lambda_j \mu_k = 1$ this shows that A can be written as a convex combination of matrices of the form $(x \otimes y)(x \otimes y)^{\mathrm{T}}$ where $\|x\| = \|y\| = 1$. Since A is an extreme point we have $m = r = 1$ and we have shown the desired form of the extreme points of $\mathcal{T}_+^{n,\otimes}$. Finally, one can verify that all these matrices are really extreme points, but we omit these details. □

We now formulate the following *density approximation problem* (DA), which we shall examine further in subsequent sections of the paper,

(DA) *Given a density matrix $A \in \mathcal{T}_+^n$ find a separable density matrix $X \in \mathcal{T}_+^{n,\otimes}$ which minimizes the distance $\|X - A\|_{\mathrm{F}}$.*

Separability here refers to the tensor product decomposition $n = pq$, as discussed above.

3. An approach based on projection

In this section we study the DA problem and show that it may be viewed as a projection problem associated with the convex set $\mathcal{T}_+^{n,\otimes}$ introduced in Section 2. This leads to a projection algorithm for solving DA.

The DA problem is to find the best approximation (in Frobenius norm) to a given density matrix $A \in \mathcal{T}_+^n$ in the convex set $\mathcal{T}_+^{n,\otimes}$ consisting of all separable density matrices. This corresponds to the optimization problem

$$\inf\{\|X - A\|_{\mathrm{F}} : X \in \mathcal{T}_+^{n,\otimes}\}. \tag{2}$$

In this problem the function to be minimized is $f : \mathcal{S}^n_+ \to \mathbb{R}$ given by $f(X) = \|X - A\|_F$. Now, f is strictly convex (on the underlying space $\mathcal{S}^n$) and therefore continuous. Moreover, as shown in Theorem 2.2, the set $\mathcal{T}^{n,\otimes}_+$ is compact and convex, so the infimum is attained. These properties imply the following fact, see also [1,2,15].

Theorem 3.1. *For given $A \in \mathcal{T}^n_+$ the approximation problem (2) has a unique optimal solution X^*.*

The unique solution X^* will be called the *projection* of A onto $\mathcal{T}^{n,\otimes}_+$, and we denote this by $X^* = \mathrm{Proj}_\otimes(A)$.

The next theorem gives a variational inequality characterization of the projection $X^* = \mathrm{Proj}_\otimes(A)$. Let $\mathrm{Ext}(\mathcal{T}^{n,\otimes}_+)$ denote the set of all extreme points of $\mathcal{T}^{n,\otimes}_+$; these extreme points were found in Theorem 2.2. The theorem is essentially the *projection theorem* in convexity, see e.g. [1]. We give a proof following the lines of [1] (except that we consider a different inner product space). The ideas in the proof will be used in our algorithm for solving DA below.

Theorem 3.2. *Let $A \in \mathcal{T}^n_+$ and $X \in \mathcal{T}^{n,\otimes}_+$. Then the following three statements are equivalent*:

$$\begin{aligned}
&\text{(i)} \quad X = \mathrm{Proj}_\otimes(A).\\
&\text{(ii)} \quad \langle A - X, Y - X\rangle \leqslant 0 \quad \textit{for all } Y \in \mathcal{T}^{n,\otimes}_+.\\
&\text{(iii)} \quad \langle A - X, Y - X\rangle \leqslant 0 \quad \textit{for all } Y \in \mathrm{Ext}(\mathcal{T}^{n,\otimes}_+).
\end{aligned} \tag{3}$$

Proof. Assume first that (ii) holds and consider $Y \in \mathcal{T}^{n,\otimes}_+$. Then we have

$$\begin{aligned}
\|A - Y\|_F^2 &= \|(A - X) - (Y - X)\|_F^2 = \|A - X\|_F^2 + \|Y - X\|_F^2 - 2\langle A - X, Y - X\rangle\\
&\geqslant \|A - X\|_F^2 + \|Y - X\|_F^2 \geqslant \|A - X\|_F^2.
\end{aligned}$$

So, $\|A - X\|_F \leqslant \|A - Y\|_F$ for all $Y \in \mathcal{T}^{n,\otimes}_+$ and therefore $X = \mathrm{Proj}_\otimes(A)$ and (i) holds.

Conversely, assume that (i) holds and let $Y \in \mathcal{T}^{n,\otimes}_+$. Let $0 \leqslant \lambda \leqslant 1$ and consider the matrix $X(\lambda) = (1 - \lambda)X + \lambda Y$. Then $X(\lambda) \in \mathcal{T}^{n,\otimes}_+$ as $\mathcal{T}^{n,\otimes}_+$ is convex. Consider the function

$$\begin{aligned}
g(\lambda) &= \|A - X(\lambda)\|_F^2 = \|(1 - \lambda)(A - X) + \lambda(A - Y)\|_F^2\\
&= (1 - \lambda)^2\|A - X\|_F^2 + \lambda^2\|A - Y\|_F^2 + 2\lambda(1 - \lambda)\langle A - X, A - Y\rangle.
\end{aligned}$$

So g is a quadratic function of λ and its (right-sided) derivative in $\lambda = 0$ is

$$g'_+(0) = -2\|A - X\|_F^2 + 2\langle A - X, A - Y\rangle = -2\langle A - X, Y - X\rangle$$

and this derivative must be non-negative as $X(0) = X = \mathrm{Proj}_\otimes(A)$. But this gives the inequality $\langle A - X, Y - X\rangle \leqslant 0$ so (ii) holds.

To see the equivalence of (ii) and (iii) recall from Theorem 2.2 that each $Y \in \mathcal{T}^{n,\otimes}_+$ may be represented as a convex combination $Y = \sum_{j=1}^t \lambda_j Y_j$ where $\lambda_j \geqslant 0\,(j \leqslant t)$ and $\sum_j \lambda_j = 1$ and each Y_j is a rank one matrix of the form described in the theorem. Therefore

$$\langle A - X, Y - X\rangle = \left\langle A - X, \sum_{j=1}^{t} \lambda_j Y_j - X \right\rangle = \left\langle A - X, \sum_{j=1}^{t} \lambda_j (Y_j - X) \right\rangle$$
$$= \sum_{j=1}^{t} \lambda_j \langle A - X, Y_j - X\rangle.$$

from which the desired equivalence easily follows. □

We remark that the *variational inequality* characterization given in (3) is the same as the optimality condition one obtains when formulating DA as the convex minimization problem

$$\min\{(1/2)\|A - X\|_{\mathrm{F}}^2 : X \in \mathscr{T}_+^{n,\otimes}\}.$$

In fact, the gradient of $f(X) = (1/2)\|X - A\|_{\mathrm{F}}^2$ is $\nabla f(X) = X - A$ and the optimality characterization here is $\langle \nabla f(X), Y - X\rangle \geqslant 0$ for all $Y \in \mathscr{T}_+^{n,\otimes}$, and this translates into (3). In the next section we consider an algorithm for DA that is based on the optimality conditions we have presented above.

4. The Frank–Wolfe method

We discuss how Theorem 3.2 may be the basis for an algorithm for solving the DA problem. The algorithm is an adaption of a general algorithm in convex programming called the *Frank–Wolfe method* (or the conditional gradient algorithm), see [2]. This is an iterative algorithm where a decent direction is found in each iteration by linearizing the objective function.

The main idea is as follows. Let $X \in \mathscr{T}_+^{n,\otimes}$ be a candidate for being the projection $\mathrm{Proj}_{\otimes}(A)$. We check if $X = \mathrm{Proj}_{\otimes}(A)$ by solving the optimization problem

$$\gamma(X) := \max\{\langle A - X, Y - X\rangle : Y \in \mathrm{Ext}(\mathscr{T}_+^{n,\otimes})\}. \tag{4}$$

We discuss *how* this problem can be solved in the next section.

The algorithm for solving DA may be described in the following way:

The DA algorithm

1. Choose an initial candidate $X \in \mathscr{T}_+^{n,\otimes}$.
2. Optimality test: solve the corresponding problem (4).
3. If $\gamma(X) \leqslant 0$, stop; the current solution X is optimal. Otherwise, let Y^* be an optimal solution of (4). Determine the matrix X' which is nearest to A on the line segment between X and Y^*.
4. Replace X by X' and repeat Steps 1–3 until an optimal solution has been found.

We now discuss this algorithm in some detail. Consider a current solution $X \in \mathscr{T}_+^{n,\otimes}$ and solve (4) as in Step 2. There are two possibilities. First, if $\gamma(X) \leqslant 0$, then, due to Theorem 3.2, we must have that $X = \mathrm{Proj}_{\otimes}(A)$. Thus, the DA problem has been solved. Alternatively, $\gamma(X) > 0$ and we have found $Y^* \in \mathrm{Ext}(\mathscr{T}_+^{n,\otimes})$ such that $\langle A - X, Y^* - X\rangle > 0$. This means that $g'_+(0) < 0$ for the function g introduced in the proof of Theorem 3.2: $g(\lambda) = \|A - X(\lambda)\|_{\mathrm{F}}^2$ where $X(\lambda) = (1 - \lambda)X + \lambda Y^*$. Let λ^* be an optimal solution in the (line search) problem $\min\{g(\lambda) : 0 \leqslant \lambda \leqslant 1\}$. Since g is a quadratic function in one variable, this minimum is easy to find analytically. Since $g'_+(0) < 0$, we have that $\lambda^* > 0$. The corresponding matrix $X' = X(\lambda^*)$ is the

projection of A onto the line segment between X and Y^*, and (by convexity) this matrix lies in $\mathcal{T}_+^n$. In the final step we replace our candidate matrix X by X' and repeat the whole procedure for this new candidate.

The convergence of this Frank–Wolfe method for solving DA is assured by the following theorem (which follows from the general convergence theorem in Chapter 2 of [2]).

Theorem 4.1. *The DA algorithm produces a sequence of matrices* $\{X^{(k)}\}$ *that converges to* $\mathrm{Proj}_{\otimes}(A)$.

Note here, however, that the method is based on solving the subproblem (4) in each iteration. We discuss this subproblem in the next section.

5. The projection subproblem

The DA algorithm is based on solving the subproblem (4). This is the optimality test of the algorithm. We now discuss an approach to solving this problem.

Consider problem (4) for a given X (and A, of course). Letting $B = A - X$ and separating out the constant term $\langle B, X\rangle$ we are led to the problem

$$\eta(B) := \max\{\langle B, Y\rangle : Y \in \mathrm{Ext}(\mathcal{T}_+^{n,\otimes})\}. \tag{5}$$

Based on Theorem 2.2 we know that the extreme points of $\mathcal{T}_+^n$ are the rank one separable density matrices $(x \otimes y)(x \otimes y)^{\mathrm{T}}$ where $x \in \mathbb{R}^p$ and $y \in \mathbb{R}^q$ satisfy $\|x\| = \|y\| = 1$. So problem (5) becomes

$$\max\{g(x, y) : \|x\| = \|y\| = 1\},$$

where we define

$$g(x, y) = \langle B, (x \otimes y)(x \otimes y)^{\mathrm{T}}\rangle.$$

This function g is a multivariate polynomial of degree 4, i.e. it is a sum of terms of the form $c_{ijkl}x_i x_j y_k y_l$. One can verify that g may not be concave (or convex). Therefore it seems difficult to find a global maximum of g subject to the two given equality constraints. However, the function g has a useful decomposable structure in the two variables x and y which leads to a practical and fast algorithm for finding a local maximum of g.

The idea is to use a *block coordinate ascent* approach (also called the *non-linear Gauss–Seidel method*, see [2]) to the maximization of g. This iterative method consists in alternately fixing x and y and maximize with respect to the other variable. We now show that the corresponding subproblems (when either x or y is fixed) can be solved by eigenvalue methods.

First note that, by the mixed-product rule for tensor products,

$$Y = (x \otimes y)(x \otimes y)^{\mathrm{T}} = (xx^{\mathrm{T}}) \otimes (yy^{\mathrm{T}}) = \begin{bmatrix} Y_{11} & \cdots & Y_{1p} \\ \vdots & & \vdots \\ Y_{p1} & \cdots & Y_{pp} \end{bmatrix},$$

where $Y_{ij} = x_i x_j (yy^{\mathrm{T}}) \in \mathbb{R}^{q\times q}$ $(i, j \leqslant p)$. Partition the fixed matrix B conformly as

$$B = \begin{bmatrix} B_{11} & \cdots & B_{1p} \\ \vdots & & \vdots \\ B_{p1} & \cdots & B_{pp} \end{bmatrix},$$

where each B_{ij} is a $q \times q$ matrix. Note here that $B_{ij} = B_{ji}^{\mathrm{T}}$ $(i, j \leqslant p)$ as B is symmetric. With this block partitioning we calculate

$$\begin{aligned} g(x, y) &= \langle B, Y\rangle = \sum_{i,j\leqslant p} \langle B_{ij}, Y_{ij}\rangle = \sum_{i,j\leqslant p} \langle B_{ij}, x_i x_j (yy^{\mathrm{T}})\rangle \\ &= \sum_{i,j\leqslant p} x_i \langle B_{ij}, yy^{\mathrm{T}}\rangle x_j = x^{\mathrm{T}} \widetilde{B}(y) x, \end{aligned} \tag{6}$$

where $\widetilde{B}(y) = [\tilde{b}_{ij}(y)]$ is a $p \times p$ matrix with entries $\tilde{b}_{ij}(y) = \langle B_{ij}, yy^{\mathrm{T}}\rangle = y^{\mathrm{T}} B_{ij} y$. The matrix $\widetilde{B}(y)$ is symmetric.

Next we find another useful expression for $g(x, y)$.

$$\begin{aligned} g(x, y) &= x^{\mathrm{T}} \widetilde{B}(y) x = \sum_{i,j\leqslant p} \tilde{b}_{ij}(y) x_i x_j \\ &= \sum_{i,j\leqslant p} y^{\mathrm{T}} B_{ij} y \cdot x_i x_j = y^{\mathrm{T}} \left(\sum_{i,j\leqslant p} x_i x_j B_{ij} \right) y = y^{\mathrm{T}} \widehat{B}(x) y, \end{aligned} \tag{7}$$

where we define the matrix $\widehat{B}(x)$ by $\widehat{B}(x) = \sum_{i,j\leqslant p} x_i x_j B_{ij}$. Note that this matrix is symmetric as $B_{ij} = B_{ji}^{\mathrm{T}}$.

We now use these calculations to solve the mentioned subproblems where x respectively y is fixed.

Theorem 5.1. *The following equations hold*:

$$\begin{aligned} \eta(B) &= \max_{x,y} g(x, y) = \max\{\lambda_{\max}(\widetilde{B}(y)) : \|y\| = 1\} \\ &= \max\{\lambda_{\max}(\widehat{B}(x)) : \|x\| = 1\}. \end{aligned}$$

Moreover, for given x, $\max_y g(x, y)$ is attained by a normalized eigenvector of $\widehat{B}(x)$, and for fixed y, $\max_x g(x, y)$ is attained by a normalized eigenvector of $\widetilde{B}(y)$.

Proof. From Eqs. (6) and (7) we get

$$\begin{aligned} \eta(B) &= \max_{x,y} g(x, y) = \max_{\|y\|=1} \max_{\|x\|=1} x^{\mathrm{T}} \widetilde{B}(y) x \\ &= \max_{\|x\|=1} \max_{\|y\|=1} y^{\mathrm{T}} \widehat{B}(x) y. \end{aligned}$$

We now obtain the theorem from the following general fact: for every real symmetric matrix C we have that $\max_{\|z\|=1} z^{\mathrm{T}} C z = \lambda_{\max}(C)$ and that a maximizing z is a normalized eigenvector of C corresponding to $\lambda_{\max}(C)$. □

Due to this theorem the block coordinate ascent method applied to the projection subproblem (4) gives the following scheme.

Algorithm (*Eigenvalue maximization*).

1. Choose an initial vector y of length one.
2. Repeat the following two steps until convergence (or g no longer increases).
   2a. Let x be a normalized eigenvector corresponding to the largest eigenvalue of the matrix $\widetilde{B}(y)$.
   2b. Let y be a normalized eigenvector corresponding to the largest eigenvalue of the matrix $\widehat{B}(x)$.

We now comment on the convergence issues for this algorithm. The constructed sequence of vectors $\{(x^{(k)}, y^{(k)})\}$ must have a convergent subsequence. Moreover, the sequence $\{g(x^{(k)}, y^{(k)})\}$ is convergent. These facts follow from standard compactness/continuity arguments since the direct product of the unit balls is compact, g is continuous, and the sequence $\{g(x^{(k)}, y^{(k)})\}$ is non-decreasing. If we assume that each of the coordinate maxima found by the algorithm is unique (which seems hard to verify theoretically), then it is known that every limit point of $\{(x^{(k)}, y^{(k)})\}$ will be a local maximum of g (see Proposition 2.7.1 in [2]).

It should be remarked here that there are some remaining open questions concerning convergence of our method. However, in view of the hardness of the DA problem (shown to be NP-hard in [5]) one can expect that solving the projection subproblem (4) is also hard. We should therefore not expect anything more than local maxima in general, although we may be lucky to find a global maximum of g in certain cases. We refer to Section 7 for some preliminary computational results for our methods.

A final remark is that it may also be of interest to consider other numerical approaches to the problem of maximizing g than the one proposed here. We have not tried this since the described eigenvalue approach seems to work quite well.

6. Improvement of the DA algorithm

The DA algorithm, as described in Section 4, turns out to show very slow convergence. In this section we discuss a modification of the method which improves the convergence speed dramatically.

The mentioned slow convergence of the DA algorithm may be explained geometrically as follows. Assume that the given matrix A is non-separable, and that the current separable matrix X is not on the boundary of the set $\mathcal{T}_+^{n,\otimes}$ of separable matrices. To find a separable matrix closer to A, in this case, a good strategy would be to move in the direction $A–X$ until the boundary is reached. The algorithm moves instead in a direction $Y–X$ which is typically almost orthogonal to $A–X$, because the product matrix Y (an extreme point) will typically be far away from X.

The basic weakness of the algorithm is that from iteration k to $k+1$ it retains only the current best estimate X, throwing away all other information about X. An alternative approach, which turns out to allow a much faster convergence, is to retain all information, writing X explicitly as a convex combination of the previously generated product matrices Y_k,

$$X = \sum_{r=1}^{k} \lambda_r Y_r,$$

where $\lambda_r \geqslant 0$ and $\sum_r \lambda_r = 1$. After generating the next product matrix Y_{k+1} as a solution of the optimization problem (5) we find a new best convex combination varying all the coefficients λ_r, $r = 1, \ldots, k+1$.

An obvious modification of this scheme is to throw away in each iteration every Y_r getting a coefficient $\lambda_r = 0$. This means in practice that the number of product matrices retained does not grow too fast.

Thus, we are faced with the quadratic programming problem to minimize the squared distance $\|A - X\|^2$ as a quadratic polynomial in the coefficients λ_k. We have implemented a version of the conjugate gradient method (see [4]) for this problem. Theoretically, in the absence of rounding errors and inequality constraints, this method converges in a finite number of steps, and it also works well if the problem is degenerate, as is likely to happen here. The algorithm was adapted so it could handle the linear constraints $\lambda_i \geqslant 0$ for each i and $\sum_i \lambda_i = 1$, but we omit describing the implementation details here. (Several fast algorithms for quadratic optimization with linear inequality constraints are available.)

7. Computational results

In this section we present preliminary results for the modified DA algorithm as described in Section 6. Moreover we present some results and experiences with the eigenvalue maximization algorithm (see Section 5), and, finally, we discuss an application which may serve as a test of our methods.

Since we want to apply the DA algorithm to the problem of separability in quantum physics, we need to work with complex matrices. Therefore we have implemented and tested the complex version of the algorithm, rather than the real version as described above. As already remarked, the generalization is quite straightforward and is not expected to change in any essential way the performance of the algorithm. The main change is that matrix transposition has to be replaced by hermitian conjugation, and hence real symmetric matrices will in general become complex and hermitian. Thus, for example, the matrices $\widetilde{B}(y)$ and $\widehat{B}(x)$ both become hermitian, which means that their eigenvalues remain real, and the problem of finding the largest eigenvalue becomes no more difficult.

7.1. Eigenvalue maximization

We have tested the performance of the eigenvalue maximization algorithm on a number of randomly generated matrices, and on some special matrices used in other calculations. We ran the algorithm for each input matrix a number of times with random starting vectors x and y, comparing the maximum values and maximum points found.

For completely random matrices we often find only one maximum. Sometimes we find two maximum points with different maximum values. In certain symmetric cases it may happen that two or more maximum points have the same maximum value. Thus, we are not in general guaranteed to find a global maximum, but we always find a local maximum.

One possible measure of the speed of convergence of the algorithm is the absolute value of the scalar product $\langle x \otimes y, x' \otimes y' \rangle$, where x, y are input vectors and x', y' output vectors in one iteration. It takes typically about five iterations before this overlap is about 10^{-3} from unity (when $p = q = 3$), which means that the convergence to a local maximum is fast. The algorithm involves the diagonalization of one $p \times p$ matrix and one $q \times q$ matrix for each iteration, and in addition it may happen that more iterations are needed when the dimensions increase.

Table 1
Performance of the modified DA algorithm

p, q	$n = pq$	t (s)	Error
2	4	312	3×10^{-13}
3	9	155	3×10^{-12}
4	16	127	3×10^{-8}
5	25	773	1×10^{-6}
6	36	2122	5×10^{-6}
7	49	3640	1.0×10^{-5}
8	64	4677	1.5×10^{-5}
9	81	5238	2.2×10^{-5}
10	100	6566	3.5×10^{-5}

7.2. Results with the present program version

In Table 1 we show the performance of the modified DA algorithm for different dimensions of the problem. We take $p = q$, and we use the maximally entangled matrices in the given dimensions, as we know the distance to the closest separable matrix in these special cases. A maximally entangled matrix is a pure state $A = uu^{\mathrm{T}}$ (or $A = uu^{\dagger}$ if u is a complex vector), where

$$u = \frac{1}{\sqrt{p}} \sum_{i=1}^{p} e_i \otimes f_i$$

and where $\{e_i\}$ and $\{f_i\}$ are two sets of orthonormal basis vectors in $\mathbb{R}^p$ (or in $\mathscr{C}^p$). The closest separable state is $A' = \lambda A + (1 - \lambda)(1/p^2)I$ with $\lambda = 1/(p + 1)$, and the distance to it from A is $\sqrt{(p - 1)/(p + 1)}$. The density matrix $(1/n)I$, where I is the $n \times n$ unit matrix, is called the maximally mixed state.

The number of iterations of the main algorithm is set to 1000, and the fixed number of iterations used in the eigenvalue maximization algorithm (see Section 5) is set to 20. The tabulated time t is the total execution time on one computer, and the tabulated error is the difference between the calculated distance and the true distance.

The main conclusion we may draw is that the accuracy obtained for a fixed number of iterations decreases with increasing dimension. It should be noted that the rank one matrices used here are somewhat special, and that higher rank matrices give less good results. We want to emphasize that this is work in progress, and some fine tuning remains to be done. Nevertheless, we conclude at this stage that the method can be used for quite large matrices, giving useful results in affordable time.

7.3. An application

In the special cases $p = q = 2$ and $p = 2, q = 3$ there exists a simple necessary and sufficient criterion for separability of complex matrices (see [12,8]). We have checked our method against this criterion for $p = q = 2$.

Fig. 1 shows a two-dimensional cross section of the 15-dimensional space of complex 4×4 density matrices. The section is defined by three matrices: a maximally entangled matrix as described above, called a Bell matrix when $p = q = 2$; the maximally mixed state $(1/4)I$; and a rank one product matrix. The plot shows correct distances according to the Frobenius norm.

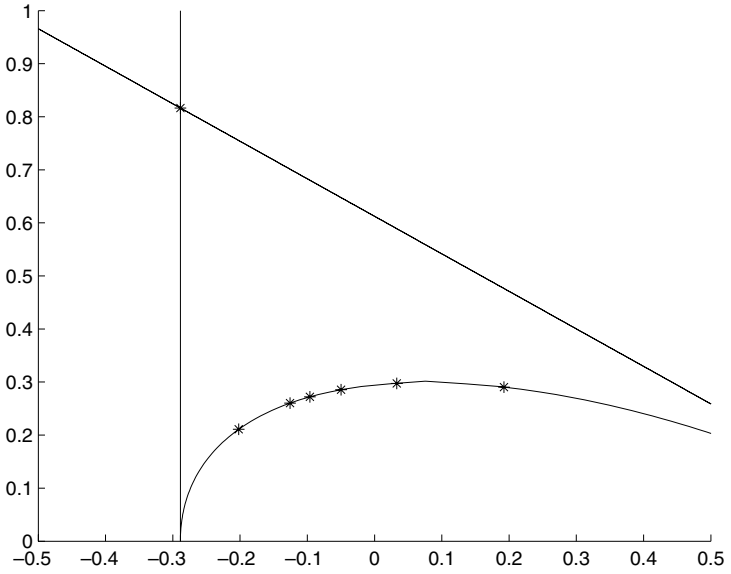


Fig. 1. The boundary of the set of separable matrices. The axes show distances by the Frobenius norm.

The algorithm finds the minimal distance from a given matrix A to a separable matrix. We choose an outside matrix A and gradually mix it with $(1/4)I$, until we find an $A' = \lambda A + (1 - \lambda)(1/4)I$, with $0 \leqslant \lambda \leqslant 1$, for which the distance is less than 5×10^{-5}. Then we plot A' as a boundary point.

When we start with A entangled, the closest separable matrix does not in general lie in the plane plotted here. Hence, we may certainly move in the plotting plane as much as the computed distance without crossing the boundary we are looking for. In this way we get very close to the boundary, approaching it from the outside, in very few steps.

In Fig. 1, the curved line is generated from the necessary, and in this case sufficient, condition for separability. All matrices below this line are separable, while the others are not. The six plotted boundary points are computed by our algorithm. The matrices to the right of the vertical straight line and below the skew straight line are positive definite, and the Bell matrix is located where the two lines cross. The maximally mixed state $(1/4)I$ is the origin of the plot.

Finally, we refer to the recent paper [10] for a further study of geometrical aspects of entanglement and applications of our algorithm in a study of bound entanglement in a composite system of two three-level systems.

Acknowledgments

The authors thank a referee for several useful comments and suggestions.

References

[1] D.P. Bertsekas, Convex Analysis and Optimization, Athena Scientific, 2003.
[2] D.P. Bertsekas, Nonlinear Programming, Athena Scientific, 1999.

[3] A.C. Doherty, P.A. Parillo, F.M. Spedalieri, Distinguishing separable and entangled states, Phys. Rev. Lett. 88 (2002) 187904.
[4] G.H. Golub, C.F. Van Loan, Matrix Computations, The John Hopkins University Press, Baltimore, 1993.
[5] L. Gurvits, Classical deterministic complexity of Edmonds' problem and quantum entanglement, in: Proceedings of the Thirty-Fifth ACM Symposium on Theory of Computing, ACM, New York, 2003, pp. 10–19.
[6] R.A. Horn, C.R. Johnson, Matrix Analysis, Cambridge University Press, 1991.
[7] R.A. Horn, C.R. Johnson, Topics in Matrix Analysis, Cambridge University Press, 1995.
[8] M. Horodecki, P. Horodecki, R. Horodecki, Separability of mixed states: necessary and sufficient conditions, Phys. Lett. A 223 (1996) 1.
[9] L.M. Ioannou, B.C. Travaglione, D. Cheung, A.K. Ekert, Improved algorithm for quantum separability and entanglement detection, Phys. Rev. A 70 (2004) 060303.
[10] J.M. Leinaas, J. Myrheim, E. Ovrum, Geometrical aspects of entanglement, Phys. Rev. A 74 (2006) 12313.
[11] M. Ozawa, Entanglement measures and the Hilbert–Schmidt distance, Phys. Lett. A 268 (2000) 158.
[12] A. Peres, Separability criterion for density matrices, Phys. Rev. Lett. 77 (1996) 1413.
[13] A.O. Pittenger, M.H. Rubin, Geometry of entanglement witnesses and local detection of entanglement, Phys. Rev. A 67 (2003) 012327.
[14] F. Verstraete, J. Dehaene, B. De Moor, On the geometry of entangled states, Jour. Mod. Opt. 49 (2002) 1277.
[15] R. Webster, Convexity, Oxford University Press, Oxford, 1994.

Paper III

Jon Magne Leinaas, Jan Myrheim and Eirik Ovrum,
Extreme points of the set of density matrices with positive partial transpose,
Submitted to PRL, arXiv:0704.3348v1 (2007)

Extreme points of the set of density matrices with positive partial transpose

J. M. Leinaas
Department of Physics, University of Oslo, N-0316 Oslo, Norway

J. Myrheim
Department of Physics, The Norwegian University of Science and Technology, 7491 Trondheim, Norway

E. Ovrum
Department of Physics and Center of Mathematics for Applications, University of Oslo, N-0316 Oslo, Norway
(Dated: April 24, 2007)

We present a necessary and sufficient condition for a finite dimensional density matrix to be an extreme point of the convex set of density matrices with positive partial transpose with respect to a subsystem. We also give an algorithm for finding such extreme points and illustrate this by some examples.

The density matrices describing states of a bipartite quantum system are uniquely classified as being either *separable* or *entangled*. However, it may be a non-trivial task in practice to classify a given density matrix. One simple method which can sometimes prove a density matrix to be entangled, and which is especially useful in systems with Hilbert spaces of low dimensions, is due to A. Peres [1]. Peres noted that a separable density matrix must remain positive when a partial transposition is performed with respect to one of the subsystems. In general the set of density matrices that have positive partial transpose, called PPT states for short, is larger than the set of separable matrices, but in low dimensions the difference between these two sets is small. In particular, for a system composed of one subsystem of dimension 2 and one subsystem of dimension 2 or 3, the two sets are known to be identical [2, 3].

A particular approach to the question of separability is to characterize geometrically the different sets of matrices as subsets of the real Hilbert space of hermitian matrices [4]. Thus, the set of all density matrices is a compact convex set $\mathcal{D}$, with the set of separable density matrices, $\mathcal{S}$, and the set of PPT matrices, $\mathcal{P}$, which we will call the *Peres set*, as convex subsets. The Peres criterion states that $\mathcal{S} \subset \mathcal{P} \subset \mathcal{D}$. We have presented elsewhere a numerical method for deciding whether a density matrix is included in $\mathcal{S}$ by computing the distance to the closest separable density matrix [5, 6].

There are two complementary ways to characterize a convex set. One way is to specify its *extreme points*. Thus, every point in a compact convex set of finite dimension d has an expansion as a convex combination of $d+1$ extreme points, while an extreme point cannot be written as a convex combination of other points in the set. The other way is to identify the set by conditions, typically algebraic equations and inequalities, that define whether or not a given point belongs to the set. For example, the convex set $\mathcal{D}$ may be defined by its extreme points, which are the pure states (one dimensional projections), or equivalently by the conditions of hermiticity, positivity and unit trace. In general there is no simple relation between these two descriptions of a convex set.

It is interesting to note that the convex sets $\mathcal{S}$ and $\mathcal{P}$ have simple charcterizations in the complementary ways just described. Thus, the extreme points of $\mathcal{S}$ are well known, they are the pure product states of the two subsystems, but there is no (known) effective test for membership of $\mathcal{S}$. In contrast, testing for membership of $\mathcal{P}$ is easy, because $\mathcal{P} = \mathcal{D} \cap \mathcal{D}^P$, where the superscript P denotes partial transposition, but the extreme points of $\mathcal{P}$ are not completely known.

The purpose of the present paper is to study the difference between the two sets $\mathcal{S}$ and $\mathcal{P}$ by studying the extreme points of $\mathcal{P}$. The extreme points of $\mathcal{S}$ are also extreme points of $\mathcal{P}$, because they are extreme points of $\mathcal{D}$ and $\mathcal{S} \subset \mathcal{P} \subset \mathcal{D}$. But in general $\mathcal{P}$ is larger than $\mathcal{S}$ and has additional extreme points, which are unknown and which make the whole difference between the two sets.

The extreme points of $\mathcal{P}$ that are not pure product states are interesting also as examples of entangled PPT states. In fact, because $\mathcal{S} \subset \mathcal{P}$, a separable extreme point of $\mathcal{P}$ must be an extreme point of $\mathcal{S}$ and hence a pure product state. It is easy to verify that any pure state which is not a product state, is not in $\mathcal{P}$ and is therefore entangled. Thus, every extreme point of $\mathcal{P}$ which is not a pure product state, is entangled and has rank higher than one. A stronger lower bound on the ranks of entangled PPT states is actually known [7].

We will in the following specify a criterion for uniquely identifying the extreme points of $\mathcal{P}$ and we will describe an algorithm for finding such points. The method is demonstrated by some examples.

To be more specific we consider a composite quantum system with Hilbert space $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B$ of finite dimension $N = N_A\, N_B$, where A and B denote the two subsystems. Then $\mathcal{D}$ is the set of density matrices on $\mathcal{H}$,

$$\rho \in \mathcal{D} \Leftrightarrow \rho = \rho^\dagger, \quad \rho \geq 0, \quad \mathrm{Tr}\,\rho = 1 \tag{1}$$

A subset is the Peres set $\mathcal{P}$,

$$\rho \in \mathcal{P} \Leftrightarrow \rho \in \mathcal{D} \text{ and } \rho^P \in \mathcal{D} \tag{2}$$

where ρ^P is the partial transpose of ρ with respect to any one of the subsystems, say system B. A subset of $\mathcal{P}$ is $\mathcal{S}$, the set

of separable density matrices,

$$\rho \in \mathcal{S} \Leftrightarrow \rho = \sum_k p_k \rho_k^A \otimes \rho_k^B, \quad p_k > 0, \quad \sum_k p_k = 1 \quad (3)$$

Thus, $\mathcal{S}$ is the *convex hull* of the product states of the two subsystems, and its extreme points are the *pure* product states.

To develop the method we need for finding the extreme points of $\mathcal{P}$ we will first apply it to $\mathcal{D}$, the full set of density matrices, with the pure states as extreme points. We recall some definitions and elementary facts.

Every $\rho \in \mathcal{D}$ is hermitian and has a spectral decomposition

$$\rho = \sum_i \lambda_i \, |\psi_i\rangle\langle\psi_i| \quad (4)$$

with real eigenvalues λ_i and orthonormal eigenvectors $|\psi_i\rangle$. The positivity condition $\rho \geq 0$ means that all $\lambda_i \geq 0$, or equivalently that $\langle\psi|\rho|\psi\rangle \geq 0$ for all $|\psi\rangle$, with $\langle\psi|\rho|\psi\rangle = 0$ if and only if $\rho|\psi\rangle = 0$. The orthogonal projection

$$P = \sum_{i,\lambda_i>0} |\psi_i\rangle\langle\psi_i| \quad (5)$$

projects onto the *image* (or *range*) of ρ, whereas $1-P$ projects onto the *kernel* of ρ, denoted by $\ker\rho$.

If ρ is not an extreme point of $\mathcal{D}$ it is a convex combination

$$\rho = x\rho' + (1-x)\rho'', \quad 0 < x < 1 \quad (6)$$

with $\rho', \rho'' \in \mathcal{D}$ and $\rho' \neq \rho''$. The identity

$$\langle\psi|\rho|\psi\rangle = x\,\langle\psi|\rho'|\psi\rangle + (1-x)\,\langle\psi|\rho''|\psi\rangle \quad (7)$$

shows that $\rho \geq 0$ when $\rho' \geq 0$ and $\rho'' \geq 0$, thus proving the convexity of $\mathcal{D}$. More interestingly, it shows that

$$\ker\rho = \ker\rho' \cap \ker\rho'' \quad (8)$$

With P defined as in (5) we have therefore $P\rho P = \rho$, $P\rho' P = \rho'$, and $P\rho'' P = \rho''$.

When (6) holds, the matrix $\sigma = \rho' - \rho$ is hermitian and nonzero, and $\mathrm{Tr}\,\sigma = 0$, hence σ has both positive and negative eigenvalues. Moreover, $P\sigma P = \sigma$. Define

$$\tau(x) = \rho + x\sigma \quad (9)$$

for x real. Since σ has both positive and negative eigenvalues, so has $\tau(x)$ for large enough $|x|$. If $\rho|\psi\rangle = 0$ then $P|\psi\rangle = 0$ and $\sigma|\psi\rangle = P\sigma P|\psi\rangle = 0$, hence $\tau(x)|\psi\rangle = 0$ for all x. Therefore only the strictly positive eigenvalues of ρ can change when $x\sigma$ is added to ρ, and since they change continuously with x, they remain positive for x in a finite interval about $x = 0$.

We conclude that there exists an $x_1 < 0$ and an $x_2 > 0$ such that $\tau(x) \geq 0$ for $x_1 \leq x \leq x_2$, and $\tau(x) \not\geq 0$ for $x < x_1$ or $x > x_2$. At $x = x_1$ or $x = x_2$, $\tau(x)$ has at least one zero eigenvalue more than ρ.

We are now prepared to search systematically for extreme points of $\mathcal{D}$. Starting with an arbitrary $\rho_1 \in \mathcal{D}$ we define the projection P_1 in the same way as we defined P from ρ in (5). If we can find a hermitian matrix σ solving the equation

$$P_1\sigma P_1 = \sigma \quad (10)$$

we define

$$\sigma_1 = \sigma - (\mathrm{Tr}\,\sigma)\,\rho_1 \quad (11)$$

in order to have $P_1\sigma_1 P_1 = \sigma_1$ and $\mathrm{Tr}\,\sigma_1 = 0$. Clearly $\sigma = \rho_1$ is a solution of (10). If this is the only solution, then only $\sigma_1 = 0$ is possible, and it follows from the above discussion that ρ_1 is an extreme point. The number of linearly independent solutions of (10) is ${n_1}^2$, where $n_1 = \mathrm{Tr}\,P_1$ is the rank of ρ_1. Hence, ρ_1 is an extreme point if and only if $n_1 = 1$ so that it is a pure state.

If ρ_1 is not an extreme point, then we can find $\sigma_1 \neq 0$ and define

$$\tau_1(x) = \rho_1 + x\sigma_1 \quad (12)$$

We increase (or decrease) x from $x = 0$ until it first happens that $\tau_1(x)$ gets one or more additional zero eigenvalues as compared to ρ_1. We will know if we go too far, because then $\tau_1(x)$ will get negative eigenvalues. We choose ρ_2 as the limiting $\tau_1(x)$ determined in this way. By construction, ρ_2 has lower rank than ρ_1.

We repeat the whole procedure with ρ_2 in place of ρ_1, and if ρ_2 is not extreme we will find a ρ_3 of lower rank. Continuing in the same way, we must end up at an extreme point ρ_K, with $K \leq N$, since we get a decreasing sequence of projections,

$$I \supseteq P_1 \supset P_2 \supset \ldots \supset P_K \quad (13)$$

of decreasing ranks $N \geq n_1 > n_2 > \ldots > n_K = 1$.

We may understand the above construction geometrically. In fact, each projection P_k defines a convex subset $P_k\mathcal{D}P_k$ of $\mathcal{D}$, with ρ_k as an interior point. This subset consists of all density matrices on the n_k dimensional Hilbert space $P_k\mathcal{H}$, and if $n_k < N$ it is a flat face of the boundary of $\mathcal{D}$.

It is straightforward to adapt the above method and use it to search for extreme points of $\mathcal{P}$. Thus, we consider an initial matrix $\rho_1 \in \mathcal{P}$, characterized by two integers (n_1, m_1), the ranks of ρ_1 and of the partial transpose ρ_1^P, since $\rho_1 \in \mathcal{P}$ means that $\rho_1 \in \mathcal{D}$ and $\rho_1^P \in \mathcal{D}$. We denote by P_1 the projection on the image of ρ_1, as before, and we introduce Q_1 as the projection on the image of ρ_1^P. We have to solve the two equations

$$P_1\sigma P_1 = \sigma\,, \quad Q_1\sigma^P Q_1 = \sigma^P \quad (14)$$

To understand these equations it may help to think of σ as a vector in the N^2 dimensional real Hilbert space $\mathcal{M}$ of $N \times N$ hermitian matrices with the scalar product

$$\langle A, B\rangle = \langle B, A\rangle = \mathrm{Tr}(AB) = \sum_{i,j} A_{ij}^* \, B_{ij} \quad (15)$$

If $\mathbf{L}$ is a linear transformation on $\mathcal{M}$, its transpose $\mathbf{L}^T$ is defined by the identity $\langle A, \mathbf{L}^T B\rangle = \langle \mathbf{L}A, B\rangle$, and $\mathbf{L}$ is symmetric if $\mathbf{L}^T = \mathbf{L}$. Partial transposition of $A \in \mathcal{M}$ permutes the matrix elements of A and is a linear transformation $\mathbf{\Pi}A = A^P$. It is its own inverse, and is an orthogonal transformation (it preserves the scalar product), hence $\mathbf{\Pi} = \mathbf{\Pi}^{-1} = \mathbf{\Pi}^T$. It also preserves the trace, $\mathrm{Tr}(\mathbf{\Pi}A) = \mathrm{Tr}\, A$. The projections P_1 and Q_1 on $\mathcal{H}$ define projections $\mathbf{P}_1$ and $\mathbf{Q}_1$ on $\mathcal{M}$ by

$$\mathbf{P}_1 A = P_1 A P_1\,, \quad \mathbf{Q}_1 A = Q_1 A Q_1 \tag{16}$$

These are both orthogonal projections: ${\mathbf{P}_1}^2 = \mathbf{P}_1$, $\mathbf{P}_1^T = \mathbf{P}_1$, ${\mathbf{Q}_1}^2 = \mathbf{Q}_1$, and $\mathbf{Q}_1^T = \mathbf{Q}_1$

In this language the equations (14) may be written as

$$\mathbf{P}_1 \sigma = \sigma\,, \quad \bar{\mathbf{Q}}_1 \sigma = \sigma \tag{17}$$

with $\bar{\mathbf{Q}}_1 = \mathbf{\Pi}\mathbf{Q}_1\mathbf{\Pi}$. Note that $\bar{\mathbf{Q}}_1$ is also an orthogonal projection: $\bar{\mathbf{Q}}_1^{\,2} = \bar{\mathbf{Q}}_1$ and $\bar{\mathbf{Q}}_1^T = \bar{\mathbf{Q}}_1$. These two equations are equivalent to the single equation

$$\mathbf{P}_1 \bar{\mathbf{Q}}_1 \mathbf{P}_1 \sigma = \sigma \tag{18}$$

or equivalently $\bar{\mathbf{Q}}_1 \mathbf{P}_1 \bar{\mathbf{Q}}_1 \sigma = \sigma$. They restrict the hermitian matrix σ to the intersection between the two subspaces of $\mathcal{M}$ defined by the projections $\mathbf{P}_1$ and $\bar{\mathbf{Q}}_1$. We shall denote by $\mathbf{B}_1$ the projection on this subspace, which is spanned by the eigenvectors with eigenvalue 1 of $\mathbf{P}_1 \bar{\mathbf{Q}}_1 \mathbf{P}_1$. Because the latter is a symmetric linear transformation on $\mathcal{M}$ it has a complete set of orthonormal real eigenvectors and eigenvalues. All its eigenvalues lie between 0 and 1. We may diagonalize it in order to find its eigenvectors with eigenvalue 1.

Having found σ as a solution of (18) we define σ_1 as in (11). If $\sigma = \rho_1$ and $\sigma_1 = 0$ is the only solution, then ρ_1 is an extreme point of $\mathcal{P}$. If we can find $\sigma_1 \neq 0$, then we define $\tau_1(x)$ as in (12), and increase (or decrease) x from $x = 0$ until we reach the first value of x where either $\tau_1(x)$ or $(\tau_1(x))^P$ has at least one new zero eigenvalue. This special $\tau_1(x)$ we take as ρ_2. By construction, when n_2 is the rank of ρ_2 and m_2 the rank of ρ_2^P, we have $n_2 \leq n_1$, $m_2 \leq m_1$, and either $n_2 < n_1$ or $m_2 < m_1$.

Next, we check whether ρ_2 is an extreme point of $\mathcal{P}$, in the same way as with ρ_1. If ρ_2 is not extreme, then we can find a third candidat ρ_3, and so on. We will reach an extreme point ρ_K in a finite number of iterations, because the sum of ranks, $n_k + m_k$, decreases in each iteration.

The geometrical interpretation of this iteration scheme is that the density matrices $\rho \in \mathcal{P}$ which satisfy the condition $\mathbf{B}_k\, \rho = \rho$ define a convex subset of $\mathcal{P}$ having ρ_k as an interior point. This subset is either the whole of $\mathcal{P}$, or a flat face of the boundary of $\mathcal{P}$, which is the intersection of a flat face of $\mathcal{D}$ and a flat face of $\mathcal{D}^P$.

The construction discussed above defines an algorithm for finding extreme points of $\mathcal{P}$, and gives at the same time a necessary and sufficient condition for a density matrix in $\mathcal{P}$ to be an extreme point. Let us restate this condition:

A density matrix $\rho \in \mathcal{P}$ is an extreme point of $\mathcal{P}$ if and only if the projection P which projects on the image of ρ and the projection Q which projects on the image of ρ^P, define a combined projection $\mathbf{B}$ of rank 1 in the real Hilbert space $\mathcal{M}$ of hermitian matrices.

The algorithm deserves some further comments. In iteration k the projections P_k, Q_k and $\mathbf{B}_k$ are uniquely defined by the density matrix ρ_k, but the matrix σ_k is usually not unique. In the applications discussed below we have simply chosen σ_k randomly. Clearly, more systematic choices are possible if one wants to search for special types of extreme points.

Another comment concerns the ranks (n, m) of a density matrix ρ and its partial transpose ρ^P if it is an extreme point. We can derive an upper limit on these ranks. The rank of the projection $\mathbf{P}$ is n^2 and the rank of $\bar{\mathbf{Q}}$ is m^2, thus, the equations $\mathbf{P}\sigma = \sigma$ and $\bar{\mathbf{Q}}\sigma = \sigma$ for the N^2 dimensional vector σ represent $N^2 - n^2$ and $N^2 - m^2$ constraints, respectively. The total number of *independent* constraints is $n_c \leq 2N^2 - n^2 - m^2$, and the rank of $\mathbf{B}$ is $N^2 - n_c \geq n^2 + m^2 - N^2$. For an extreme point the rank of $\mathbf{B}$ is 1, implying the inequality

$$n^2 + m^2 \leq N^2 + 1 \tag{19}$$

We have used our algorithm in numerical studies. In one approach we use as initial density matrix the maximally mixed state $\rho_1 = 1/N$ and choose in iteration k a random direction in the subspace $\mathbf{B}_k\mathcal{M}$. We list in Table 1 all the ranks of extreme points found in this way in various dimensions $N = N_A N_B$. We do not distinguish between ranks (n, m) and (m, n) since there is full symmetry between ρ and ρ^P.

$N_A \times N_B = N$	(n, m)		$n + m$
$2 \times 4 = 8$	(5,6)		11
$3 \times 3 = 9$	(6,6)	(5,7)	12
$2 \times 5 = 10$	(7,7)	(6,8)	14
$2 \times 6 = 12$	(8,9)		17
$3 \times 4 = 12$	(8,9)		17
$3 \times 5 = 15$	(10,11)		21
$4 \times 4 = 16$	(11,11)	(10,12)	22
$3 \times 6 = 18$	(12,13)		25
$4 \times 5 = 20$	(14,14)	(13,15)	28
$5 \times 5 = 25$	(17,18)		35

TABLE I: Typical ranks (n, m) for extreme points

We find only solutions of *maximal rank*, in the sense that increasing either n or m will violate the inequality (19). Maximal rank means that the constraints given by the equations $\mathbf{P}\sigma = \sigma$ and $\bar{\mathbf{Q}}\sigma = \sigma$ are mostly independent. Furthermore, we find only the most symmetric ranks, in the sense that $m \approx n$. For example, in the 4×4 system we find ranks $(11, 11)$ and $(10, 12)$, but not $(9, 13)$, $(7, 14)$ or $(5, 15)$, which are also maximal.

The 3×3 system we have examined further in the following way. For one specific sequence $\rho_1, \rho_2, \ldots, \rho_K$ with ρ_K extreme, we repeat the final step, keeping ρ_{K-1} fixed but choosing different directions in the subspace $\mathbf{B}_{K-1}\mathcal{M}$. We find

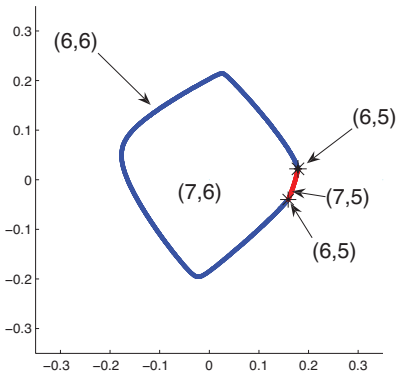


FIG. 1: Section through the boundary of $\mathcal{P}$, in 3×3 dimensions. A closed curve of extreme points surrounds a region of entangled PPT matrices of rank $(7, 6)$. The curve has two parts, characterized by ranks $(7, 5)$ and $(6, 6)$, joining at two rank $(6, 5)$ extreme points.

that every direction points directly towards an extreme point. Thus, ρ_{K-1} is an interior point of a flat face of the boundary of $\mathcal{P}$ bounded by a hypersurface of extreme points. Fig. 1 shows a two dimensional section through this flat face.

This shows that extreme points of non-maximal rank do exist, and the fact that we do not find them in random searches just indicates that they define subsets of lower dimension than the extreme points of maximal rank (n, m) with $m \approx n$. Note that an extreme point which is not a pure product state cannot have arbitrarily low rank, since in [7] there is a proof that all PPT matrices of rank less than $N_0 \equiv \min\{N_A, N_B\}$ are separable. This implies a lower limit of (N_0, N_0) for the ranks of an extreme point of $\mathcal{P}$ which is not a pure product state. It is not known whether there exist entangled PPT states of the minimal rank N_0.

Several of the low rank entangled PPT states that are known turn out, in our test, to be extreme points. Examples in 3×3 dimensions include the unextendible product basis state of rank $(4, 4)$ [8]; another state of rank $(4, 4)$ [9]; and explicit examples of rank $(5, 5)$ and $(6, 6)$ states [10].

The entangled PPT states first discovered [3], in 3×3 dimensions to be specific, are not extreme points of $\mathcal{P}$, but on the flat face defined by the corresponding projection $\mathbf{B}$ they seem to be completely surrounded by extreme points. Figure 2 is a two dimensional section chosen so as to show one such state (with parameter value $a = 0.42$, called here the "Horodecki state"), as a convex combination of two extreme points of $\mathcal{P}$. We would expect a two dimensional section through two extreme points of $\mathcal{P}$ to show maximum difference between the sets $\mathcal{S}$ and $\mathcal{P}$. Thus, this plot illustrates the fact that the difference is indeed very small in 3×3 dimensions.

In conclusion, the method discussed has the potential of producing a clearer picture of the difference between the two sets $\mathcal{S}$ and $\mathcal{P}$ and thereby the set of states with *bound entanglement*. We intend to follow up the work presented in this paper by other numerical studies of composite systems of low Hilbert space dimensions.

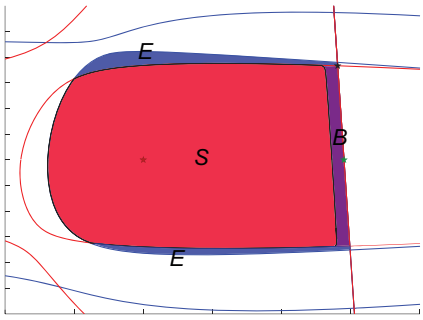


FIG. 2: Section through the set of density matrices in 3×3 dimensions. The star in the middle of a straight line is the "Horodecki state" (see text). It is a convex combination of two extreme points, one of which is plotted as a star. The maximally mixed state $1/N$ is the star close to the center. The separable matrices lie in the large (red) region marked S, the entangled PPT states in the (purple) region marked B, and the entangled non-PPT states in the two (blue) regions marked E. The lines are solutions of the equations $\det \rho = 0$ (blue) and $\det \rho^P = 0$ (red lines).

This work has been supported by NordForsk.

[1] A. Peres, *Separability criterion for density matrices*, Phys. Rev. Lett. **77**, 1413 (1996).
[2] M. Horodecki, P. Horodecki, and R. Horodecki, *Separability of mixed states: necessary and sufficient conditions*, Phys. Lett. A **223**, 1 (1996).
[3] P. Horodecki, *Separability criterion and inseparable mixed states with positive partial transposition*, Phys. Lett. A **232**, 333 (1997).
[4] I. Bengtsson and K. Zyczkowski, *Geometry of quantum states*, Cambridge University Press (2006).
[5] J. M. Leinaas, J. Myrheim, and E. Ovrum, *Geometrical aspects of entanglement*, Phys. Rev. A **74**, 012313 (2006).
[6] G. Dahl, J. M. Leinaas, J. Myrheim, and E. Ovrum, *A tensor product matrix approximation problem in quantum physics*, Linear Algebra and its Applications **420**, 711 (2007).
[7] P. Horodecki, J. Smolin, B. Terhal, and A. Thapliyal, *Rank two bipartite bound entangled states do not exist*, Theoretical Computer Science **292**, 589 (2003).
[8] C. Bennet, D. DiVincenzo, T. Mor, P. Shor, J. Smolin, and B. Terhal, *Unextendible product bases and bound entanglement* Phys. Rev. Lett. **82**, 5385 (1999).
[9] K. Ha, S. Kye, and Y. Park, *Entangled states with positive partial transposes arising from indecomposable positive linear maps*, Phys. Lett. A **313**, 163 (2003).
[10] L. Clarisse, *Construction of bound entangled edge states with special ranks*, Phys. Lett. A **359**, 603 (2006).

Paper IV

Morten Hjorth-Jensen and Eirik Ovrum,
Quantum computation algorithm for many-body studies,
arXiv:0705.1928v1 (2007)

Quantum computation algorithm for many-body studies

E. Ovrum and M. Hjorth-Jensen
Department of Physics and Center of Mathematics for Applications, University of Oslo, N-0316 Oslo, Norway
(Dated: May 14, 2007)

We show in detail how the Jordan-Wigner transformation can be used to simulate any fermionic many-body Hamiltonian on a quantum computer. We develop an algorithm based on appropriate qubit gates that takes a general fermionic Hamiltonian, written as products of a given number of creation and annihilation operators, as input. To demonstrate the applicability of the algorithm, we calculate eigenvalues and eigenvectors of two model Hamiltonians, the well-known Hubbard model and a generalized pairing Hamiltonian. Extensions to other systems are discussed.

I. INTRODUCTION

A theoretical understanding of the behavior of many-body systems is a great challenge and provides fundamental insights into quantum mechanical studies, as well as offering potential areas of applications. However, apart from some few analytically solvable problems, the typical absence of an exactly solvable contribution to the many-particle Hamiltonian means that we need reliable numerical many-body methods. These methods should allow for controlled approximations and provide a computational scheme which accounts for successive many-body corrections in a systematic way. Typical examples of popular many-body methods are coupled-cluster methods [1, 2, 3], various types of Monte Carlo methods [4, 5, 6], perturbative expansions [7, 8], Green's function methods [9, 10], the density-matrix renormalization group [11, 12], ab initio density functional theory [13, 14, 15] and large-scale diagonalization methods [16, 17, 18, 19].

However, all these methods have to face in some form or the other the problem of an exponential growth in dimensionality. For a system of P fermions which can be placed into N levels, the total number of basis states are given by $\left(\begin{array}{c} N \\ P \end{array}\right)$. The dimensional curse means that most quantum mechanical calculations on classical computers have exponential complexity and therefore are very hard to solve for larger systems. On the other hand, a so-called quantum computer, a particularly dedicated computer, can improve greatly on the size of systems that can be simulated, as foreseen by Feynman [22, 23]. A quantum computer does not need an exponential amount of memory to represent a quantum state. The basic unit of information for a quantum computer is the so-called qubit or quantum bit. Any suitable two-level quantum system can be a qubit, but the standard model of quantum computation is a model where two-level quantum systems are located at different points in space, and are manipulated by a small universal set of operations. These operations are called gates in the same fashion as operations on bits in classical computers are called gates.

For the example of P spin 1/2 particles, a classical computer needs 2^P bits to represent all possible states, while a quantum computer needs only P qubits. The complexity in number of qubits is thus linear. Based on these ideas, several groups have proposed various algorithms for simulating quantal many-body systems on quantum computers. Abrams and Lloyd, see for example Refs. [20, 21], introduced a quantum algorithm that uses the quantum fast Fourier transform to find eigenvalues and eigenvectors of a given Hamiltonian, illustrating how one could solve classically intractable problems with less than 100 qubits. Achieving a polynomial complexity in the number of operations needed to simulate a quantum system is not that straightforward however. To get efficient simulations in time one needs to transform the many-body Hamiltonian into a sum of operations on qubits, the building blocks of the quantum simulator and computer, so that the time evolution operator can be implemented in polynomial time. In Refs. [24, 25, 26] it was shown how the Jordan-Wigner transformation in principle does this for all Hamiltonians acting on fermionic many-body states. Based on this approach, recently two groups, see Refs. [27, 28], published results where they used Nuclear Magnetic Resonance (NMR) qubits to simulate the pairing Hamiltonian.

The aim of this work is to develop an algorithm than allows one to perform a quantum computer simulation (or simply quantum simulation hereafter) of any many-body fermionic Hamiltonian. We show how to generate, via various Jordan-Wigner transformations, all qubit operations needed to simulate the time evolution operator of a given Hamiltonian. We also show that for a given term in an m-body fermionic Hamiltonian, the number of operations needed to simulate it is linear in the number of qubits or energy-levels of the system. The number of terms in the Hamiltonian is of the order of m^2 for a general m-body interaction, making the simulation increasingly harder with higher order interactions. We specialize our examples to a two-body Hamiltonian, since this is also the most general type of Hamiltonian encountered in many-body physics. Besides fields like nuclear physics, where three-body forces play a non-neglible role, a two-body Hamiltonian captures most of the relevant physics. The various transformations are detailed in the next section. In Sec. III we show in detail how to simulate a quantum computer finding the eigenvalues of any two-body Hamiltonian, with all available particle numbers, using the simulated time evolution operator. In that section we describe also the techniques which are necessary for the extraction of information using a phase-estimation

algorithm.

To demonstrate the feasibility of our algorithm, we present in Sec. IV selected results from applications of our algorithm to two simple model-Hamiltonians, a pairing Hamiltonian and the Hubbard model. We summarize our results and present future perspectives in Sec. V.

II. ALGORITHM FOR QUANTUM COMPUTATIONS OF FERMIONIC SYSTEMS

A. Hamiltonians

A general two-body Hamiltonian for fermionic system can be written as

$$H = E_0 + \sum_{ij=1} E_{ij} a_i^\dagger a_j + \sum_{ijkl=1} V_{ijkl} a_i^\dagger a_j^\dagger a_l a_k, \qquad (1)$$

where E_0 is a constant energy term, E_{ij} represent all the one-particle terms, allowing for non-diagonal terms as well. The one-body term can represent a chosen single-particle potential, the kinetic energy or other more specialized terms such as those discussed in connection with the Hubbard model [29] or the pairing Hamiltonian discussed below. The two-body interaction part is given by V_{ijkl} and can be any two-body interaction, from Coulomb interaction to the interaction between nucleons. The sums run over all possible single-particle levels N. Note that this model includes particle numbers from zero to the number of available quantum levels, n. To simulate states with fixed numbers of fermions one would have to either rewrite the Hamiltonian or generate specialized input states in the simulation.

The algorithm which we will develop in this section and in Sec. III can treat any two-body Hamiltonian. However, in our demonstrations of the quantum computing algorithm, we will limit ourselves to two simple models, which however capture much of the important physics in quantum mechanical many-body systems. We will also limit ourselves to spin $j = 1/2$ systems, although our algorithm can also simulate higher j-values, such as those which occur in nuclear, atomic and molecular physics, it simply uses one qubit for every available quantum state. These simple models are the Hubbard model and a pairing Hamiltonian. We start with the spin 1/2 Hubbard model, described by the following Hamiltonian

$$\begin{aligned} H_H &= \epsilon \sum_{i,\sigma} a_{i\sigma}^\dagger a_{i\sigma} - t \sum_{i,\sigma} \left(a_{i+1,\sigma}^\dagger a_{i,\sigma} + a_{i,\sigma}^\dagger a_{i+1,\sigma} \right) \\ &\quad + U \sum_{i=1} a_{i+}^\dagger a_{i-}^\dagger a_{i-} a_{i+}, \end{aligned} \qquad (2)$$

where $a^\dagger$ and a are fermion creation and annihilation operators, respectively. This is a chain of sites where each site has room for one spin up fermion and one spin down fermion. The number of sites is N, and the sums over σ are sums over spin up and down only. Each site has a single-particle energy ϵ. There is a repulsive term U if there is a pair of particles at the same site. It is energetically favourable to tunnel to neighbouring sites, described by the hopping terms with coupling constant $-t$.

The second model-Hamiltonian is the simple pairing Hamiltonian

$$H_P = \sum_i \varepsilon_i a_i^\dagger a_i - \frac{1}{2} g \sum_{ij>0} a_i^\dagger a_{\bar{i}}^\dagger a_{\bar{j}} a_j, \qquad (3)$$

The indices i and j run over the number of levels N, and the label $\bar{\imath}$ stands for a time-reversed state. The parameter g is the strength of the pairing force while ε_i is the single-particle energy of level i. In our case we assume that the single-particle levels are equidistant (or degenerate) with a fixed spacing d. Moreover, in our simple model, the degeneracy of the single-particle levels is set to $2j+1 = 2$, with $j = 1/2$ being the spin of the particle. This gives a set of single-particle states with the same spin projections as for the Hubbard model. Whereas in the Hubbard model we operate with different sites with spin up or spin down particles, our pairing models deals thus with levels with double degeneracy. Introducing the pair-creation operator $S_i^+ = a_{im}^\dagger a_{i-m}^\dagger$, one can rewrite the Hamiltonian in Eq. (3) as

$$H_P = d \sum_i i N_i + \frac{1}{2} G \sum_{ij>0} S_i^+ S_j^-,$$

where $N_i = a_i^\dagger a_i$ is the number operator, and $\varepsilon_i = id$ so that the single-particle orbitals are equally spaced at intervals d. The latter commutes with the Hamiltonian H. In this model, quantum numbers like seniority $\mathcal{S}$ are good quantum numbers, and the eigenvalue problem can be rewritten in terms of blocks with good seniority. Loosely speaking, the seniority quantum number $\mathcal{S}$ is equal to the number of unpaired particles; see [30] for further details. Furthermore, in a series of papers, Richardson, see for example Refs. [31, 32, 33], obtained the exact solution of the pairing Hamiltonian, with semi-analytic (since there is still the need for a numerical solution) expressions for the eigenvalues and eigenvectors. The exact solutions have had important consequences for several fields, from Bose condensates to nuclear superconductivity and is currently a very active field of studies, see for example Refs. [34, 35]. Finally, for particle numbers up to $P \sim 20$, the above model can be solved exactly through numerical diagonalization and one can obtain all eigenvalues. It serves therefore also as an excellent ground for comparison with our algorithm based on models from quantum computing.

B. Basic quantum gates

Benioff showed that one could make a quantum mechanical Turing machine by using various unitary opera-

tions on a quantum system, see Ref. [36]. Benioff demonstrated that a quantum computer can calculate anything a classical computer can. To do this one needs a quantum system and basic operations that can approximate all unitary operations on the chosen many-body system. We describe in this subsection the basic ingredients entering our algorithms.

1. Qubits, gates and circuits

In this article we will use the standard model of quantum information, where the basic unit of information is the qubit, the quantum bit. As mentioned in the introduction, any suitable two-level quantum system can be a qubit, it is the smallest system there is with the least complex dynamics. Qubits are both abstract measures of information and physical objects. Actual physical qubits can be ions trapped in magnetic fields where lasers can access only two energy levels or the nuclear spins of some of the atoms in molecules accessed and manipulated by an NMR machine. Several other ideas have been proposed and some tested, see [37].

The computational basis for one qubit is $|0\rangle$ (representing for example bit 0) for the first state and $|1\rangle$ (representing bit 1) for the second, and for a set of qubits the tensor products of these basis states for each qubit form a product basis. Below we write out the different basis states for a system of n qubits.

$$\begin{aligned} |0\rangle &\equiv |00\cdots 0\rangle = |0\rangle\otimes|0\rangle\otimes\cdots\otimes|0\rangle \\ |1\rangle &\equiv |00\cdots 1\rangle = |0\rangle\otimes|0\rangle\otimes\cdots\otimes|1\rangle \\ &\vdots \\ |2^n-1\rangle &\equiv |11\cdots 1\rangle = |1\rangle\otimes|1\rangle\otimes\cdots\otimes|1\rangle. \end{aligned} \tag{4}$$

This is a 2^n-dimensional system and we number the different basis states using binary numbers corresponding to the order in which they appear in the tensor product.

Quantum computing means to manipulate and measure qubits in such a way that the results from a measurement yield the solutions to a given problem. The quantum operations we need to be able to perform our simulations are a small set of elementary single-qubit operations, or single-qubit gates, and one universal two-qubit gate, in our case the so-called CNOT gate defined below.

To represent quantum computer algorithms graphically we use circuit diagrams. In a circuit diagram each qubit is represented by a line, and operations on the different qubits are represented by boxes. In fig. 1 we show an example of a quantum circuit, with the arrow indicating the time evolution, The states $|a\rangle$ and $|b\rangle$ in the figure represent qubit states. In general, the total state will be a superposition of different qubit states. A single-qubit gate is an operation that only affects one physical qubit, for example one ion or one nuclear spin in a molecule. It

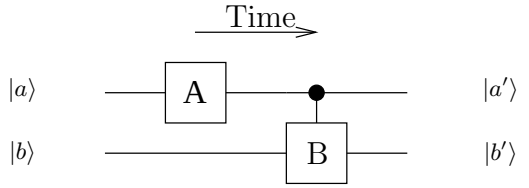


FIG. 1: A quantum circuit showing a single-qubit gate A and a two-qubit gate acting on a pair of qubits, represented by the horizontal lines.

is represented by a box on the line corresponding to the qubit in question. A single-qubit gate operates on one qubit and is therefore represented mathematically by a 2×2 matrix while a two-qubit gate is represented by a 4×4 matrix. As an example we can portray the so-called CNOT gate as matrix,

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{5}$$

This is a very important gate, since one can show that it behaves as a universal two-qubit gate, and that we only need this two-qubit gate and a small set of single-qubit gates to be able to approximate any multi-qubit operation. One example of a universal set of single-qubit gates is given in Fig. 2. The products of these three operations on one qubit can approximate to an arbitrary precision any unitary operation on that qubit.

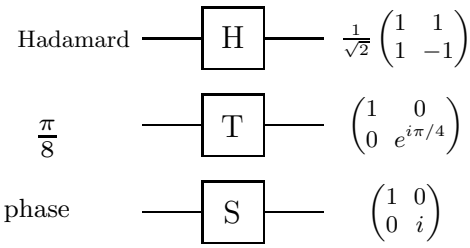


FIG. 2: Set of three elementary single-qubit gates and their matrix representations. The products of these three operations on one qubit can approximate to an arbitrary precision any unitary operation on that qubit.

2. Decomposing unitary operations into gates

The next step is to find elementary operations on a set of qubits that can be put together in order to approximate any unitary operation on the qubits. In this way we can perform computations on a quantum computer by performing many of these elementary operations in the correct order.

There are three steps in finding the elementary operations needed to simulate any unitary operation. First,

any $d \times d$ unitary matrix can be factorized into a product of at most $d(d-1)/2$ two-level unitary matrices, see for example Ref. [37] for details. A two-level unitary matrix is a matrix that only acts non-trivially on two vector components when multiplied with a vector. For all other vector components it acts as the identity operation.

The next step is to prove that any two-level unitary matrix can be implemented by one kind of two-qubit gate, for example the CNOT gate in Eq. (5), and single-qubit gates only. This simplifies the making of actual quantum computers as we only need one type of interaction between pairs of qubits. All other operations are operations on one qubit at the time.

Finally, these single-qubit operations can be approximated to an arbitrary precision by a finite set of single-qubit gates. Such a set is called a universal set and one example is the phase gate, the so-called Hadamard gate and the $\pi/8$ gate. Fig. 2 shows these gates. By combining these properly with the CNOT gate one can approximate any unitary operation on a set of qubits.

3. Quantum calculations

The aspect of quantum computers we are focusing on in this article is their use in computing properties of quantum systems. When we want to use a quantum computer to find the energy levels of a quantum system or simulate it's dynamics, we need to simulate the time evolution operator of the Hamiltonian, $U = \exp(-iH\Delta t)$. To do that on a quantum computer we must find a set of single- and two-qubit gates that would implement the time evolution on a set of qubits. For example, if we have one qubit in the state $|a\rangle$, we must find the single-qubit gates that when applied results in the qubit being in the state $\exp(-iH\Delta t)|a\rangle$.

From what we have written so far the naive way of simulating U would be to calculate it directly as a matrix in an appropriate basis, factorize it into two-level unitary matrices and then implement these by a set of universal gates. In a many-body fermion system for example, one could use the Fock basis to calculate U as a matrix. A fermion system with n different quantum levels can have from zero to n particles in each Fock basis state. A two-level system has four different basis states, $|00\rangle$, $|01\rangle$, $|10\rangle$ and $|11\rangle$, where $|0\rangle$ corresponds to an occupied quantum level. The time evolution matrix is then a $2^n \times 2^n$ matrix. This matrix is then factorized into at most $2^n(2^n-1)/2$ two-level unitary matrices. An exponential amount of operations, in terms of the number of quantum levels, is needed to simulate U; by definition not an effective simulation.

This shows that quantum computers performing quantum simulations not necessarily fulfill their promise. For each physical system to be simulated one has to find representations of the Hamiltonian that leads to polynomial complexity in the number of operations. After one has found a proper representation of the Hamiltonian, the time evolution operator $\exp(-iH\Delta t)$ is found by using a Trotter approximation, for example

$$U = e^{-iH\Delta t} = e^{-i(\sum_k H_k)\Delta t} = \prod_k e^{-iH_k\Delta t} + \mathcal{O}(\Delta t^2). \tag{6}$$

There are different ways to approximate U by products of exponentials of the different terms of the Hamiltonian, see Ref. [37] and Eq. (41). The essential idea is to find a form of the Hamiltonian where these factors in the approximated time evolution operator can be further factorized into single- and two-qubit operations effectively. In Refs. [24, 38] it was shown how to do this in principle for any many-body fermion system using the Jordan-Wigner transformation. In this article we show how to create a quantum compiler that takes any many-body fermion Hamiltonian and outputs the quantum gates needed to simulate the time evolution operator. We implement it for the case of two-body fermion Hamiltonians and show results from numerical calculations finding the energylevels of the well known pairing and Hubbard models.

C. The Jordan-Wigner transformation

For a spin-1/2 one-dimensional quantum spin-chain a fermionization procedure exists which allows the mapping between spin operators and fermionic creation-annihilation operators. The algebra governing the spin chain is the $SU(2)$ algebra, represented by the σ-matrices. The Jordan-Wigner transformation is a transformation from fermionic annihilation and creation operators to the σ-matrices of a spin-1/2 chain, see for example Ref. [39] for more details on the Jordan-Wigner transformation.

There is an isomorphism between the two systems, meaning that any a or $a^\dagger$ operator can be transformed into a tensor product of σ-matrices operating on a set of qubits. This was explored by Somma *et al.* in Refs. [24, 25]. The authors demonstrated, with an emphasis on single-particle fermionic operators, that the Jordan-Wigner transformation ensures efficient, i.e., not exponential complexity, simulations of a fermionic system on a quantum computer. Similar transformations must be found for other systems, in order to efficiently simulate many-body systems. This was the main point in Ref. [25].

We present here the various ingredients needed in order to transform a given Hamiltonian into a practical form suitable for quantum mechanical simulations.

We begin with the fermionic creation and annihilation operators, which satisfy the following anticommutation relations

$$\{a_k, a_l\} = \{a_k^\dagger, a_l^\dagger\} = 0, \quad \{a_k^\dagger, a_l\} = \delta_{kl}. \tag{7}$$

Thereafter we define the three traceless and Hermitian generators of the $SU(2)$ group, the σ-matrices σ_x, σ_y and σ_z. Together with the identity matrix $\mathbf{1}$ they form

a complete basis for all Hermitian 2×2 matrices. They can be used to write all Hamiltonians on a spin 1/2 chain when taking sums of tensor products of these, in other words they form a product basis for the operators on the qubits. The three σ-matrices are

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{8}$$

We define the raising and lowering matrices as

$$\sigma_+ = \frac{1}{2}(\sigma_x + i\sigma_y) = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix},$$

$$\sigma_- = \frac{1}{2}(\sigma_x - i\sigma_y) = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}. \tag{9}$$

The transformation is based on the fact that for each possible quantum state of the fermion system, there can be either one or zero fermions. Therefore we need n qubits for a system with n possible fermion states. A qubit in state $|0\rangle^i = a_i^\dagger |vacuum\rangle$ represents a state with a fermion, while $|1\rangle^i = |vacuum\rangle$ represents no fermions. Then the raising operator σ_+ changes $|1\rangle$ into $|0\rangle$ when

$$|0\rangle \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{10}$$

This means that σ_+ acts as a creation operator, and σ_- acts as an annihilation operator. In addition, because of the anticommutation of creation(annihilation) operators for different states we have $a_1^\dagger a_2^\dagger |vacuum\rangle = -a_2^\dagger a_1^\dagger |vacuum\rangle$, meaning that for creation and annihilation operators for states higher than the state corresponding to the first qubit, we need to multiply with a σ_z-matrix on all the qubits leading up to the one in question, in order to get the correct sign in the final operation. This leads us to the Jordan-Wigner transformation [24, 25]

$$a_n^\dagger = \left(\prod_{k=1}^{n-1} \sigma_z^k \right) \sigma_+^n, \quad a_n = \left(\prod_{k=1}^{n-1} \sigma_z^k \right) \sigma_-^n. \tag{11}$$

The notation $\sigma_z^i \sigma_+^j$ means a tensor product of the identity matrix on all qubits other than i and j, $\mathbf{1} \otimes \sigma_z \otimes \mathbf{1} \otimes \sigma_+ \otimes \mathbf{1}$, if $i < j$, with $\mathbf{1}$ being the identity matrices of appropriate dimension.

D. Single-particle Hamiltonian

What we must do now is to apply the Jordan-Wigner transformation to a general fermionic Hamiltonian composed of creation and annihilation operators, so we can write it as a sum of products of σ matrices. The matrix σ^k is then an operation on the k^{th} qubit representing the k^{th} quantum level of the fermion system. When we have expressed the Hamiltonian as a sum of products of operations on the qubits representing the system, we must find a representation of the time evolution operator as products of two-qubit operations. These operations can be further decomposed into elementary operations, see subsection II B 1 for further discussion.

1. Jordan-Wigner transformation of the one-body part

We first describe the procedure for the simplest case of a general single-particle Hamiltonian,

$$H_1 = \sum_i E_{ii} a_i^\dagger a_i + \sum_{i<j} E_{ij} (a_i^\dagger a_j + a_j^\dagger a_i). \tag{12}$$

We now use the transformation of Eq. (11) on the terms $a_i^\dagger a_j$.

The diagonal terms of the one-particle Hamiltonian, that is the case where $i = j$, can be rewritten as

$$a_i^\dagger a_i = \left(\prod_{k=1}^{i-1} \sigma_z^k \right) \sigma_+^i \left(\prod_{k=1}^{i-1} \sigma_z^k \right) \sigma_-^i$$

$$= \sigma_+^i \sigma_-^i = \frac{1}{2} \left(\mathbf{1}^i + \sigma_z^i \right), \tag{13}$$

since $(\sigma_z)^2 = \mathbf{1}$ which is the number operator. It counts whether or not a fermion is in state i. In the case of qubits counting whether or not the qubit is in state $|0\rangle$, we have eigenvalue one for $|0\rangle$ and eigenvalue zero for $|1\rangle$. The action of this Hamiltonian on qubit i can be simulated using the single-qubit operation

$$U = e^{-i(\mathbf{1}+\sigma_z)E_{ii}\Delta t} = \begin{pmatrix} e^{-iE_{ii}\Delta t} & 0 \\ 0 & 1 \end{pmatrix}, \tag{14}$$

see subsection II B 1 for other examples of single-qubit gates.

For the non-diagonal elements, $i < j$, not all of the σ_z matrices multiply with each other and end up in the identity operation. As an example we will consider a five level system, $n = 5$, and look at the transformation of the term $a_i^\dagger a_j$ whith $i = 2$ and $j = 4$,

$$\begin{aligned} a_2^\dagger &= \sigma_z \otimes \sigma_+ \otimes \mathbf{1} \otimes \mathbf{1} \otimes \mathbf{1}, \\ a_4 &= \sigma_z \otimes \sigma_z \otimes \sigma_z \otimes \sigma_- \otimes \mathbf{1}, \\ &\Downarrow \\ a_2^\dagger a_4 &= \mathbf{1} \otimes (\sigma_+ \sigma_z) \otimes \sigma_z \otimes \sigma_- \otimes \mathbf{1}. \end{aligned} \tag{15}$$

The operation on all qubits before i and after j is identity, on qubits $i+1$ through $j-1$ it is σ_z. We can then write the non-diagonal one-body operators as

$$
\begin{aligned}
a_i^\dagger a_j + a_j^\dagger a_i &= (\sigma_+^i \sigma_z^i)\left(\prod_{k=i+1}^{j-1}\sigma_z^k\right)\sigma_-^j + (\sigma_z^i\sigma_-^i)\left(\prod_{k=i+1}^{j-1}\sigma_z^k\right)\sigma_+^j \\
&= -\sigma_+^i\left(\prod_{k=i+1}^{j-1}\sigma_z^k\right)\sigma_-^j - \sigma_-^i\left(\prod_{k=i+1}^{j-1}\sigma_z^k\right)\sigma_+^j \\
&= -\frac{1}{2}\left\{\sigma_x^i\left(\prod_{k=i+1}^{j-1}\sigma_z^k\right)\sigma_x^j + \sigma_y^i\left(\prod_{k=i+1}^{j-1}\sigma_z^k\right)\sigma_y^j\right\}. \qquad (16)
\end{aligned}
$$

Using Eqs. (13) and (16) the total single-particle fermionic Hamiltonian of n quantum levels, transformed using the Jordan-Wigner transformation of Eq. (11), is written as

$$
\begin{aligned}
H_1 &= \sum_i E_{ii} a_i^\dagger a_i + \sum_{i<j} E_{ij}(a_i^\dagger a_j + a_j^\dagger a_i) \\
&= \frac{1}{2}\sum_i E_{ii}\left(\mathbf{1}^i + \sigma_z^i\right) \\
&- \frac{1}{2}\sum_{i<j} E_{ij}\left\{\sigma_x^i\left(\prod_{k=i+1}^{j-1}\sigma_z^k\right)\sigma_x^j\right. \\
&+ \left.\sigma_y^i\left(\prod_{k=i+1}^{j-1}\sigma_z^k\right)\sigma_y^j\right\}. \qquad (17)
\end{aligned}
$$

2. *Transformation into two-qubit operations*

The Hamiltonian is now transformed into a sum of many-qubit operations, $H = \sum_l H_l$. The $a_2^\dagger a_4$ term in Eq. (15) for example, is transformed into a three-qubit operation. The next step is to factorize these many-qubit operations H_l into products of two-qubit operations, so that we in the end can get a product of two-qubit operations U_{kl}, that when performed in order give us the time evolution operator corresponding to each term in the Hamiltonian, $\exp(-iH_l\Delta t) = \prod_k U_{kl}$.

The first thing we do is to find a set of two-qubit operations that together give us the Hamiltonian, and later we will see that to find the time evolution from there is straightforward. The many-qubit terms in Eq. (17) are products of the type $\sigma_x\sigma_z\cdots\sigma_z\sigma_x$ with σ_x or σ_y at either end. These products have to be factorized into a series of two-qubit operations. What we wish to do is successively build up the operator using different unitary transformations. This can be achieved with successive operations with the σ-matrices, starting with for example σ_z^i, which can be transformed into σ_x^i, then transformed into $\sigma_y^i\sigma_z^{i+1}$ and so forth. Our goal now is to express each term in the Hamiltonian Eq. (17) as a product of the type $\sigma_x^i\sigma_z\cdots\sigma_z\sigma_x^j = (\prod_k U_k^\dagger)\sigma_z^i(\prod_{k'} U_{k'})$, with a different form in the case where the Hamiltonian term starts and ends with a σ_y matrix. To achieve this we need the transformations in Eqs. (A.8)-(A.11). We will use this to find the time-evolution operator for each Hamiltonian, see Eq. (21) below.

To understand how we factorize the Hamiltonian terms into single- and two-qubit operations we follow a bottom up procedure. First, if we have a two qubit system, with the operator $\sigma_z \otimes \mathbf{1}$, we see that the unitary operation $\exp(i\pi/4\sigma_z\otimes\sigma_z)$ transforms it into

$$
e^{-i\pi/4\sigma_z\otimes\sigma_z}\,(\sigma_z\otimes\mathbf{1})\,e^{i\pi/4\sigma_z\otimes\sigma_z} = \sigma_z\otimes\sigma_z. \qquad (18)
$$

In addition, if we start out with the operator σ_z^i we can transform it into σ_x^i or σ_y^i using the operators $\exp(i\pi/4\sigma_y)$ or $\exp(-i\pi/4\sigma_x)$ accordingly.

We can then generate the $\prod_k \sigma_z^k$ part of the terms in Eq. (17) by succesively applying the operator $\exp(i\pi/4\sigma_z^i\sigma_z^l)$ for $l = 2$ through $l = j$. Yielding the operator $\sigma_a^i\prod_{k=i+1}^j\sigma_z^k$ with a phase of ±1, because of the sign change in Eqs. (A.10) and (A.11). We write σ_a to show that we can start with both a σ_x and a σ_y matrix. To avoid the sign change we can simply use the operator $\exp(-i\pi/4\sigma_z^i\sigma_z^l)$ instead for those cases where we have σ_y^i on site i instead of σ_x^i. This way we always keep the same phase.

Finally, we use the operator $\exp(i\pi/4\sigma_y)$ if we want the string of operators to end with σ_x, or $\exp(-i\pi/4\sigma_x)$ if we want it to end with σ_y. The string of operators starts with either σ_x or σ_y. For an odd number of $\exp(\pm i\pi/4\sigma_z^i\sigma_z^l)$ operations, the operations that add a σ_z to the string, the first operator has changed from what we started with. In other words we have σ_x instead of σ_y at the start of the string or vice versa, see Eqs. (A.10) and (A.11). By counting, we see that we do $j-i$ of the $\exp(\pm i\pi/4\sigma_z^i\sigma_z^l)$ operations to get the string $\sigma_a^i\sigma_z^{i+1}\cdots\sigma_z^j$. and therefore if $j-i$ is odd, the first matrix is the opposite of what we want in the final string. The following simple example can serve to clarify. We want the Hamiltonian $\sigma_x^1\sigma_z^2\sigma_x^3 = \sigma_x\otimes\sigma_z\otimes\sigma_x$, and by using the transformations in Eqs. (A.8)-(A.11) we can construct it as a product of single- and two-qubit operations in the

following way,

$$\begin{aligned}(e^{-\pi/4\sigma_y^1})\sigma_z^1(e^{\pi/4\sigma_y^1}) &= \sigma_x^1\\ (e^{-i\pi/4\sigma_z^1\sigma_z^2})\sigma_x^1(e^{i\pi/4\sigma_z^1\sigma_z^2}) &= \sigma_y^1\sigma_z^2\\ (e^{i\pi/4\sigma_z^1\sigma_z^3})\sigma_y^1\sigma_z^2(e^{-i\pi/4\sigma_z^1\sigma_z^3}) &= \sigma_x^1\sigma_z^2\sigma_z^3\\ (e^{-i\pi/4\sigma_y^3})\sigma_x^1\sigma_z^2\sigma_z^3(e^{i\pi/4\sigma_y^3}) &= \sigma_x^1\sigma_z^2\sigma_x^3. \end{aligned} \quad (19)$$

We see that we have factorized $\sigma_x^1\sigma_z^2\sigma_x^3$ into $U_4^\dagger U_3^\dagger U_2^\dagger U_1^\dagger \sigma_z^1 U_1 U_2 U_3 U_4$.

Now we can find the time-evolution operator $\exp(-iH\Delta t)$ corresponding to each term of the Hamiltonian, which is the quantity of interest. Instead of starting with the operator σ_z^i we start with the corresponding evolution operator and observe that

$$\begin{aligned}U^\dagger e^{-i\sigma_z a}U &= U^\dagger\left(\cos(a)\mathbf{1} - i\sin(a)\sigma_z\right)U\\ &= \cos(a)\mathbf{1} - i\sin(a)U^\dagger\sigma_z U\\ &= e^{-iU^\dagger\sigma_z U a},\end{aligned} \quad (20)$$

where a is a scalar. This means that we have a series of unitary transformations on this operator yielding the final evolution, namely

$$e^{-i\sigma_x^i\sigma_z\cdots\sigma_z\sigma_x^j a} = \left(\prod_k U_k^\dagger\right)e^{-i\sigma_z^i a}\left(\prod_{k'} U_{k'}\right), \quad (21)$$

with the exact same unitary operations U_k as we find when we factorize the Hamiltonian. These are now the single- and two-qubit operations we were looking for, first we apply the operations U_k to the appropriate qubits, then $\exp(-i\sigma_z^i a)$ to qubit i, and then the $U_k^\dagger$ operations, all in usual matrix multiplication order.

E. Two-body Hamiltonian

In this section we will do the same for the general two-body fermionic Hamiltonian. The two-body part of the Hamiltonian can be classified into diagonal elements and non-diagonal elements. Because of the Pauli principle and the anti-commutation relations for the creation and annihilation operators, some combinations of indices are not allowed. The two-body part of our Hamiltonian is

$$H_2 = \sum_{ijkl} V_{ijkl}a_i^\dagger a_j^\dagger a_l a_k, \quad (22)$$

where the indices run over all possible states and n is the total number of available quantum states. The single-particle labels $ijkl$ refer to their corresponding sets of quantum numbers, such as projection of total spin, number of nodes in the single-particle wave function etc. Since every state $ijkl$ is uniquely defined, we cannot have two equal creation or annihilation operators and therefore $i \neq j$ and $k \neq l$.

When $i = l$ and $j = k$, or $i = k$ and $j = l$, we have a diagonal element in the Hamiltonian matrix, and the output state is the same as the input state. The operator term corresponding to V_{ijji} has these equalities due to the anti-commutation relations

$$\begin{aligned}a_i^\dagger a_j^\dagger a_i a_j &= a_j^\dagger a_i^\dagger a_j a_i\\ &= -a_i^\dagger a_j^\dagger a_j a_i\\ &= -a_j^\dagger a_i^\dagger a_i a_j,\end{aligned} \quad (23)$$

which means that

$$V_{ijji} = V_{jiij} = -V_{ijij} = -V_{jiji}. \quad (24)$$

The term $a_i^\dagger a_j^\dagger a_i a_j$ with $i < j$ is described using the Pauli matrices

$$a_i^\dagger a_j^\dagger a_i a_j \quad (25)$$

$$\begin{aligned}&= \left(\prod_{s=1}^{i-1}\sigma_z\right)\sigma_+^i\left(\prod_{t=1}^{j-i}\sigma_z\right)\sigma_+^j\\ &\times\left(\prod_{t=1}^{j-i}\sigma_z\right)\sigma_-^j\left(\prod_{s=1}^{i-1}\sigma_z\right)\sigma_-^i\\ &= \left(\prod_{s=1}^{i-1}(\sigma_z)^4\right)\left(\sigma_+^i\sigma_z^i\sigma_z^i\sigma_-^i\right)\left(\prod_{t=i+1}^{j-1}(\sigma_z)^2\right)\left(\sigma_+^j\sigma_-^j\right)\\ &= \sigma_+^i\sigma_-^i\sigma_+^j\sigma_-^j\\ &= \frac{1}{16}\left(\mathbf{1}+\sigma_z^i\right)\left(\mathbf{1}+\sigma_z^j\right).\end{aligned} \quad (26)$$

When we add all four different permutations of i and j this is the number operator on qubit i multiplied with the number operator on qubit j. The eigenvalue is one if both qubits are in the state $|0\rangle$, that is the corresponding quantum states are both populated, and zero otherwise. We can in turn rewrite the sets of creation and annihilations in terms of the σ-matrices as

$$\begin{aligned}a_i^\dagger a_j^\dagger a_i a_j + a_j^\dagger a_i^\dagger a_j a_i - a_i^\dagger a_j^\dagger a_j a_i - a_j^\dagger a_i^\dagger a_i a_j\\ = \frac{1}{4}\left(\mathbf{1}+\sigma_z^i+\sigma_z^j+\sigma_z^i\sigma_z^j\right).\end{aligned} \quad (27)$$

In the general case we can have three different sets of non-equal indices. Firstly, we see that $a_i^\dagger a_j^\dagger a_l a_k = a_k^\dagger a_l^\dagger a_j a_i$, meaning that the exchange of i with k and j with l gives the same operator $\rightarrow V_{ijkl} = V_{klij}$. This results in a two-body Hamiltonian with no equal indices

$$H_{ijkl} = \sum_{i<k}\sum_{j<l} V_{ijkl}(a_i^\dagger a_j^\dagger a_l a_k + a_k^\dagger a_l^\dagger a_j a_i). \quad (28)$$

Choosing to order the indices from lowest to highest gives us the position where there will be σ_z-matrices to multiply with the different raising and lowering operators, when we perform the Jordan-Wigner transformation Eq. (11). The order of matrix multiplications is fixed once and for all, resulting in three different groups into which these terms fall, namely

$$\begin{array}{lll} I & i<j<l<k, & i\leftrightarrow j,\ k\leftrightarrow l,\\ II & i<l<j<k, & i\leftrightarrow j,\ k\leftrightarrow l,\\ III & i<l<k<j, & i\leftrightarrow j,\ k\leftrightarrow l. \end{array} \quad (29)$$

These 12 possibilities for $a_i^\dagger a_j^\dagger a_l a_k$ are mirrored in the symmetric term in Eq. (28) giving us the 24 different possibilities when permuting four indices.

The $ijkl$ term of Eq. (28) is

$$\begin{aligned} a_i^\dagger a_j^\dagger a_l a_k + a_k^\dagger a_l^\dagger a_j a_i = \\ \left(\prod \sigma_z\right) \sigma_+^i \left(\prod \sigma_z\right) \sigma_+^j \\ \times \left(\prod \sigma_z\right) \sigma_-^l \left(\prod \sigma_z\right) \sigma_-^k \\ + \left(\prod \sigma_z\right) \sigma_+^k \left(\prod \sigma_z\right) \sigma_+^l \\ \times \left(\prod \sigma_z\right) \sigma_-^j \left(\prod \sigma_z\right) \sigma_-^i. \end{aligned} \tag{30}$$

In the case of $i < j < l < k$ we have

$$\begin{aligned} a_i^\dagger a_j^\dagger a_l a_k + a_k^\dagger a_l^\dagger a_j a_i = \\ \left(\prod (\sigma_z)^4\right) \left(\sigma_+^i \sigma_z^i\right) \left(\prod (\sigma_z)^3\right) \sigma_+^j \\ \times \left(\prod (\sigma_z)^2\right) \left(\sigma_-^l \sigma_z^l\right) \left(\prod \sigma_z\right) \sigma_-^k \\ + \left(\prod (\sigma_z)^4\right) \left(\sigma_z^i \sigma_-^i\right) \left(\prod (\sigma_z)^3\right) \sigma_-^j \\ \times \left(\prod (\sigma_z)^2\right) \left(\sigma_z^l \sigma_+^l\right) \left(\prod \sigma_z\right) \sigma_+^k. \end{aligned} \tag{31}$$

Using Eq. (A.12), where we have the rules for sign changes when multiplying the raising and lowering operators with the σ_z matrices, gives us

$$\begin{aligned} -\Big(\sigma_+^i \sigma_z^{i+1} \cdots \sigma_z^{j-1} \sigma_+^j \sigma_-^l \sigma_z^{l+1} \cdots \sigma_z^{k-1} \sigma_-^k \\ + \sigma_-^i \sigma_z^{i+1} \cdots \sigma_z^{j-1} \sigma_-^j \sigma_+^l \sigma_z^{l+1} \cdots \sigma_z^{k-1} \sigma_+^k \Big). \end{aligned} \tag{32}$$

If we switch the order of i and j so that $j < i < l < k$, we change the order in which the σ_z-matrix is multiplied with the first raising and lowering matrices, resulting in a sign change.

$$\begin{aligned} a_i^\dagger a_j^\dagger a_l a_k + a_k^\dagger a_l^\dagger a_j a_i = \\ \left(\prod (\sigma_z)^4\right) \left(\sigma_z^j \sigma_+^j\right) \left(\prod (\sigma_z)^3\right) \sigma_+^i \\ \times \left(\prod (\sigma_z)^2\right) \left(\sigma_-^l \sigma_z^l\right) \left(\prod \sigma_z\right) \sigma_-^k \\ + \left(\prod (\sigma_z)^4\right) \left(\sigma_-^j \sigma_z^j\right) \left(\prod (\sigma_z)^3\right) \sigma_-^i \\ \times \left(\prod (\sigma_z)^2\right) \left(\sigma_z^l \sigma_+^l\right) \left(\prod \sigma_z\right) \sigma_+^k \\ = + \Big(\sigma_+^j \sigma_z^{j+1} \cdots \sigma_z^{i-1} \sigma_+^i \sigma_-^l \sigma_z^{l+1} \cdots \sigma_z^{k-1} \sigma_-^k \\ + \sigma_-^j \sigma_z^{j+1} \cdots \sigma_z^{i-1} \sigma_-^i \sigma_+^l \sigma_z^{l+1} \cdots \sigma_z^{k-1} \sigma_+^k \Big). \end{aligned} \tag{33}$$

We get a change in sign for every permutation of the ordering of the indices from lowest to highest because of the matrix multiplication ordering. The ordering is described by another set of indices $\{s_\alpha, s_\beta, s_\gamma, s_\delta\} \in \{i, j, k, l\}$ where $s_\alpha < s_\beta < s_\gamma < s_\delta$. We assign a number to each of the four indices, $i \leftrightarrow 1$, $j \leftrightarrow 2$, $l \leftrightarrow 3$ and $k \leftrightarrow 4$. If $i < j < l < k$ we say the ordering is $\alpha = 1$, $\beta = 2$, $\gamma = 3$ and $\delta = 4$, where α is a number from one to four indicating which of the indices i, j, l and k is the smallest. If i is the smallest, $\alpha = 1$ and $s_\alpha = i$. This allows us to give the sign of a given $(a_i^\dagger a_j^\dagger a_l a_k + a_k^\dagger a_l^\dagger a_j a_i)$ term using the totally anti-symmetric tensor with four indices, which is $+1$ for even permutations, and -1 for odd permutations. For each of the three groups in Eq. (29) we get a different set of raising and lowering operators on the lowest, next lowest and so on, indices, while the sign for the whole set is given by $-\varepsilon^{\alpha\beta\gamma\delta}$.

We are in the position where we can use the relation in Eq. (9) to express the Hamiltonian in terms of the σ-matrices. We get 16 terms with products of four σ_x and or σ_y matrices in the first part of Eq. (31), then when we add the Hermitian conjugate we get another 16 terms. The terms with an odd number of σ_y matrices have an imaginary phase and are therefore cancelled out when adding the conjugates in Eq. (28). This leaves us with just the terms with four σ_x matrices, four σ_y matrices and two of each in different orderings. The final result is given as an array with a global sign and factor given by the permutation of the ordering, and eight terms with different signs depending on which of the three groups, Eq. (29), the set of indices belong to. These differing rules are due to the rules for σ_z multiplication with the raising and lowering operators, resulting in

$$\begin{aligned} a_i^\dagger a_j^\dagger a_l a_k + a_k^\dagger a_l^\dagger a_j a_i = \\ -\frac{\varepsilon^{\alpha\beta\gamma\delta}}{8} \begin{cases} I \;\; II \;\; III & \\ + \;\; + \;\; + & \sigma_x^{s_\alpha} \sigma_z \cdots \sigma_z \sigma_x^{s_\beta} \sigma_x^{s_\gamma} \sigma_z \cdots \sigma_z \sigma_x^{s_\delta} \\ - \;\; + \;\; + & \sigma_x \cdots \sigma_x \quad \sigma_y \cdots \sigma_y \\ + \;\; - \;\; + & \sigma_x \cdots \sigma_y \quad \sigma_x \cdots \sigma_y \\ + \;\; + \;\; - & \sigma_x \cdots \sigma_y \quad \sigma_y \cdots \sigma_x \\ + \;\; + \;\; - & \sigma_y \cdots \sigma_x \quad \sigma_x \cdots \sigma_y \\ + \;\; - \;\; + & \sigma_y \cdots \sigma_x \quad \sigma_y \cdots \sigma_x \\ - \;\; + \;\; + & \sigma_y \cdots \sigma_y \quad \sigma_x \cdots \sigma_x \\ + \;\; + \;\; + & \sigma_y \cdots \sigma_y \quad \sigma_y \cdots \sigma_y \end{cases} \end{aligned} \tag{34}$$

where the letters I, II and III refer to the subgroups defined in Eq. (29).

As for the single-particle operators in subsection II D we now need to factorize these multi-qubit terms in the Hamiltonian to products of two-qubit and single-qubit operators. Instead of transforming a product of the form $az \cdots zb$, we now need to transform a product of the form $az \cdots zbcz \cdots zd$, where a, b, c and d are short for either σ_x or σ_y while z is short for σ_z. The generalization is quite straightforward, as we see that if the initial operator is $\sigma_z^{s_\alpha} \sigma_z^{s_\gamma}$ instead of just $\sigma_z^{s_\alpha}$, we can use the same set of transformations as for the single-particle case,

$$\begin{aligned} & U_k^\dagger \cdots U_1^\dagger \sigma_z^{s_\alpha} U_1 \cdots U_k \\ & = \sigma_a^{s_\alpha} \sigma_z \cdots \sigma_z \sigma_b^\beta \\ \Rightarrow \quad & U_k^\dagger \cdots U_1^\dagger \sigma_z^{s_\alpha} \sigma_z^{s_\gamma} U_1 \cdots U_k \\ & = \sigma_a^{s_\alpha} \sigma_z \cdots \sigma_z \sigma_b^{s_\beta} \sigma_z^{s_\gamma}. \end{aligned} \tag{35}$$

Using the same unitary two-qubit transformations, which we now call V, that take $\sigma_z^{s_\gamma}$ to $\sigma_c^{s_\gamma}\sigma_z\cdots\sigma_z\sigma_d^{s_\delta}$, we find

$$V_s^\dagger\cdots V_1^\dagger U_k^\dagger\cdots U_1^\dagger\sigma_z^{s_\alpha}\sigma_z^{s_\gamma}U_1\cdots U_kV_1\cdots V_s = \sigma_a^{s_\alpha}\sigma_z\cdots\sigma_z\sigma_b^\beta\sigma_c^{s_\gamma}\sigma_z\cdots\sigma_z\sigma_d^\delta. \quad (36)$$

This straightforward generalization of the procedure from the single-particle Hamiltonian case is possible because the operations commute when performed on different qubits.

With the above expressions, we can start with the unitary operator $\exp(-ia\sigma_z^{s_\alpha}\sigma_z^{s_\gamma})$ and have two different series of unitary operators that give us the evolution operator of the desired Hamiltonian. The U operators are defined as in Eq. (21),

$$e^{-i\sigma^{s_\alpha}\sigma_z\cdots\sigma_z\sigma^{s_\beta}a} = \left(\prod_k U_k^\dagger\right)e^{-i\sigma_z^{s_\alpha}a}\left(\prod_{k'}U_{k'}\right), \quad (37)$$

while the V operators are defined in a similar way

$$e^{-i\sigma^{s_\gamma}\sigma_z\cdots\sigma_z\sigma^{s_\delta}a} = \left(\prod_s V_s^\dagger\right)e^{-i\sigma_z^{s_\gamma}a}\left(\prod_{s'}V_{s'}\right), \quad (38)$$

where the σ-matrices without subscripts represent that we can have σ_x or σ_y in each position.

This gives us the total evolution operator for each term in Eq. (34)

$$e^{-i\sigma^{s_\alpha}\sigma_z\cdots\sigma_z\sigma^{s_\beta}\sigma^{s_\gamma}\sigma_z\cdots\sigma_z\sigma^{s_\delta}a} = \left(\prod_s V_s^\dagger\right)\left(\prod_k U_k^\dagger\right)e^{-i\sigma_z^{s_\alpha}\sigma_z^{s_\gamma}a} \times\left(\prod_{k'}U_{k'}\right)\left(\prod_{s'}V_{s'}\right). \quad (39)$$

Here we have all the single- and two-qubit operations we need to perform on our set of qubits, that were initially in the state $|\psi\rangle$, to simulate the time evolution $\exp(-iH_k\Delta t)|\psi\rangle$ of the Hamiltonian term $H_k = \sigma^{s_\alpha}\sigma_z\cdots\sigma_z\sigma^{s_\beta}\sigma^{s_\gamma}\sigma_z\cdots\sigma_z\sigma^{s_\delta}$. Every factor in the above equation is a single- or two-qubit operation that must be performed on the qubits in proper matrix multiplication order.

When using the Jordan-Wigner transformation of Eq. (11) applied to our two model Hamiltonians of Eqs. (2) and (3), we choose a representation with two qubits at each site. These correspond to fermions with spin up and down, respectively. The number of qubits, n, is always the total number of available quantum states and therefore it is straightforward to use this model on systems with higher degeneracy, such as those encountered in quantum chemistry [3] or nuclear physics [16]. Site one spin up is qubit one, site one spin down is qubit two and site two spin up is qubit three and so on. To get all the quantum gates one needs to simulate a given Hamiltonian one needs to input the correct E_{ij} and V_{ijkl} values.

F. Complexity of the quantum computing algorithm

In order to test the efficiency of a quantum algorithm, one needs to know how many qubits, and how many operations on these, are needed to implement the algorithm. Usually this is a function of the dimension of the Hilbert space on which the Hamiltonian acts. The natural input scale in the fermionic simulator is the number of quantum states, n, that are available to the fermions. In our simulations of the Hubbard and the pairing models of Eqs. (2) and (3), respectively, the number of qubits is $n = 2N$ since we have chosen systems with double-degeneracy for every single-particle state, where N is the number of energy-levels in the model. We use one qubit to represent each possible fermion state, on a real quantum computer, however, one should implement some error-correction procedure using several qubits for each state, see Ref. [37]. The complexity in number of qubits remains linear, however, since $\mathcal{O}(n)$ qubits are needed for error correction.

The single-particle Hamiltonian has potentially $\mathcal{O}(n^2)$ different E_{ij} terms. The two-particle Hamiltonian has up to $\mathcal{O}(n^4)$ V_{ijkl} terms. A general m-body interaction has in the worst case $\mathcal{O}(n^{2m})$ terms. It is straightforward to convince oneself that the pairing model has $\mathcal{O}(n^2)$ terms while in the Hubbard model we end up with $\mathcal{O}(n)$ terms. Not all models have maximum complexity in the different m-body interactions.

How many two-qubit operations do each of these terms need to be simulated? First of all a two-qubit operation will in general have to be decomposed into a series of universal single- and two-qubit operations, depending entirely on the given quantum simulator. A particular physical realization might have a natural implementation of the $\sigma_z^i\otimes\sigma_z^j$ gate and save a lot of intermediary operations. Others will have to use a fixed number of operations in order to apply the operation on any two qubits. A system with only nearest neighbor interactions would have to use $\mathcal{O}(n)$ operations for each $\sigma_z^i\otimes\sigma_z^j$ gate, and thereby increase the polynomial complexity by one degree.

In our discussion on the one-body part of the Hamiltonian, we saw that for each E_{ij} we obtained the $a_i^\dagger a_j + a_j^\dagger a_i$ operator which is transformed into the two terms in Eq. (16), $\sigma_x\sigma_z\cdots\sigma_z\sigma_x$ and $\sigma_y\sigma_z\cdots\sigma_z\sigma_y$. We showed how these terms are decomposed into $j-i+2$ operations, leading to twice as many unitary transformations on an operator, $VAV^\dagger$ for the time evolution. The average of $j-i$ is $n/2$ in this case and in total we need to perform $2\times2\times n/2 = 2n$ two-qubit operations per single-particle term in the Hamiltonian, a linear complexity.

In the two-particle case each term $V_{ijkl}(a_i^\dagger a_j^\dagger a_l a_k + a_k^\dagger a_l^\dagger a_j a_i)$ is transformed into a sum of eight operators of the form $\sigma^{s_\alpha}\sigma_z\cdots\sigma_z\sigma^{s_\beta}\sigma^{s_\gamma}\sigma_z\cdots\sigma_z\sigma^{s_\delta}$, Eq. (34). The two parts of these operators are implemented in the same way as the $\sigma^i\sigma_z\cdots\sigma_z\sigma^j$ term of the single-particle Hamil-

tonian, which means they require $s_\beta - s_\alpha$ and $s_\delta - s_\gamma$ operations, since $s_\alpha < s_\beta < s_\gamma < s_\delta$ the average is $n/4$. For both of these parts we need to perform both the unitary operation V and it's Hermitian conjugate $V^\dagger$. In the end we need $2 \times 2 \times 8 \times n/4 = 8n$ two-qubit operations per two-particle term in the Hamiltonian, the complexity is linear.

A term of an m-body Hamiltonian will be transformed into 2^{2m} operators since each annihilation and creation operator is transformed into a sum of σ_x and σ_y matrices. All the imaginary terms cancel out and we are left with 2^{2m-1} terms. Each of these terms will include $2m$ σ matrices, in products of the form $\prod_{k=1}^{m} \sigma^i \sigma_z \cdots \sigma_z \sigma^j$, and we use the same procedure as discussed above to decompose these m factors into unitary transformations. In this case each factor will require an average of $n/2m$ operations for the same reasons as in the two-body case. All in all, each m-body term in the Hamiltonian requires $2^{2m-1} \times 2 \times m \times n/2m = 2^{2m-1} n$ operations.

Thus, the complexity for simulating one m-body term of a fermionic many-body Hamiltonian is linear in the number of two-qubit operations, but the number of terms is not. For a full-fledged simulation of general three-body forces, in common use in nuclear physics [40, 41, 42], the total complexity of the simulation is $\mathcal{O}(n^7)$. A complete two-particle Hamiltonian would be $\mathcal{O}(n^5)$.The bottleneck in these simulations is the number of terms in the Hamiltonian, and for systems with less than the full number of terms the simulation will be faster. This is much better than the exponential complexity of most simulations on classical computers.

III. ALGORITHMIC DETAILS

Having detailed how a general Hamiltonian, of two-body nature in our case, can be decomposed in terms of various quantum gates,we present here details of the implementation of our algorithm for finding eigenvalues and eigenvectors of a many-fermion system. For our tests of the fermionic simulation algorithm we have implemented the phase-estimation algorithm from [37] which finds the eigenvalues of an Hamiltonian operating on a set of simulation qubits. There are also other quantum computer algorithms for finding expectation values and correlation functions, as discussed by Somma *et al.* in Refs. [25, 26]. In the following we first describe the phase-estimation algorithm, and then describe its implementation and methods we have developed in using this algorithm. A much more thorough description of quantum computers and the phase-estimation algorithm can be found in [43].

A. Phase-estimation algorithm

To find the eigenvalues of the Hamiltonian we use the unitary time evolution operator we get from the Hamiltonian. We have a set of simulation qubits representing the system governed by the Hamiltonian, and a set of auxiliary qubits, called work qubits [20, 21], in which we will store the eigenvalues of the time evolution operator. The procedure is to perform several controlled time evolutions with work qubits as control qubits and the simulation qubits as targets, see for example Ref. [37] for information on controlled qubit operations. For each work qubit we perform the controlled operation on the simulation qubits with a different time parameter, giving all the work qubits different phases. The information stored in their phases is extracted using first an inverse Fourier transform on the work qubits alone, and then performing a measurement on them. The values of the measurements give us directly the eigenvalues of the Hamiltonian after the algorithm has been performed a number of times.

The input state of the simulation qubits is a random state in our implementation, which is also a random superposition of the eigenvectors of the Hamiltonian $|\psi\rangle = \sum_k c_k |k\rangle$. It does not have to be a random state, and in [44] the authors describe a quasi-adiabatic approach, where the initial state is created by starting in the ground state for the non-interacting Hamiltonian, a qubit basis state, e.g. $|0101\cdots101\rangle$, and then slowly the interacting part of the Hamiltonian is turned on. This gives us an initial state mostly comprising the true ground state, but it can also have parts of the lower excited states if the interacting Hamiltonian is turned on a bit faster. In for example nuclear physics it is common to use a starting state for large-scale diagonalizations that reflects some of the features of the states one wishes to study. A typical example is to include pairing correlations in the trial wave function, see for example Refs. [16, 34]. Iterative methods such as the Lanczo's diagonalization technique [19, 45] converge much faster if such starting vectors are used. However, although more iterations are needed, even a random starting vector converges to the wanted states.

The final state of all the qubits after an inverse Fourier transform on the work qubits is

$$\sum_k c_k |\phi^{[k]} 2^t\rangle \otimes |k\rangle. \tag{40}$$

If the algorithm works perfectly, $|k\rangle$ should be an exact eigenstate of U, with an exact eigenvalue $\phi^{[k]}$. When we have the eigenvalues of the time evolution operator we easily find the eigenvalues of the Hamiltonian. We can summarize schematically the phase-estimation algorithm as follows:

1. Intialize each of the work qubits to $1/\sqrt{2}(|0\rangle + |1\rangle)$ by initializing to $|0\rangle$ and applying the Hadamard gate, H, see Fig. 2.

2. Initialize the simulation qubits to a random or specified state, depending on the whether one wants the whole eigenvalue spectrum.

3. Perform conditional time evolutions on the simulation qubits, with different timesteps Δt and different work qubits as the control qubits.

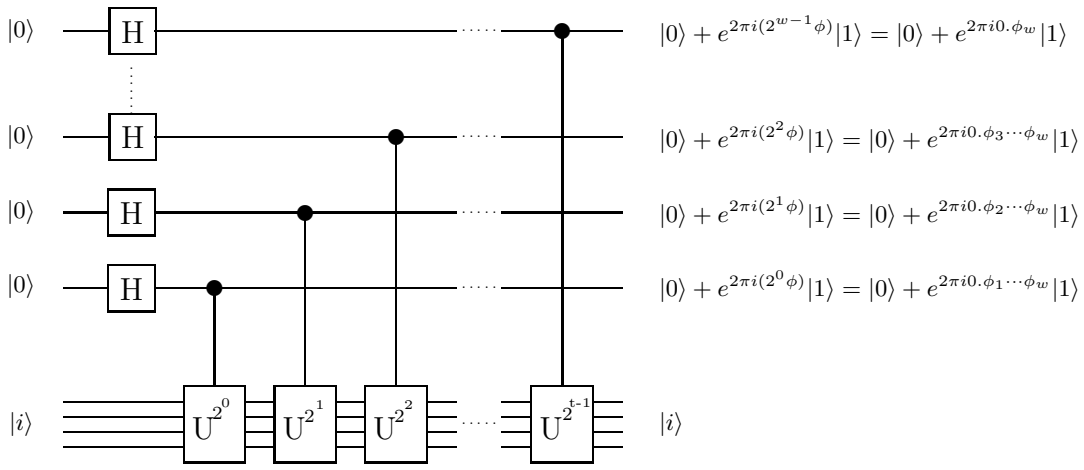


FIG. 3: Phase estimation circuit showing all the different qubit lines schematically with operations represented by boxes. The boxes connected by vertical lines to other qubit lines are controlled operations, with the qubit with the black dot as the control qubit.

4. Perform an inverse Fourier transform on the work qubits.

5. Measure the work qubits to extract the phase.

6. Repeat steps 1-6 until the probability distribution gathered from the measurement results is good enough to read out the wanted eigenvalues.

As discussed above a set of two-qubit operations can be simulated by the CNOT two-qubit operation and a universal set of single-qubit operations. We will not use or discuss any such implementation in this article, as one will have to use a different set for each physical realization of a quantum computer. When simulating a fermion system with a given quantum computer, our algorithm will first take the fermionic many-body evolution operator to a series of two-qubit and single-qubit operations, and then one will have to have a system dependent setup that takes these operations to the basic building blocks that form the appropriate universal set.

In subsection II E we showed how to take any two-particle fermionic Hamiltonian to a set of two-qubit operations that approximate the evolution operator. In addition we must use one of the Trotter approximations [46, 47, 48] Eqs. (6) and (41) that take the evolution operator of a sum of terms to the product of the evolution operator of the individual terms, see for example Ref. [37] for details. To order $\mathcal{O}(\Delta t^2)$ in the error we use Eq. (6) while to order $\mathcal{O}(\Delta t^3)$ we have

$$e^{-i(A+B)\Delta t} = e^{-iA\Delta t/2}e^{-iB\Delta t}e^{-iA\Delta t/2} + \mathcal{O}(\Delta t^3). \quad (41)$$

B. Output of the phase-estimation algorithm

The output of the phase-estimation algorithm is a series of measurements of the w number of work qubits. Putting them all together we get a probability distribution that estimates the amplitudes $|c_k|^2$ for each eigenvalue $\phi^{[k]}$. The $\phi^{[k]}2^w$ values we measure from the work qubits, see Eq. (40), are binary numbers from zero to $2^w - 1$, where each one translates to a given eigenvalue of the Hamiltonian depending on the parameters we have used in our simulation. When accurate, a set of simulated measurements will give a distribution with peaks around the true eigenvalues. The probability distribution is calculated by applying non-normalized projection operators to the qubit state,

$$\left(|\phi^{[k]}2^t\rangle\langle\phi^{[k]}2^t| \otimes \mathbf{1}\right)\left(\sum_i c_i|\phi_i 2^t\rangle \otimes |i\rangle\right) = c_k|\phi^{[k]}2^t\rangle\otimes|k\rangle.$$

The length of this vector squared gives us the probability,

$$\left|c_k|\phi^{[k]}\rangle 2^t \otimes |k\rangle\right|^2 = |c_k|^2\langle\phi^{[k]}2^t|\phi^{[k]}2^t\rangle\langle k|k\rangle = |c_k|^2. \quad (42)$$

Since we do not employ the exact evolution due to different approximations, we can have non-zero probabilities

for all values of ϕ, yielding a distribution without sharp peaks for the correct eigenvalues and possibly peaks in the wrong places. If we use different random input states for every run through the quantum computer and gather all the measurements in one probability distribution, all the eigenvectors in the input state $|\psi\rangle = \sum_k c_k|k\rangle$ should average out to the same amplitude. This means that eigenvalues with higher multiplicity, i.e., higher degeneracy, will show up as taller peaks in the probability distribution, while non-degenerate eigenvalues might be difficult to find.

To properly estimate the eigenvalues E_k of the Hamiltonian from this distribution, one must take into account the periodicity of $e^{2\pi i\phi}$. If $0 < \phi' < 1$ and $\phi = \phi' + s$, where s is an integer, then $e^{2\pi i\phi} = e^{2\pi i\phi'}$. This means that to get all the eigenvalues correctly ϕ must be positive and less than one. Since $\phi = -E_k\Delta t/2\pi$ this means all the eigenvalues E_k must be negative, this merely means subtracting a constant we denote E_{max} from the Hamiltonian, $H' = H - E_{max}$, where E_{max} is greater than the largest eigenvalue of H. The values we read out from the work qubits are integers from zero to $2^w - 1$. In other words, we have $\phi^{[k]}2^w \in [0, 2^w - 1]$, with $\phi = 0$ for $\Delta t = 0$.

The value $\phi = 0$ corresponds to the lowest eigenvalue possible to measure, E_{min}, while $\phi = 1$ corresponds to E_{max}. The interval of possible values is then $E_{max} - E_{min} = 2\pi/\Delta t$. If we want to have all possible eigenvalues in the interval the largest value Δt can have is

$$\max(\Delta t) = \frac{2\pi}{E_{max} - E_{min}} \tag{43}$$

1. Spectrum analysis

In the general case one does not know the upper and lower bounds on the eigenvalues beforehand, and therefore for a given E_{max} and Δt one does not know if the $\phi^{[k]}$ values are the correct ones, or if an integer has been lost in the exponential function.

When $\phi = \phi' + s$ for one Δt, and we slightly change Δt, ϕ' will change if $s \neq 0$ as the period of the exponential function is a function of Δt. To find out which of $\phi^{[k]}$s are greater than one, we perform the phase-estimation algorithm with different values for Δt and see which eigenvalues shift. If we measure the same ϕ after adding δt to the time step, and $(\Delta t + \delta t)/\Delta t$ is not a rational number, we know that $\phi < 1$. In practice it does not have to be an irrational number, but only some unlikely fraction.

There are at least two methods for finding the eigenvalues. One can start with a large positive E_{max} and a small Δt, hoping to find that the whole spectrum falls within the range $[E_{min}, E_{max}]$, and from there zoom in until the maximal eigenvalue is slightly less than E_{max} and the groundstate energy is slightly larger than E_{min}. This way the whole spectrum is covered at once. From there we can also zoom in on specific areas of the spectrum, searching the location of the true eigenvalues by shifting Δt.

The number of measurements needed will depend entirely on the statistics of the probability distribution. The number of eigenvalues within the given energy range determines the resolution needed. That said, the number of measurements is not a bottleneck in quantum computer calculations. The quantum computer will prepare the states, apply all the operations in the circuit and measure. Then it will do it all again. Each measurement will be independent of the others as the system is restarted each time. This way the serious problem of decoherence only apply within each run, and the number of measurements is only limited by the patience of the scientists operating the quantum computer.

IV. RESULTS AND DISCUSSION

In this section we present the results for the Hubbard model and the pairing model of Eqs. (2) and (3), respectively, and compare the simulations to exact diagonalization results. In Fig. 4 we see the resulting probability distribution from the simulated measurements, giving us the eigenvalues of the pairing model with six degenerate energy levels and from zero to 12 particles. The pairing strength was set to $g = 1$. The eigenvalues from the exact solutions of these many-particle problems are 0, -1, -2, -3, -4, -5, -6, -8, -9, -12. All the eigenvalues are not seen as this is the probability distribution resulting from one random input state. A different random input state in each run could be implemented on an actual quantum computer. These are results for the degenerate model, where the single-particle energies of the doubly degenerate levels are set to zero for illustrate purposes only, since analytic formula are available for the exact eigenvalues. The block diagonal structure of the pairing Hamiltonian has not been used to our advantage in this straightforward simulation as the qubit basis includes all particle numbers.

We have also performed tests of the algorithm for the non-degenerate case, with excellent agreement with our diagonalization codes, see discussion in Ref. [34]. This is seen in Fig. 5 where we have simulated the pairing model with four energy levels with a total possibility of eight fermions. We have chosen $g = 1$ and $d = 0.5$, so this is a model with low degeneray and since the dimension of the system is $2^8 = 256$ there is a lot of different eigenvalues. To find the whole spectrum one would have to employ some of the techniques discussed in subsection III B.

A. Number of work qubits versus number of simulation qubits

The largest possible amount of different eigenvalues is 2^s, where s is the number of simulation qubits. The

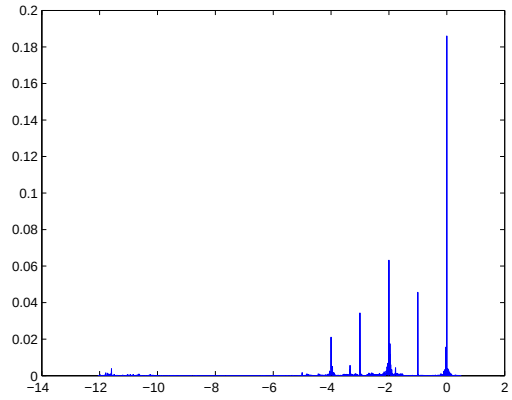


FIG. 4: Resulting probability distribution from the simulated measurements, giving us the eigenvalues of the pairing model with six degenerate energy levels with a total possibility of 12 particles and pairing strength $g = 1$. The correct eigenvalues are 0, −1, −2, −3, −4, −5, −6, −8, −9, −12. All the eigenvalues are not seen as this is the probability distribution resulting from one random input state. A different random input state in each run could be implemented on an actual quantum computer and would eventually yield peaks of height corresponding to the degeneracy of each eigenvalue.

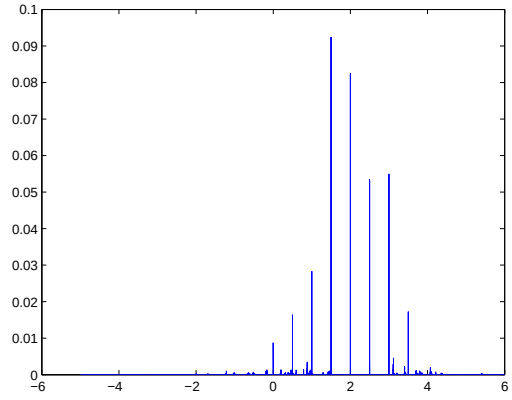


FIG. 5: The eigenvalues of the non-degenerate pairing model with four energy levels with a total possibility of 8 particles, the level spacing d is 0.5 and the pairing strength g is 1. The correct eigenvalues are obtained from exact diagonalization, but in this case there is a multitude of eigenvalues and only some eigenvalues are found from this first simulation.

resolution in the energy spectrum we get from measuring upon the work qubits is 2^w, with w the number of work qubits. Therefore the resolution per eigenvalue in a non-degenerate system is 2^{w-s}. The higher the degeneracy the less work qubits are needed.

In Fig. 6 we see the results for the Hubbard model Eq. (2) with $\epsilon = 1$, $t = 0$ and $U = 1$. The reason we chose $t = 0$ was just because of the higher degeneracy and therefore fewer eigenvalues. The number of work qubits is 16 and the number of simulation qubits is eight for a total of 24 qubits. The difference between work qubits and simulation qubits is eight which means there are 2^8 possible energy values for each eigenvalue. Combining that with the high degeneracy we get a very sharp resolution. The correct eigenvalues with degeneracies are obtained from exact diagonalization of the Hamiltonian, the degeneracy follows the eigenvalue in paranthesis: 0(1), 1(8), 2(24), 3(36), 4(40), 5(48), 6(38), 7(24), 8(24), 9(4), 10(8), 12(1). We can clearly see that even though we have a random input state, with a random superposition of the eigenvectors, there is a correspondence between the height of the peaks in the plot and the degeneracy of the eigenvalues they represent.

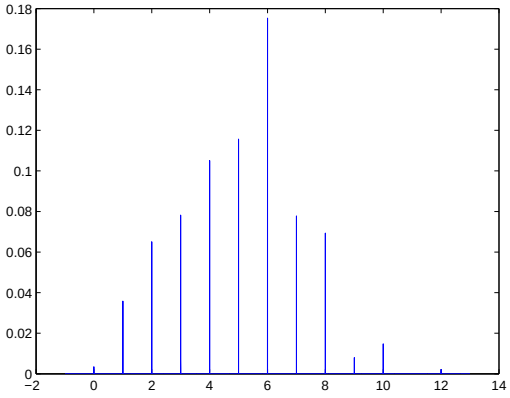


FIG. 6: The energy levels of the Hubbard model of Eq. (2), simulated with a total of 24 qubits, of which eight were simulation qubits and 16 were work qubits. In this run we chose $\epsilon = 1$, $t = 0$ and $U = 1$. The reason we chose $t = 0$ was just because of the higher degeneracy and therefore fewer eigenvalues. The correct eigenvalues are obtained from exact diagonalization, with the level of degeneracy following in paranthesis: 0(1), 1(8), 2(24), 3(36), 4(40), 5(48), 6(38), 7(24), 8(24), 9(4), 10(8), 12(1).

B. Number of time intervals

The number of time intervals, I, is the number of times we must implement the time evolution operator in order to reduce the error in the Trotter approximation [46, 47, 48], see Eq. (6). In our program we have only implemented the simplest Trotter approximation and in our case we find that we do not need a large I before the error is small enough. In Fig. 6 I is only one, but here we have a large number of work qubits. For other or larger systems it might pay off to use a higher order Trotter approximation. The total number of operations that have to be done is a multiple of I, but this number

also increases for higher order Trotter approximations, so for each case there is an optimal choice of approximation.

In Figs. 7 and 8 we see the errors deriving from the Trotter approximation, and how they are reduced by increasing the number of time intervals. The results in this figure are for the degenerate pairing model with 24 qubits in total, and ten simulation qubits with $d = 0$ and $g = 1$. In Fig. 7 we had $I = 1$ while in Fig. 8 I was set to ten. Both simulations used the same starting state. The errors are seen as the small spikes around the large ones which represent some of the eigenvalues of the system. The exact eigenvalues are 0, -1, -2, -3, -4, -5, -6, -8, -9.

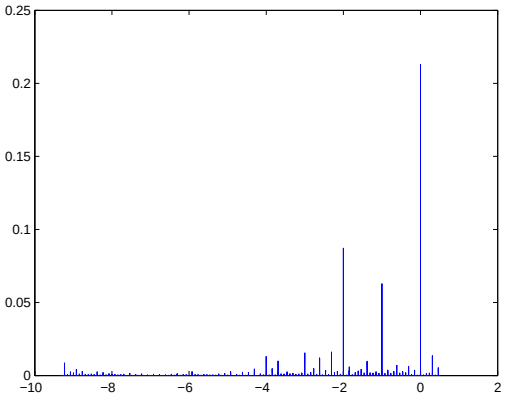


FIG. 7: Pairing model simulated with 24 qubits, where 14 were simulation qubits, i.e. there are 14 available quantum levels, and 10 were work qubits. The correct eigenvalues are 0, -1, -2, -3, -4, -5, -6, -8, -9. In this run we did not divide up the time interval to reduce the error in the Trotter approximation, i.e., $I = 1$.

C. Number of operations

Counting the number of single-qubit and $\sigma_z\sigma_z$ operations for different sizes of systems simulated gives us an indication of the decoherence time needed for different physical realizations of a quantum simulator or computer. The decoherence time is an average time in which the state of the qubits will be destroyed by noise, also called decoherence, while the operation time is the average time an operation takes to perform on the given system. Their fraction is the number of operations possible to perform before decoherence destroys the computation. In table I we have listed the number of gates used for the pairing model, H_P, and the Hubbard model, H_H, for different number of simulation qubits.

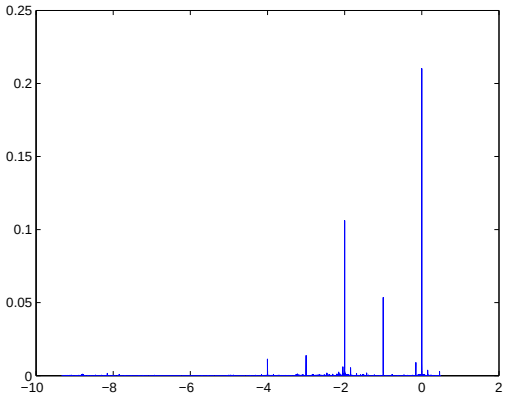


FIG. 8: Pairing model simulated with 24 qubits, where 14 were simulation qubits, i.e. there are 14 available quantum levels, and 10 were work qubits. The correct eigenvalues are 0, -1, -2, -3, -4, -5, -6, -8, -9. In this run we divided the time interval into 10 equally space parts in order to reduce the error in the Trotter approximation, i.e., $I = 10$.

	$s = 2$	$s = 4$	$s = 6$	$s = 8$	$s = 10$	$s = 12$
H_P	9	119	333	651	1073	1598
H_H	9	51	93	135	177	219

TABLE I: Number of two-qubit gates used in simulating the time evolution operator of the pairing model, H_P, and the Hubbard model, H_H, for different number of simulation qubits s.

V. CONCLUSION

In this article we have shown explicitly how the Jordan-Wigner transformation is used to simulate any many-body fermionic Hamiltonian by two-qubit operations. We have shown how the simulation of such Hamiltonian terms of products of creation and annihilation operators are represented by a number of operations linear in the number of qubits. To perform efficient quantum simulations on quantum computers one needs transformations that take the Hamiltonian in question to a set of operations on the qubits simulating the physical system. An example of such a transformation employed in ths work, is the Jordan-Wigner transformation. With the appropriate transformation and relevant gates or quantum circuits, one can taylor an actual quantum computer to simulate and solve the eigenvalue and eigenvector problems for different quantum systems. Specialized quantum simulators might be more efficient in solving some problems than others because of similarities in algebras between physical system of qubits and the physical system simulated.

We have limited the applications to two simple and well-studied models that provide, via exact eigenvalues,

a good testing ground for our quantum computing based algorithm. For both the pairing model and the Hubbard model we obtain an excellent agreement. We plan to extend the area of application to quantum mechanical studies of systems in nuclear physics, such as a comparison of shell-model studies of oxygen or calcium isotopes where the nucleons are active in a given number of single-particle orbits [7, 16]. These single-particle orbits have normally a higher degeneracy than 2, a degeneracy studied here. However, the algorithm we have developed allows for the inclusion of any degeneracy, meaning in turn that with a given interaction V_{ijkl} and single-particle energies, we can compare the nuclear shell-model (configuration interaction) calculations with our algorithm.

Acknowledgment

This work has received support from the Research Council of Norway through the center of excellence program.

APPENDIX: USEFUL RELATIONS

We list here some useful relations involving different σ matrices,

$$\sigma_x\sigma_z = -i\sigma_y, \quad \sigma_z\sigma_x = i\sigma_y, \quad [\sigma_x,\sigma_z] = -2i\sigma_y, \quad \text{(A.1)}$$

$$\sigma_x\sigma_y = i\sigma_z, \quad \sigma_y\sigma_x = -i\sigma_z, \quad [\sigma_x,\sigma_y] = 2i\sigma_z, \quad \text{(A.2)}$$

and

$$\sigma_y\sigma_z = i\sigma_x, \quad \sigma_z\sigma_y = -i\sigma_x, \quad [\sigma_y,\sigma_z] = 2i\sigma_x. \quad \text{(A.3)}$$

For any two non-equal σ-matrices a and b we have

$$aba = -b. \quad \text{(A.4)}$$

The Hermitian σ-matrices σ_x, σ_y and σ_z result in the identity matrix when squared

$$\sigma_x^2 = \mathbf{1}, \quad \sigma_y^2 = \mathbf{1}, \quad \sigma_z^2 = \mathbf{1}, \quad \text{(A.5)}$$

which can be used to obtain simplified expressions for exponential functions involving σ-matrices

$$e^{\pm i\alpha\sigma} = \cos(\alpha)\mathbf{1} \pm i\sin(\alpha)\sigma. \quad \text{(A.6)}$$

The equations we list below are necessary for the relation between a general unitary transformation on a set of qubits with a product of two-qubit unitary transformations. We have the general equation for $a, b \in \{\sigma_x, \sigma_y, \sigma_z\}$, where $a \neq b$.

$$\begin{aligned} e^{-i\pi/4a}be^{i\pi/4a} &= \frac{1}{2}(\mathbf{1} - ia)b(\mathbf{1} + ia) \\ &= \frac{1}{2}(b + aba + i[b,a]) \\ &= \frac{i}{2}[b,a]. \end{aligned} \quad \text{(A.7)}$$

The more specialized equations read

$$e^{-i\pi/4\sigma_x}\sigma_z e^{i\pi/4\sigma_x} = -\sigma_y, \quad \text{(A.8)}$$

$$e^{-i\pi/4\sigma_y}\sigma_z e^{i\pi/4\sigma_y} = \sigma_x, \quad \text{(A.9)}$$

$$e^{-i\pi/4\sigma_z}\sigma_x e^{i\pi/4\sigma_z} = \sigma_y, \quad \text{(A.10)}$$

$$e^{-i\pi/4\sigma_z}\sigma_y e^{i\pi/4\sigma_z} = -\sigma_x. \quad \text{(A.11)}$$

We need also different products of the operatorσ_z with the raising and lowering operators

$$\sigma_+\sigma_z = -\sigma_+ \quad \text{(A.12)}$$

$$\sigma_z\sigma_+ = \sigma_+, \quad \text{(A.13)}$$

$$\sigma_-\sigma_z = \sigma_-, \quad \text{(A.14)}$$

$$\sigma_z\sigma_- = -\sigma_-. \quad \text{(A.15)}$$

$$\quad \text{(A.16)}$$

[1] R. J. Bartlett. Many-body perturbation theory and coupled-cluster theory for electron correlations in molecules. *Ann. Rev. Phys. Chem.*, 32:359, 1981.

[2] D. J. Dean and M. Hjorth-Jensen. Coupled-cluster approach to nuclear physics. *Phys. Rev. C*, 69:054320, 2004.

[3] T. Helgaker, P. Jørgensen, and J. Olsen. *Molecular Electronic Structure Theory. Energy and Wave Functions.* Wiley, Chichester, 2000.

[4] D. M. Ceperley. Path integrals in the theory of condensed helium. *Rev. Mod. Phys.*, 67:279, 1995.

[5] S.E. Koonin, D.J. Dean, and K. Langanke. *Phys. Rep.*, 278:1, 1997.

[6] B. S. Pudliner, V. R. Pandharipande, J. Carlson, Steven C. Pieper, and R. B. Wiringa. Quantum monte carlo calculations of nuclei with $A \leq 7$. *Phys. Rev. C*, 56:1720, 1997.

[7] M. Hjorth-Jensen, T. T. S. Kuo, and E. Osnes. Realistic effective interactions for nuclear systems. *Phys. Rep.*, 261:125, 1995.

[8] I. Lindgren and J. Morrison. *Atomic Many-Body Theory.* Springer, Berlin, 1985.

[9] J. P. Blaizot and G. Ripka. *Quantum theory of Finite Systems.* MIT press, Cambridge, USA, 1986.

[10] W. H. Dickhoff and D. Van Neck. *Many-Body Theory exposed!* World Scientific, 2005.

[11] U. Schollwock. The density-matrix renormalization group. *Rev. Mod. Phys.*, 77:259, 2005.

[12] S. R. White. Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.*, 69:2863, 1992.

[13] R. J. Bartlett, V. F. Lotrich, and I. V. Schweigert. Ab initio density functional theory: The best of both worlds? *J. Chem. Phys.*, 123:062205, 2005.

[14] D. Van Neck, S. Verdonck, G. Bonny, P. W. Ayers, and M. Waroquier. Quasiparticle properties in a density-functional framework. *Phys. Rev. A*, 74:042501, 2006.

[15] K. Peirs, D. Van Neck, and M. Waroquier. Algorithm to derive exact exchange-correlation potentials from correlated densities in atoms. *Phys. Rev. A*, 67:012505, 2003.

[16] E. Caurier, G. Martinez-Pinedo, F. Nowacki, A. Poves, and A. P. Zuker. The shell model as a unified view of nuclear structure. *Rev. Mod. Phys.*, 77:427, 2005.

[17] M. Horoi, B. A. Brown, T. Otsuka, M. Honma, and T. Mizusaki. Shell model analysis of the ^{56}Ni spectrum in the full pf model space. *Phys. Rev. C*, 73:061305, 2006.

[18] P. Navratil and E. Caurier. Nuclear structure with accurate chiral perturbation theory nucleon-nucleon potential: Application to ^{6}Li and ^{10}B. *Phys. Rev. C*, 69:014311, 2004.

[19] R. R. Whitehead, A. Watt, B. J. Cole, and I. Morrison. Computational methods for shell-model calculations. *Adv.Nucl.Phys.*, 9:123, 1977.

[20] D. S. Abrams and S. Lloyd. Simulation of many-body fermi systems on a universal quantum computer. *Phys. Rev. Lett.*, 79:2586, 1997.

[21] D. S. Abrams and S. Lloyd. Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors. *Phys. Rev. Lett.*, 83:5162, 1999.

[22] R. P. Feynman. Simulating physics with computers. *Int. J. Theor. Phys.*, 21:467, 1982.

[23] R. P. Feynman. Quantum mechanical computers. *Foundations. Phys.*, 16:507, 1986.

[24] G. Ortiz, J. E. Gubernatis, E. Knill, and R. Laflamme. Simulating fermions on a quantum computer. *Comp. Phys. Comm.*, 146:302, 2002.

[25] R. Somma, G. Ortiz, J. E. Gubernatis, E. Knill, and R. Laflamme. Simulating physical phenomena by quantum networks. *Phys. Rev. A*, 65:042323, 2002.

[26] R. D. Somma. *Quantum Computation, Complexity, and Many-Body Physics*. PhD thesis, Instituto Balseiro, S.C. de Bariloche, Argentina and Los Alamos National Laboratory, Los Alamos, U.S.A., 2005.

[27] K. R. Brown, R. J. Clark, and I. L. Chuang. Limitations of quantum simulation examined by simulating a pairing hamiltonian using nuclear magnetic resonance. *Phys. Rev. Lett.*, 97:050504, 2006.

[28] X. Yang, A. Wang, F. Xu, and J. Du. Experimental simulation of a pairing hamiltonian on an NMR quantum computer. *Chem. Phys. Lett.*, 422:20, 2006.

[29] J. Hubbard. Electron correlations in narrow energy bands. *Proc. R. Soc. A*, 276:238, 1963.

[30] I. Talmi. *Simple Models of Complex Nuclei*. Harwood Academic Publishers, 1993.

[31] R. W. Richardson. A restricted class of exact eigenstates of the pairing-force Hamiltonian *Phys. Lett.*, 3:277, 1963.

[32] R. W. Richardson and N. Sherman. Exact Eigenstates of the Pairing-Force Hamiltonian. I *Nucl. Phys.*, 52:221, 1964.

[33] R. W. Richardson. Exact Eigenstates of the Pairing-Force Hamiltonian. II *J. Math. Phys.*, 6:1034, 1965.

[34] D. J. Dean and M. Hjorth-Jensen. Pairing in nuclear systems: from neutron stars to finite nuclei. *Rev. Mod. Phys.*, 75:607, 2003.

[35] J. Dukelsky, S. Pittel, and G. Sierra. Exactly solvable Richardson-Gaudin models for many-body quantum systems. *Rev. Mod. Phys.*, 76:643, 2004.

[36] P.Benioff. The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines. *J. Stat. Phys.*, 22:563, 1980.

[37] M. A. Nielsen and I. L. Chaung. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[38] G. Ortiz, J. E. Gubernatis, E. Knill, and R. Laflamme. Quantum algorithms for fermionic simulations. *Phys. Rev. A*, 64:022319, 2001.

[39] P. Dargis and Z. Maassarani. Fermionization and Hubbard models. *Nucl. Phys. B*, 535:681, 1998.

[40] Steven C. Pieper, V. R. Pandharipande, R. B. Wiringa, and J. Carlson. Realistic models of pion-exchange three-nucleon interactions. *Phys. Rev. C*, 64:014001, 2001.

[41] P. Navrátil and W. E. Ormand. Ab initio shell model calculations with three-body effective interactions for p-shell nuclei. *Phys. Rev. Lett.*, 88:152502, 2002.

[42] G. Hagen, T. Papenbrock, D. J. Dean, A. Schwenk, M. Włoch, P. Piecuch, and A. Nogga. Coupled-cluster theory for three-body hamiltonians. arXiv:nucl-th/0704.2854, 2007.

[43] E. Ovrum. Quantum computing and many-body physics. Master's thesis, University of Oslo, 2003.

[44] L.-A. Wu, M. S. Byrd, and D. A. Lidar. Polynomial-time simulation of pairing models on a quantum computer. *Phys. Rev. Lett.*, 89:057904, 2002.

[45] G.H. Golub and C.F. Van Loan. *Matrix Computations*. John Hopkins University Press, 1996.

[46] H. F. Trotter. On the product of semi-groups of operators. *Proc. Am. Math. Soc.*, 10:545, 1959.

[47] M. Suzuki. Transfer-matrix method and monte carlo simulation in quantum spin systems. *Phys. Rev. B*, 31:2957, Mar 1985.

[48] M. Suzuki. Decomposition formulas of exponential operators and Lie exponentials with some applications to quantum mechanics and statistical physics. *J. Math. Phys.*, 26:601, 1985.