Characterizing quantum correlations: the genuine multiparticle negativity as entanglement monotone

Dissertation
zur Erlangung des Grades eines Doktors
der Naturwissenschaften

vorgelegt von
Dipl.-Phys. Martin Hofmann

eingereicht bei der Naturwissenschaftlich-Technischen Fakultät
der Universität Siegen
Siegen 2014

Gutachter:

- Prof. Dr. Otfried Gühne
- Prof. Dr. Jens Siewert

Datum der mündlichen Prüfung: 28.05.2014

Prüfer:

- Prof. Dr. Christian Gutt
- Prof. Dr. Otfried Gühne (Vorsitz der Prüfungskommision)
- Prof. Dr. Thomas Mannel
- Prof. Dr. Christof Wunderlich

Abstract

Multiparticle entanglement is a useful resource in quantum information processing. It is involved in some quantum key distribution protocols, quantum metrology and many other physical applications and phenomena and can be experimentally observed in various quantum systems. Having said this, its classification, detection and especially its quantification is quite challenging. To this day there exists no general mixed state measure for genuine multiparticle entanglement, which can be computed and analytically treated at the same time.

In this thesis the analytical characterisation of genuine multiparticle entanglement in quantum systems using the computable genuine multiparticle negativity as entanglement measure is provided. Furthermore, the notion of stabiliser states, which are families of symmetric genuine multiparticle entangled states, is generalised and a useful method to exploit local symmetries to speed up the computation of the investigated entanglement measure is provided.

In the first part, after a short introduction, the genuine multiparticle negativity, which is defined as an optimisation problem known as semidefinite programming problem, is investigated. It is discussed, how this entanglement measure can be characterised in an analytical way. First, it is shown that the genuine multiparticle negativity with an appropriate renormalisation can be considered as coming from a mixed convex roof construction. Using this result, its analytical value for generalised n-qubit Greenberger–Horne–Zeilinger-diagonal states and four-qubit cluster-diagonal states is determined.

In the second part of this thesis, the genuine multiparticle negativity is used to study the scaling and spatial distribution of genuine multiparticle entanglement in three- and four-spin reduced states of a one-dimensional spin model at its quantum phase transition. At the quantum phase transition of the one dimensional XY-model, which can be studied with analytic rigour, a logarithmic divergence is observed in the first derivative of the genuine multiparticle negativity. It is then shown that genuine three- and four-particle entanglement obeys finite-size scaling and that the genuine three-particle entanglement has a finite spatial range.

In the third part, a generalisation to the so-called stabiliser formalism is introduced. The idea of characterising pure stabiliser states via a maximal commuting group of local symmetries is extended to mixed states. These so-called stabilised states are then characterised by a not necessarily maximal commuting group of local symmetries and unite graph diagonal states and X-states in a single framework. Finally, the ambiguity of local base change is studied and a method to obtain a classification of the underlying symmetry groups into equivalence classes under local Clifford operations is provided.

In the last part of this thesis, it is shown how symmetries of a state may be used to simplify the optimisation problem defining the genuine multiparticle negativity. A relationship between the symmetry of a state and the internal structure of the optimisation is established that can be used to reduce the number of variables in the optimisation problem. The latter is an instance of a semidefinite programming problem for which efficient numerical optimisation algorithms with a certified solution exist.

Zusammenfassung

Echte Mehrteilchenverschränkung ist eine wichtige Ressource in der Quanteninformation. Sie ist ein wesentlicher Bestandteil in einigen Quanten-Schlüsselaustausch-Protokollen, in der Quantenmetrologie und vielen anderen physikalischen Anwendungen und Phänomenen. Nichtsdestotrotz ist ihre Klassifizierung, Detektion und vor ihre allem Quantifizierung enorm herausfordernd. Bisher existiert kein allgemeines gemischtes Zustandsmaß, welches sowohl numerisch berechenbar als auch analytisch behandelbar ist.

Diese Arbeit behandelt die analytische Charakterisierung echter Mehrteilchenverschränkung in Quantensystemen mit der numerisch berechenbaren echten Mehrteilchennegativität als Verschränkungsmaß. Sie verallgemeinert den Begriff stabilisierter Zustände, die Familien symmetrischer echt mehrteilchenverschränkter Zustände definieren, und zeigt auf, wie lokale Symmetrien genutzt werden können um die Berechnung des Maßes zu optimieren.

Nach einer kurzen Einführung wird im ersten Teil die echte Mehrteilchennegativität untersucht. Diese ist über ein bekanntes Optmimierungsproblem, ein sogenanntes semidefinites Programmierungsproblem, definiert. Es wird gezeigt, dass dieses Verschränkungsmaß auch analytisch behandelt werden kann. Hierfür wird zuerst bewiesen, dass die echte Mehrteilchennegativität richtig renormalisiert auch über ein konvexes Dach definiert werden kann. Dieses Resultat wird anschließend verwendet um analytische Werte für verallgemeinerte n-Qubit Greenberger-Horne-Zeilinger-diagonale und vier Qubit clusterdiagonale Zustände zu erhalten.

Im zweiten Teil dieser Arbeit wird die echte Mehrteilchennegativität verwendet um das Skalenverhalten und die räumliche Verteilung echter Mehrteilchenverschränkung in drei- und vierteilchenreduzierten Zuständen eines eindimensionalen Spinmodels nahe eines Quantenphasenübergangs zu studieren. Unter Verwendung des exakt lösbaren eindimensionalen XY-Models wird eine logarithmische Divergenz in der ersten Ableitung der echten Mehrteilchennegativität am Quantenphasenübergang festgestellt. Es wird gezeigt, dass die echte Drei- und Vierteilchenverschränkung mit der Größe des XY-Systems skaliert und dass die echte Dreiteilchenverschränkung räumlich begrenzt auftritt.

Im vorletzten Teil wird eine Verallgemeinerung des Stabilisatorformalismus eingeführt. Die Idee, reine Stabilisatorzustände über eine maximal kommutierende Gruppe lokaler Symmetrien zu charakterisieren, wird auf gemischte Zustände erweitert. Diese sogenannten stabilisierten Zustände werden über eine nicht notwendigerweise maximal kommutierende Gruppe lokaler Symmetrien charakterisiert und vereinen X-Zustände und graphendiagonale Zustände in einem Konzept. Abschließend werden die Symmetriegruppen auf Equivalenz unter lokalen Basiswechseln hin untersucht. Hierzu wird eine Methode aufgezeigt, welche die Gruppen in Equivalenzklassen unter lokalen Cliffordoperationen aufteilt.

Im letzten Teil der Arbeit wird gezeigt, wie die Symmetrien eines Zustands dazu beitragen können das Optimierungsproblem zu vereinfachen, das die echte Mehrteilchennegativität definiert. Die Optimierung ist ein Problem des semidefiniten Programmierens, für das effiziente numerische Algorithmen zur Verfügung stehen, welche eine zertifizierte Lösung liefern. Es wird ein Zusammenhang zwischen den Symmetrien und der internen Struktur des Optimierungsproblems aufgezeigt, welcher dann zu einer Reduzierung der Zahl freier Parameter des Optimierungsproblems verwendet werden kann.

Contents

Introduction

Soon after quantum mechanics was developed in the early last century, Einstein, Podolsky and Rosen (EPR) [1] and Schrödinger [2] described a "spooky" superposed state in a composite quantum system. In their famous argument, EPR used these "entangled" states to question the completeness of quantum mechanics. It was in 1964, when Bell took the EPR assumption of a "local hidden variable" model for quantum mechanics as a working hypothesis to prove that entangled states can be used to experimentally rule out such a description [3]. Nearly 20 years later, Aspect et al. [4, 5] performed the first convincing experiment to violate Bell's inequalities, providing strong arguments against a local hidden variable model for quantum mechanics.

During the 80's, it was found that entanglement is the key ingredient of several interesting information theoretical applications. Nowadays, it is viewed as a resource rather than a spooky phenomena. It can be used to distribute a key between two distant parties [6] for secure communication. In dense coding entanglement is crucial to transmit two bits of classical information by sending only one quantum bit [7]. Furthermore, it has been pointed out that an entangled state is necessary to teleport an unknown state of one quantum system to another [8].

Another field, where entanglement is believed to be essential, is quantum computation [9], where the laws of quantum mechanics are exploited to reduce the complexity of computational tasks compared to known classical strategies. Prominent examples are the Deutsch algorithm [10] that solves a decision problem, and the Shor algorithm [11] that factorises prime numbers.

For quite some time now, entanglement of more than two parties has been the subject of investigations and there is continuous progress in understanding and utilizing its rich structure. In 1989, Greenberger, Horne and Zeilinger showed that entangled states of more than two parties can be used to extend the statements of Bell's theorem [12]. Later, it was pointed out that highly entangled states in multiparticle systems can be used for quantum computation in a measurement based scheme [13]. Further, one can use multiparticle entanglement as a resource for high precision measurements by estimating a parameter with fewer measurements than any classical strategy [14].

Nowadays, it is one of the main theoretical challenges to detect and quantify quantum correlations such as "genuine multiparticle entanglement", where *all* parts of a quantum system are entangled with each other. Recent progress has been made by the introduction of the so-called "genuine multiparticle negativity" [15], which is a measure of genuine multiparticle entanglement. Using this tool one can identify genuine multiparticle entanglement in photon states of the triple Compton effect [16, 17], the high-energy process in which a photon splits into three after colliding with a free electron. One can use it to track the dynamics of entanglement of a multiparticle open quantum system from genuine multiparticle entanglement to full separability [18]. Additionally, it can be utilised to study the robustness of different types of entangled states against decoherence due to the interaction with the environment [19].

To reduce the complexity in classifying and detecting entanglement one often restricts studies to suitable theoretical "laboratories". These are usually families of quantum states that can be described by a small number of parameters. An instance is the family of stabiliser states [20, 21]. These highly entangled states were originally introduced in the context of quantum computation. They can be used as universal resource in measurement based quantum computation and protect quantum information from errors to achieve fault tolerant quantum computation [21, 22]. Beyond that, stabiliser states have been proven very useful to investigate entanglement in the multiparticle setting [23–25] and to study state equivalence under local basis transformation [26–29].

More realistic scenarios are considered in condensed matter systems. An interesting instance are the ground and thermal states of spin models close to a quantum phase transition. Here, the investigation of entanglement has shed some light on the nature of quantum phase transitions [30–40] and at the same time

has provided new insights into the simulation of spin models [41, 42]. In such systems the entanglement in the reduced state of two particles of a multiparticle system [30, 31] and the entanglement between one block of particles and the remainder of the system [32] have been studied. More recently, these investigations were extended to a qualitative study of genuine multiparticle entanglement in the reduced states of a few particles [43–49].

In this thesis, we present an analytical study of the genuine multiparticle negativity. We show that a renormalised version of this entanglement measure can be expressed as mixed convex roof of the minimum of bipartite negativities [50, 51]. These mixed convex roofs were already studied in the context of entanglement quantification in the bipartite setting in Refs. [52–55]. In our case and contrast to the usual pure state convex roof constructions, the renormalised genuine multiparticle negativity can be efficiently computed using semidefinite programming. We derive analytic expressions for the genuine multiparticle negativity for two different state families. These are the GHZ-diagonal n-qubit and the cluster-diagonal four-qubit states. These analytic formulas, given in terms of the fidelities of the GHZ and cluster basis, also provide lower bounds of the genuine multiparticle entanglement for general mixed quantum states.

Using the genuine multiparticle negativity and a recently proposed separability algorithm [56, 57], we also study the scaling and spatial distribution of genuine multiparticle entanglement at a quantum phase transition in a one-dimensional spin model. For the reduced three- and four-particle states we demonstrate that its derivative diverges at the critical point. For both cases we show that the entanglement obeys finite-size scaling, which is used to compute the critical exponent for the infinite system from finite-size data.

Additionally, we provide a generalisation of the stabiliser formalism and introduce the notion of stabilised states. Similar to stabiliser states, these states are characterised by a symmetry group. In the stabiliser formalism this symmetry group induces a certain set of mixed states, which is diagonal in the stabiliser basis. In our generalisation the restrictions on the stabilised states due to symmetry are in general weaker and hence states are solely block-diagonal in the right basis. Similar to stabiliser states, however, many properties of the state can be derived directly from the underlying symmetry group. Subsequently, a method will be provided to identify and classify stabilised states with respect to LC-equivalence. Finally, the symmetry group is used to simplify the problem of quantifying entanglement using the genuine multiparticle negativity.

This thesis is structured as follows: First, in Chapter 1, the basic notions and definitions are introduced. On this basis, an analytical study of the genuine multiparticle negativity is presented in Chapter 2. In Chapter 3, the genuine multiparticle negativity is used to study the scaling and spatial distribution of genuine multiparticle entanglement in three- and four-spin reduced states of a one-dimensional spin model at its quantum phase transition. Thereafter, a generalisation of the so-called stabiliser formalism is introduced and a method to obtain a classification of the underlying symmetry groups into equivalence classes under local Clifford operations is provided in Chapter 4. In Chapter 5, it is shown how symmetries of a state may be exploited to simplify the optimisation problem defining the genuine multiparticle negativity. At the end the main results are summarised and an outlook is given.

Chapter 1

Basic concepts

Whatever you do will be insignificant, but it is very important that you do it.
Mahatma Gandhi

In this chapter we[1] review several basic concepts related to entanglement. These are bipartite and multiparticle entanglement, positive partial transposed (PPT) states, PPT mixtures, entanglement witnesses, entanglement monotones, the bipartite and the genuine multiparticle negativity and the stabiliser formalism. Note that it is not within the scope of this thesis to give a complete overview over all the subjects. It will solely introduce the most important notions, which are necessary to understand this thesis. Following this approach we will omit most of the technical details and proofs and refer the interested reader to the literature on the subjects in Refs. [15, 20, 58–61, 61].

1.1 Bipartite entanglement

In this section the notion of bipartite entanglement in pure and mixed quantum systems will be introduced. Note that within this thesis we will consider finite dimensional Hilbert spaces only.

1.1.1 Pure state

Consider two finite dimensional arbitrary quantum systems. These systems are controlled by two physicists named Alice and Bob[2]. Then the physical states (pure states) of Alice and Bob are described by normalised vectors $|\psi_A\rangle$ and $|\psi_B\rangle$ in the Hilbert spaces $\mathcal{H}_A$ and $\mathcal{H}_B$, respectively, where the dimensions of the Hilbert spaces $\dim(\mathcal{H}_A) = n$ and $\dim(\mathcal{H}_B) = m$ are arbitrary.

Let $\{|\psi_A^{(i)}\rangle\}$ be a basis of $\mathcal{H}_A$ and $\{|\psi_B^{(j)}\rangle\}$ be a basis of $\mathcal{H}_B$. Then any vector of the joint system $\mathcal{H}_{AB} = \mathcal{H}_A \otimes \mathcal{H}_B$ can be written as a superposition

$$|\psi_{AB}\rangle = \sum_{i=1,j=1}^{n,m} c_{ij} |\psi_A^{(i)}\rangle \otimes |\psi_B^{(j)}\rangle, \tag{1.1}$$

with the complex matrix of coefficients c_{ij} and total dimensions of the joint Hilbert space $\dim(\mathcal{H}_{AB}) = nm$. To improve readability one often omits the tensor product and shortens the notation to $|a\rangle \otimes |b\rangle \equiv |a\rangle|b\rangle \equiv |ab\rangle$.

Given any bipartite pure state $|\psi_{AB}\rangle \in \mathcal{H}_{AB}$, it is called separable if it can be written as a product of two vectors of the respective subsystems $|\psi_A\rangle \in \mathcal{H}_A$ and $|\psi_B\rangle \in \mathcal{H}_B$

$$|\psi_{AB}\rangle = |\psi_A\rangle|\psi_B\rangle. \tag{1.2}$$

If the state cannot be written in the above form it is called entangled.

[1] Throughout this work "we" refers to the author and the reader ("pluralis auctoris"). An exception constitutes the Chapters 2 and 3, where "we" refers to the author and its coauthors.

[2] Due to the close connection of cryptography and quantum information one chooses the same naming convention for different parties as in the original work on the RSA public key cryptosystem in Ref. [62].

For a general pure state as given by Eq. (1.1) it is known how to determine if a state is separable or entangled. Given the rank of its matrix of coefficients $(C)_{ij} = c_{ij}$ the state is separable if and only if that rank equals one. The rank $r(|\psi_{AB}\rangle)$ of C is also denoted as the Schmidt rank of the vector, which is upper bounded by $\min(n, m)$. It can be alternatively obtained by calculation of the rank of the reduced states $\varrho_A = \mathrm{tr}_B(|\psi_{AB}\rangle\langle\psi_{AB}|)$ and $\varrho_B = \mathrm{tr}_A(|\psi_{AB}\rangle\langle\psi_{AB}|)$, which are given by partially tracing out either system B or A.

In essence, entanglement is a direct consequence of the composition of quantum systems and the superposition principle, which clearly delimits quantum theory from classical physical theories, which lack the quantum mechanical superposition principle. This is best illustrated in the preparation procedure of a separable state. To create such a state Alice and Bob simply prepare the states $|\psi_A\rangle$ and $|\psi_B\rangle$ locally in their laboratories. The result is a separable state as given by Eq. (1.2). If subsequently Alice and Bob perform measurements on their subsystems, the measurement outcomes factorise, i.e., the results of Alice and Bob are uncorrelated in terms of their mathematical correlation coefficients.

The creation of an entangled state on the other hand requires some sort of interaction between the quantum systems of Alice and Bob or at least some interaction with some other entangled system. Depending on a subsequent measurement the outcomes of Alice and Bob can then be highly correlated.

1.1.2 Statistical local operations and classical communication

Referring again to the two experimenters Alice and Bob in spatially separated laboratories a natural question to ask is: What transformations are possible, if both experimenters are limited to quantum operations[3] on their parts of a shared state and classical communication only? I.e., given a bipartite state shared between Alice and Bob, both parties are allowed the following: Both parties may use any number of additional quantum systems to combine with their local quantum state. On such an enlarged local system they may perform arbitrary measurements, the outcomes of which they are free to communicate to the other party. During the process they are also allowed to apply quantum transformations on their subsystem. All possible protocols, which only consist of the steps above are called statistical local operations and classical communication (SLOCC) protocols.

Two states which can be transformed into each other via any SLOCC protocol are said to be SLOCC-equivalent. Cast into mathematical terms two states $|\psi\rangle$ and $|\phi\rangle$ are SLOCC-equivalent iff they are related by an invertible local operation [63]

$$|\psi\rangle = A \otimes B|\phi\rangle. \tag{1.3}$$

Note that the notions of SLOCC protocols and SLOCC-equivalence naturally generalises to more parties.

Furthermore, the concept of SLOCC-equivalence is closely related to entanglement as no separable state is SLOCC-equivalent to any entangled state and if any SLOCC protocol is applied to an entangled state, then entanglement does not increase.

1.1.3 Mixed states

Unfortunately, in most experimental cases one unavoidably has to deal with mixed rather than pure quantum states. That is because in a realistic experimental situation one might not have full control over all degrees of freedom and thus ends up in a situation, where in each experimental run the system is in some pure state $|\psi_i\rangle \in \mathcal{H}$ from a given ensemble. Such a mixed state is described by the density matrix

$$\varrho = \sum_i p_i |\psi_i\rangle\langle\psi_i|, \tag{1.4}$$

with probabilities p_i, $\sum p_i = 1$. The set of mixed states is convex, i.e., for all mixed states ϱ and σ and $1 \leq \lambda \leq 0$ all convex combinations $\lambda\varrho + (1-\lambda)\sigma$ are also mixed states. Given any basis $\{|i\rangle\}$ in $\mathcal{H}$, ϱ can be represented as

$$\varrho = \sum_{ij} M_{ij} |i\rangle\langle j|, \tag{1.5}$$

[3] These quantum operations may involve such operations, which have non-unit success probability.

with positive semidefinite Hermitian[4] matrix M_{ij} with $\mathrm{tr}M = 1$. Conversely, any positive semidefinite Hermitian matrix of trace one corresponds to a mixed state.

Given any mixed state ϱ_{AB} on the joint system $\mathcal{H}_{AB}$, it is separable if it can be written as statistical ensemble of product states of the local subsystems

$$\varrho_{AB}^{\mathrm{sep}} = \sum_i p_i \varrho_A^{(i)} \otimes \varrho_B^{(i)}. \tag{1.6}$$

Note that the set of all separable states $\mathcal{S}$ is convex, compact and invariant under change of local basis (unitary product operations $U_A \otimes U_B$). As for the pure states if such a decomposition does not exist the state is said to be bipartite entangled.

Mixed states can be interpreted as a statistical ensemble, which is known from classical statistical mechanics. In any physical scenario, we end up in one of three situations. We either have a product state $\varrho = \varrho_A \otimes \varrho_B$. In this scenario, Alice and Bob have two independent sources and each one prepares a mixed quantum state ϱ_A and ϱ_B, respectively. This procedure involves local operations on either of the subsystems only. Another possibility is to have the more general separable state as given by Eq. (1.6). Here Alice and Bob would make use of a joint random number generator to prepare their joint state, which results in a statistical operation together with classical communication. Upon receiving the random number i, which is sent with probability p_i, Alice and Bob would locally prepare the mixed states $\varrho_A^{(i)}$ and $\varrho_B^{(i)}$, respectively. This procedure is still not employing any true quantum mechanical features and thus the statistics of any succeeding experiment could also be generated in a suitable classical experiment. In the remaining scenario, however, the mixed state is entangled. Here it is not possible to devise any procedure containing just statistical operations and classical communication (SLOCC) to create the state.

The main problem of the theory of entanglement is called the separability problem. That is, for any given state to check if it is separable or entangled. Although this problem is easy to solve for pure states by using the Schmidt rank criterion, for mixed states this problem is extremely complex and there is a wide range of criteria and mathematical tools available to detect entangled states[5]. This thesis will concentrate on a criterion derived from the positive partial transpose (PPT) criterion, which partially describes the set of mixed separable states.

1.1.4 Positive partial transpose

One of the strongest criteria for separability has been proposed by Peres in Ref. [59]. It is build on the idea that the transposition operation maps the set of quantum states onto itself. Given any state as defined in Eq. (1.5) then its transpose ϱ^T is given by

$$\varrho^T = \sum_{ij} M_{ij} |j\rangle\langle i|. \tag{1.7}$$

Recall that for any given pair of bases $\{|i\rangle\}$ and $\{|k\rangle\}$ of the respective subsystems $\mathcal{H}_A$ and $\mathcal{H}_B$ any bipartite state ϱ_{AB} on the joint system $\mathcal{H}_{AB}$ can be expressed as

$$\varrho = \sum_{ij,kl} \varrho_{ij,kl} |i\rangle\langle j| \otimes |k\rangle\langle l|. \tag{1.8}$$

Given such a state the partial transposition of subsystem A is defined by transposition of the first subsystem

$$\varrho^{T_A} = \sum_{ij,kl} \varrho_{ij,kl} |j\rangle\langle i| \otimes |k\rangle\langle l|. \tag{1.9}$$

The transposition of the second subsystem can be defined analogously. The positive partial transpose (PPT) criterion then states that if ϱ_{AB} is a separable state, then $\varrho_{AB}^{T_A}$ is a quantum state. If on the other hand $\varrho_{AB}^{T_A}$ has negative eigenvalues and hence is no quantum state, ϱ_{AB} is entangled.

In case the dimension of the joint Hilbert space $\mathcal{H}_{AB}$ does not exceed six the PPT criterion is necessary and sufficient [64], i.e., a state is separable if and only if it has positive partial transpose. For higher total dimension there exist entangled states with positive partial transpose and hence the set of PPT $\mathcal{S}_{PPT}$ states is a superset of the set of separable states $\mathcal{S}$.

[4] A finite dimensional matrix M is said to be Hermitian if it coincides with its conjugate transpose $M = M^\dagger$. Further, M is said to be positive semidefinite if for all vectors x, $x^\dagger M x \geq 0$.

[5] At present there is no computable tool available to solve the separability problem for arbitrary mixed quantum states

1.1.5 Entanglement witnesses

Another way to characterise the set of separable states is to use a geometric approach [64, 65]. Since every convex set can be described by hyperplanes, a state $\varrho_{AB} \in \mathcal{H}_{AB}$ is separable if it has non-negative expectation value for all entanglement witnesses W

$$\mathrm{tr}\,(W\varrho_{AB}) \geq 0. \tag{1.10}$$

These are observables W that have at least one negative eigenvalue and satisfy

$$\langle\psi_A\phi_B|W|\psi_A\phi_B\rangle \geq 0 \tag{1.11}$$

for all product states $|\psi_A\phi_B\rangle$. Conversely, a state $\varrho_{AB} \in \mathcal{H}_{AB}$ is entangled if there exists an entanglement witness W with negative expectation value with respect to that state $\mathrm{tr}(W\varrho_{AB}) < 0$.

Note that the framework of witnesses can be also used to describe the set of PPT states $\mathcal{S}_{PPT}$. In Ref. [66], the authors introduce the set of decomposable witnesses

$$W^{dec} = P + Q^{T_A}, \tag{1.12}$$

with positive operators $P, Q \geq 0$. It has been shown that these witnesses have non-negative expectation value on $\mathcal{S}_{PPT}$ and fully characterise the set of PPT states. For this characterisation it is even sufficient to consider decomposable witnesses, where $P = 0$.

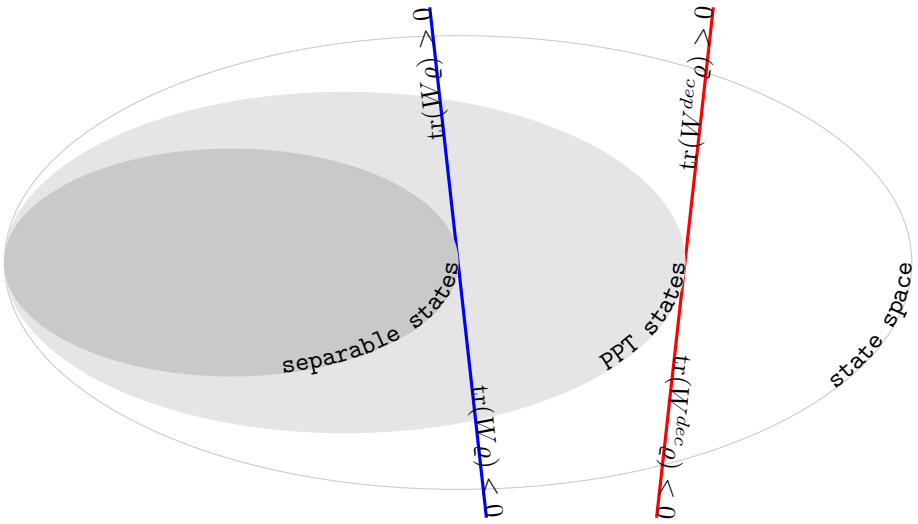


Figure 1.1: Shown are the convex set of all states, the convex set of PPT states and its convex subset: the separable states. States outside the set of separable states are entangled. The red line shows the hyperplane, where the expectation value of a decomposable witness W^{dec} vanishes. All states to the right of line are detected to have **no** positive partial transpose. Similarly the blue line shows, where the expectation value of an entanglement witness W vanishes and all states to the right of this line are detected to be entangled.

1.1.6 Negativity

Using the PPT criterion Życzkowsi et al. [50] first introduced the negativity, which was later shown to be a measure of entanglement by Vidal and Werner [51]. Given any state ϱ_{AB} on a bipartite quantum system, the negativity is defined as the sum $\sum_i |\lambda_i^-(\varrho_{AB}^{T_A})|$ of the negative eigenvalues λ_i^- of the partially transposed state $\varrho_{AB}^{T_A}$

$$N(\varrho_{AB}) = \sum_i |\lambda_i^-(\varrho_{AB}^{T_A})|. \tag{1.13}$$

Hence the negativity measures the violation of the PPT criterion.

In Ref. [51], it was further shown that the negativity can be cast into an optimisation problem, which involves to write the partial transpose as a difference of two positive operators. That is

$$\begin{aligned} N(\varrho_{AB}) = \inf & a_- \\ \text{subjected to: } & \varrho_{AB}^{T_A} = a_+\varrho^+ - a_-\varrho^-, \text{ where} \\ & a_\pm \geq 0 \text{ and} \\ & \varrho^\pm \text{ are density matrices.} \end{aligned} \tag{1.14}$$

We finish this section by recalling the main properties of the GMN as proved in Ref. [50, 51]:

Lemma 1.1. *The negativity N as defined by Eq. (1.13) satisfies:*

1. *N vanishes on all separable states $\varrho_{AB}^{\text{sep}}$, i.e., $N(\varrho_{AB}^{\text{sep}}) = 0$. Further, if ϱ_{AB} has no PPT, then $N_g(\varrho_{AB}) > 0$.*
2. *N is non-increasing under full LOCC operations (no joint operations on more than one part are allowed), i.e., $N(\Lambda_{\text{LOCC}}(\varrho_{AB})) \leq N(\varrho_{AB})$.*
3. *N is invariant under local basis changes U_{loc}, i.e., $N(U_{\text{loc}}\varrho_{AB}U_{\text{loc}^\dagger}) = N(\varrho_{AB})$.*
4. *N is convex, i.e., $N(\varrho_{AB}) \leq \sum_i p_i N(\varrho_{AB}^{(i)})$ holds for all convex decompositions $\varrho_{AB} = \sum_i p_i \varrho_{AB}^{(i)}$.*
5. *N is bounded by $N(\varrho_{AB}) \leq \frac{1}{2}(d_{\min} - 1)$, where $d_{\min} = \min(\dim(\mathcal{H}_A), \dim(\mathcal{H}_B))$ is the minimal dimension of either of the systems.*

1.2 Genuine multiparticle entanglement

So far, only bipartite quantum systems composed of two quantum systems have been considered. In many experimental situations, however, one wants to use genuine multiparticle systems, which consist of more than two parts.

In this case, the notions of entanglement and separability become much richer. Additionally to the so-called full separability and genuine multiparticle entanglement, which can be seen as counter parts to the bipartite separability and entanglement, there are many types of partial separability and complementary types of entanglement occurring.

1.2.1 The three-particle case

The simplest scenario, where genuine multiparticle entanglement can occur, is the case of three parties Alice (A), Bob (B) and Charlie (C) with Hilbert space $\mathcal{H}_{ABC} = \mathcal{H}_A \otimes \mathcal{H}_B \otimes \mathcal{H}_C$. In contrast to the bipartite case, pure states will not be discussed separately.

A mixed three-particle state is fully separable and contains no entanglement if it can be written as a statistical mixture of three particle product states

$$\varrho^{\text{fulsep}} = \sum_k p_k |\psi_A^k\rangle\langle\psi_A^k| \otimes |\phi_B^k\rangle\langle\phi_B^k| \otimes |\phi_C^k\rangle\langle\phi_C^k|. \tag{1.15}$$

Apart from full separability it might also happen that particles A and B are entangled with each other, whereas the particle C is not. Such a state is said to be partial separable with respect to the partition C versus AB ($C|AB$) and can be written as a statistical mixture of product states on the system AB and the system C

$$\varrho_{C|AB}^{\text{sep}} = \sum_k p_k |\psi_{AB}^k\rangle\langle\psi_{AB}^k| \otimes |\phi_C^k\rangle\langle\phi_C^k|. \tag{1.16}$$

These states are a straightforward generalisation of the bipartite separable states given by Eq. (1.6), where one considers the systems AB as a single system without internal structure. This notion then generalises naturally to states, which are separable with respect to other possible splittings $B|AC$ and $A|BC$.

Taking the convex hull of all states, which are separable with respect to some partition $A|BC$, $B|AC$, or $A|BC$ yields the set of biseparable states. Each such state can be written as

$$\varrho^{\text{bisep}} = p_A \varrho^{\text{sep}}_{A|BC} + p_B \varrho^{\text{sep}}_{B|AC} + p_C \varrho^{\text{sep}}_{C|AC}. \tag{1.17}$$

States outside this set are called genuine multiparticle entangled. They cannot be written in the above form.

In summary, the different sets of separable states create a nested structure all of, which are contained within the biseparable states (see Fig. 1.2).

1.2.2 Partial separability

The nested structure of different kinds of separable states generalises naturally to the case of an n-particle system $\mathcal{H} = \bigotimes_{k=1}^{n} \mathcal{H}_{A_1}$. Let $I = \{1, 2, \ldots, n\}$ be the set of all indices, then a state ϱ is separable with respect to a k-partition[6] $\{I_1, I_2, \ldots, I_k\}$ if and only if it can be written as statistical mixture

$$\varrho_{k-part} = \sum_i p_i \varrho_1^{(i)} \otimes \varrho_2^{(i)} \otimes \cdots \otimes \varrho_k^{(i)}, \tag{1.18}$$

where the I_l are disjoint subsets of I with $\bigcup_l I_l = I$ and $\varrho_l^{(i)}$ is defined as a state on the subsystems $\bigotimes_{m \in I_l} \mathcal{H}_{A_m}$. As an example consider the state $\varrho^{\text{sep}}_{C|AB}$ as given by Eq. (1.16), with $I = \{A, B, C\}$, $I_1 = \{C\}$ and $I_2 = \{AB\}$.

Similar to the biseparable states one defines k-separable states as the convex hull of all states, which are separable with respect to some $k - partition$. Such a state is then given by

$$\varrho_{k-sep} = \sum_i p_i \varrho^{(i)}_{k-part}, \tag{1.19}$$

where the states $\varrho^{(i)}_{k-part}$ are in general separable with respect to different k partitions. Note that this notion also includes the fully separable states and the biseparable states, which we already introduced in the last section. A n-separable state is also said to be fully separable and a 2-separable state is said to be biseparable. As in the three-particle case a state is genuine multiparticle entangled if and only if it is not biseparable.

Note that the separability problem introduced in the bipartite case generalises quite naturally to the multiparticle case. That is for a given state to check if it is k-separable for any k. Henceforth, we will leave aside the different kinds of k-separability and concentrate on the detection and quantification of genuine multiparticle entanglement.

1.2.3 Three-qubit SLOCC-equivalence

The simplest case, where genuine multiparticle entangled states might occur is the case of three-qubits $\mathcal{H} = \mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2$. Recall the notion of SLOCC-equivalence as introduced in Paragraph 1.1.2. One may ask if for example all three-qubit genuine multiparticle entangled states are SLOCC-equivalent.

The answer to this question is no: For three qubits there are two inequivalent classes of genuine multiparticle entangled states [67]. The class of GHZ (Greenberger-Horne-Zeilinger) states represented by

$$|GHZ\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle) \tag{1.20}$$

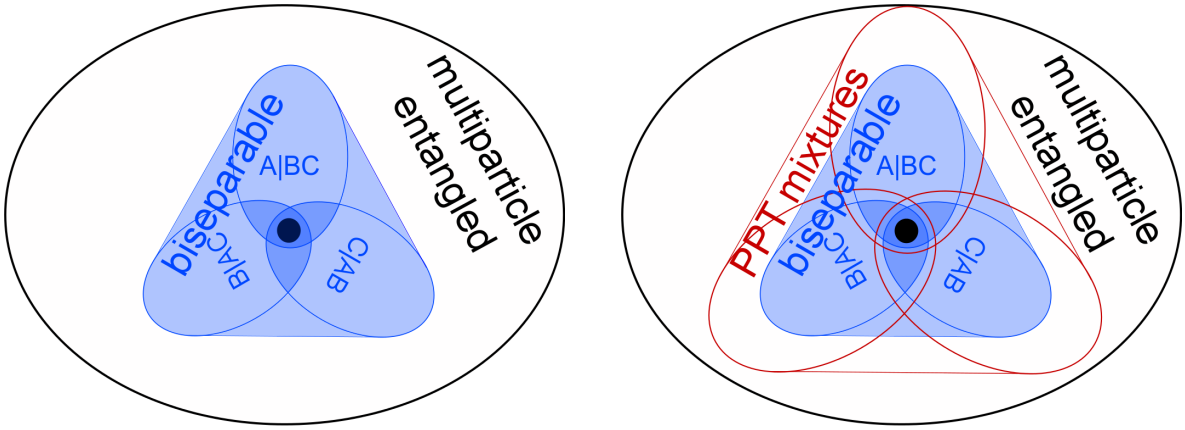
and the class of W states represented by

$$|W\rangle = \frac{1}{\sqrt{3}}(|001\rangle + |010\rangle + |100\rangle). \tag{1.21}$$

GHZ-states can be transformed into all other states using SLOCC transformations but it is impossible to transform a W-state into a GHZ-state.

Although at first sight this classification seems to be a mere mathematical distinction both states have distinct entanglement properties with respect to particle loss. That is if one party sharing a W-state losses its particle then the remaining two parties still share a bipartite entangled state. If, however, the shared system was in a GHZ-state then after particle loss the remaining parties share a separable state.

[6] If $k = 2$ one often denotes the 2-partition as bipartition.



(a) (b)

Figure 1.2: (a) The nested structure of the different convex sets of separable states within the set of all states. The innermost set (dark dot) represents the fully separable states. The three sets of states, which are separable (with respect to a partition $A|BC$, $B|AC$, or $C|AB$) are supersets of the set of fully separable states, which is contained in but not equal to the intersection of the three. The biseparable states form the convex hull of the separable states with respect to the different partitions. All states outside the set of biseparable states are genuine multiparticle entangled. (b) The PPT mixtures are the convex hull of states, which are PPT with respect to a single partition. As all states, which are separable with respect to a certain partition are also PPT, the PPT mixtures form a superset of the biseparable states. States, which are not PPT mixtures are genuine multiparticle entangled.

1.2.4 PPT mixtures and witnesses

At present, there is no general framework to prove or disprove the existence of a biseparable decomposition for arbitrary mixed states. In Ref. [15], this problem was studied by introducing a relaxation. Instead of trying to characterise the set of biseparable states as given in Eqs. (1.17) and (1.19)[7] one characterises the superset of so-called PPT mixtures.

The idea builds on the fact that the separable states are a subset of the states with a positive partial transpose (PPT states) [59]. Recall the notion of bipartite PPT states from Section 1.1.4.

In the three-particle case [see Fig. 1.2(b)] PPT mixtures are states, which can be written as

$$\varrho^{\mathrm{PPTmix}} = p_A \varrho^{\mathrm{PPT}}_{A|BC} + p_B \varrho^{\mathrm{PPT}}_{B|AC} + p_C \varrho^{\mathrm{PPT}}_{C|AB}, \tag{1.22}$$

where $\varrho^{PPT}_{A|BC}$, $\varrho^{PPT}_{B|AC}$ and $\varrho^{PPT}_{C|AB}$ is a state, which has positive partial transpose with respect to the transposition of subsystem A, B and C, respectively. A general PPT mixture on more than three parties will have $2^{n-1}-1$ terms, one for each partition $m|\bar{m}$ of the system into two parts. Such a bipartition is a splitting of the system into a part m and its complement $\bar{m}$. Note, however, that $m|\bar{m}$ and $\bar{m}|m$ label the same bipartition.

The main advantage of this approach is that for any given multiparticle state ϱ one can directly check whether it is a PPT mixture or not by using the concept of witnesses. In Ref. [15], it was shown that the non-existence of such a decomposition is equivalent to the existence of a fully decomposable witness $\mathcal{W}$ detecting the state $tr(\mathcal{W}\varrho) < 0$. Such a witness is an operator $\mathcal{W}$, which can be written for all possible bipartitions $m|\bar{m}$ as $\mathcal{W} = P_m + Q_m^{T_m}$, with positive operators P_m and Q_m. That way, it is a direct generalisation of a decomposable witness as defined in Eq. (1.12) taking into account that there is more than one possible bipartition of a multiparticle system. One of the interesting features of this approach is that finding such a witness can be cast into a so-called semidefinite program (SDP, see also below), which can be solved efficiently with standard numerical routines [68, 69].

[7] Recall that biseparable states correspond to the case, where $k = 2$.

1.3 Entanglement monotones

In the last two sections, the definitions of bipartite and multiparticle entanglement were reviewed and some methods of entanglement detection were introduced. In this section, we will go a step further and review the axiomatic approach of entanglement quantification that some of the most prominent entanglement monotones are based upon. Finally, we will review the genuine multiparticle negativity, which is a genuine multiparticle entanglement monotone based on fully decomposable witnesses introduced in Ref. [15].

1.3.1 Axioms

At first, many entanglement measures such as distillable entanglement or entanglement cost connect entanglement with its usefulness in specific quantum communication tasks [70, 71]. In the axiomatic approach the entanglement quantifying functions only needs to satisfy two axioms and might be completely unrelated to any quantum task.

Let E be a real valued function on the state space, then E is an entanglement monotone iff the following two axioms are satisfied [72]:

1. E is monotonous under LOCC transformations. That is for any local operations and classical communication protocols Λ, $E(\Lambda(\varrho)) \leq E(\varrho)$. Such a LOCC protocol is a special instance of a SLOCC protocoll discussed above in Paragraph 1.1.2, where we demand the success probability of the protocol to be one. Note contrary to SLOCC transformations, LOCC transformations cannot be as easily characterised by invertible local operations.

2. From the monotonicity under LOCC transformations, it follows that E is constant on the set of separable states [72] and one usually demands the monotone to vanish on this set.

Remark 1.1. Note that both axioms also impose E being non-negative [72]. Furthermore, there are other properties, which might be satisfied by some monotones, although these are not required by the definition of a monotone.

As an example the negativity [50, 51]:

- is convex, i.e., $E(\varrho) \leq \sum_i p_i E(\sigma_i)$ for all ensemble decompositions of ϱ into $\varrho = \sum_i p_i \sigma_i$.
- is invariant under local basis change U_{loc}, i.e., $E(U_{\mathrm{loc}} \varrho U_{\mathrm{loc}^\dagger}) = E(\varrho)$.

Its logarithm, called logarithmic negativity [51] is no longer convex but is:

- normalised in an information theoretic way. That is $E(\varrho^{\otimes n}) = nE(\varrho)$.

1.3.2 Bipartite convex roof measures

A convenient method to obtain entanglement monotones in a bipartite setting, where the Hilbert space consists of two parts $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B$, is to use a measure defined on pure states and extend it to mixed ones via a pure state convex roof [73].

Let E be a function defined on the pure states $E : \mathcal{H} \to \mathbb{R}$, which is monotonous under LOCC and vanishes on the set of separable pure states. Then its pure state convex roof given by

$$E(\varrho) = \inf \sum_i p_i E(|\psi_i\rangle), \quad \sum_i p_i = 1, \quad p_i \geq 0, \tag{1.23}$$

where the infimum is taken over all possible decompositions $\varrho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$, is an entanglement monotone [72].

Moreover, if E is a necessary and sufficient entanglement criterion for pure states, then its convex roof extension also is. On the downside one has to add that it is in general not possible to numerically calculate the minimisation given in Eq. (1.23) and hence one is often restricted to use lower bounds to estimate these monotones.

Example 1.1. To mention but two prominent examples, where the convex roof construction is used to obtain entanglement monotones consider the following examples:

- Using the von Neumann entropy $S(\varrho) = -\mathrm{tr}\varrho \log_2 \varrho$ one defines the pure state measure entropy of entanglement. It is given by by $E(|\psi\rangle) = S(\varrho_A) = S(\varrho_B)$, where ϱ_A and ϱ_B are the reduced density matrices with subsystem A and B, respectively, traced out. Given this pure state measure, its convex roof extension E_F is an entanglement monotone called entanglement of formation [71].

  Note that, although we did not require any quantum informational interpretation, this monotone upper bounds the distillable entanglement [71].

- Another measure, whose convex roof extension is frequently used is the concurrence

$$C(|\psi\rangle) = \sqrt{2(1 - \mathrm{tr}\varrho^2)}, \tag{1.24}$$

  where ϱ is either of the reduced states ϱ_A or ϱ_B.

  Quite remarkably, the convex roof extension of the concurrence can be expressed in a closed form in the case a two-qubit system is considered. In that case the concurrence is given by

$$C(\varrho) = \max(0, \lambda_1 - \lambda_2 - \lambda_3 - \lambda_4), \tag{1.25}$$

  where λ_i are the eigenvalues of $\sqrt{\sqrt{\varrho}\tilde{\varrho}\sqrt{\varrho}}$ with $\tilde{\varrho} = (\sigma_y \otimes \sigma_y)\varrho^*(\sigma_y \otimes \sigma_y)$[8] in decreasing order [74].

  Moreover, in Ref. [74] it has been shown that the concurrence is closely related to the entanglement of formation. That is

$$E_F(\varrho) = H\left(\frac{1 + \sqrt{1 - C^2(\varrho)}}{2}\right), \tag{1.26}$$

  where $H(x) = -x \log_2 x$.

1.3.3 Genuine multiparticle convex roof measures

Now let us consider a multiparticle system. In the simplest case, it consists of three parts A, B and C. A generalisation to more parts is straightforward. Given this setting, there is natural way to extend the convex roof construction introduced in the last paragraph to obtain genuine multiparticle entanglement monotones from bipartite pure state entanglement measures.

In analogy to the bipartite case, a genuine multiparticle entanglement monotone is a function mapping mixed states to the real numbers, such that it is monotonous under LOCC transformations and vanishes for all biseparable states.

Let $\{E_k\}$, $k \in \{A, B, C\}$, be a family of pure state entanglement monotones defined on $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B \otimes \mathcal{H}_C$, such that E_A, E_B and E_C vanish on states, which are separable with respect to the splitting $A|BC$, $B|AC$ and $C|BA$, respectively. Then define

$$\mu(|\psi\rangle) := \min\left\{E_A(|\psi\rangle), E_B(|\psi\rangle), E_C(|\psi\rangle)\right\}. \tag{1.27}$$

Then, μ is a pure state entanglement measure, which vanishes on all states that are separable with respect to some bipartition. If $\mu(|\psi\rangle) > 0$, then $|\psi\rangle$ is genuine multiparticle entangled.

One can then extend μ to mixed states using the pure state convex roof construction as given by Eq. (1.23)

$$E(\varrho) = \inf \sum_i p_i \mu(|\psi_i\rangle) \tag{1.28}$$

to obtain a genuine multiparticle entanglement monotone. Then $E(\varrho) > 0$ is sufficient for ϱ to be genuine multiparticle entangled.

This general method was used in Ref. [75] to extend the bipartite concurrence as given by Eq. (1.24) to a genuine multiparticle entanglement monotone. This monotone is denoted by GME-concurrence and is necessary and sufficient to detect genuine multiparticle entanglement in mixed quantum states. As already discussed it is in general not possible to calculate the expression given in Eq. (1.28). If analytic formulas are known, then they are usually restricted to highly symmetric state families as for the GME-concurrence [76].

[8] Here ϱ^* denotes the entry-wise complex conjugate of ϱ.

Remark 1.2. So far, we considered the case where a family of bipartite measures on pure states was given. If on the other hand bipartite monotones are given, which are already defined on mixed states, then one can also perform a mixed convex roof construction [52–55] to construct a genuine multiparticle entanglement monotone. Moreover, in certain instances the monotones defined that way are a convex optimisation problem known as semidefinite program[9], which can be efficiently numerically solved.

Let $\{E_k\}$, $k \in \{A, B, C\}$, be a family of mixed state bipartite entanglement monotones defined on $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B \otimes \mathcal{H}_C$ with the same properties, as in the pure state case. Given that, one can define a genuine multiparticle entanglement monotone by

$$E(\varrho) = \inf p_A E_A(\varrho_A) + p_B E_B(\varrho_B) + p_C E_C(\varrho_C), \tag{1.29}$$

where the optimisation is performed over all mixed state decompositions $\varrho = p_A\varrho_A + p_B\varrho_B + p_C\varrho_C$. Later we will show that a renormalised version of the genuine multiparticle negativity originates from such a mixed state construction.

1.3.4 Genuine multiparticle negativity (GMN)

We already considered the construction of genuine multiparticle entanglement monotones using bipartite ones. One might on the other hand also approach the quantification of genuine multiparticle entanglement from a different direction.

Recalling Section 1.2 we showed that fully decomposable witnesses provide sufficient criteria for a state to be genuine multiparticle entangled. Building upon this idea, Ref. [15] introduced a computable entanglement monotone called the genuine multiparticle negativity (GMN). The basic idea is to take a fully decomposable witness as above and use the violation of it as a quantifier of entanglement. More precisely, one defines the GMN $\tilde{N}_g(\varrho)$ via the optimisation problem:

$$\begin{aligned} \tilde{N}_g(\varrho) = -\min &\operatorname{tr}(\varrho\mathcal{W}) \\ \text{subjected to: } &\mathcal{W} = P_m + Q_m^{T_m}, \\ &0 \leq P_m \leq \mathbb{1}, \\ &0 \leq Q_m \leq \mathbb{1} \quad \text{for all partitions } m|\bar{m}. \end{aligned} \tag{1.30}$$

That is, for the three-particle case the witness operator has to be decomposable into $\mathcal{W} = P_A + Q_A^{T_A}$, $\mathcal{W} = P_B + Q_B^{T_B}$ and $\mathcal{W} = P_C + Q_C^{T_C}$ with $0 \leq P_m, Q_m \leq \mathbb{1}$. Since this measure is defined as an optimisation over a set of witnesses, it is closely related to the approach to quantify entanglement based on entanglement witnesses as in Ref. [77]. Furthermore, it can be directly computed using SDP [69], since the optimisation problem in Eq. (1.30) is an optimisation problem of this class. We finish this section by recalling the main properties of the GMN as proved in Ref. [15, 24]:

Lemma 1.2. *The measure $\tilde{N}_g$ as defined by Eq. (1.30) satisfies:*

i $\tilde{N}_g$ *vanishes on all biseparable states ϱ^{bisep}, i.e., $\tilde{N}_g(\varrho^{\text{bisep}}) = 0$. Further if ϱ is no PPT mixture, then $\tilde{N}_g(\varrho) > 0$.*

ii $\tilde{N}_g$ *is non-increasing under full LOCC operations (no joint operations on more than one part are allowed), i.e., $\tilde{N}_g(\Lambda_{\text{LOCC}}(\varrho)) \leq \tilde{N}_g(\varrho)$.*

iii $\tilde{N}_g$ *is invariant under local basis changes U_{loc}, i.e., $\tilde{N}_g(U_{\text{loc}}\varrho U_{\text{loc}^\dagger}) = \tilde{N}_g(\varrho)$.*

iv $\tilde{N}_g$ *is convex, i.e., $\tilde{N}_g(\varrho) \leq \sum_i p_i \tilde{N}_g(\varrho_i)$ holds for all convex decompositions $\varrho = \sum_i p_i \varrho_i$.*

v $\tilde{N}_g$ *is bounded by $\tilde{N}_g(\varrho) \leq \frac{1}{2}(d_{\min} - 1)$, where $d_{\min}$ is the lowest dimension of any subsystem [24].*

vi If the system consists of two parties only, then $\tilde{N}_g$ equals the bipartite negativity [50, 51].

[9] Note that semidefinite programming will be discussed later in this thesis.

1.4 Stabiliser formalism

The stabiliser formalism was originally developed for quantum error correction [21, 22] and yields a special set of many-qubit states called stabiliser states [20, 21]. These states are defined as common eigenstates of a set of commuting local operators and contain the so-called graph states, which are a particularly useful class of states [61], as subset.

In this section, we review the basic notions of stabilisers and stabiliser states and their connection to graph states. For a detailed review on the subject we refer the reader to Refs. [20, 61].

1.4.1 Stabiliser states and graph states

To begin with define the Pauli group G_n on n qubits as the set of elements consisting of arbitrary tensor products of the operators $\{\mathbb{1}, \sigma_x, \sigma_y, \sigma_z\}$ with either of the phases $\{1, -1, i, -i\}$. Using $\sigma_0 \equiv \mathbb{1}$, $\sigma_1 \equiv \sigma_x$, $\sigma_2 \equiv \sigma_y$ and $\sigma_3 = \sigma_z$ the Pauli group is given by

$$G_n = \{\alpha\sigma_{j_1} \otimes \sigma_{j_2} \otimes \cdots \otimes \sigma_{j_n} | j_k \in \{0, 1, 2, 3\}, \alpha \in \{1, -1, i, -i\}\}\,. \tag{1.31}$$

The n qubit Pauli group has 4^{n+1} elements, which either commute or anti-commute. Furthermore, each element of the Pauli group is a Hermitian or anti-Hermitian operator on the n-qubit Hilbert space $\mathcal{H}_n = \bigotimes_{i=1}^{n} \mathbb{C}^2$.

In the stabiliser formalism one does not need the full Pauli group, but considers maximal Abelian subgroups[10] $S \subseteq G_n$ with $-\mathbb{1} \notin S$ only. These subgroups contain only Hermitian operators with eigenvalues $+1$ and/or -1 and are also referred to as stabiliser groups.

A quite convenient method to describe stabiliser groups is in terms of algebraic independent[11] generators $g_1, \ldots, g_n \in S$. Using the fact that $s^2 = \mathbb{1}$ for all $s \in S$ and the commutativity of S, one can uniquely express each $t \in S$ as

$$t = g_1^{x_1} g_2^{x_2} \ldots g_k^{x_k}, \qquad x_l \in \{0, 1\}\,. \tag{1.32}$$

Here the order of the generators in the product in Eq. (1.32) is irrelevant due to the commutativity of S.

One concludes that $|S| = 2^n$ and that S is completely determined by its generators. Hence we will often identify S with the set of its generators $\{g_1, g_2, \ldots, g_n\}$. Note, however, that there are many possible equivalent generator sets, which generate the same group.

Now let S be a stabiliser group and $\{g_1, g_2, \ldots, g_n\}$ be a fixed set of generators. Then one can define projectors onto the positive and negative subspaces of each of the g_j by

$$\Pi_j^{\pm} = \frac{1}{2}(\mathbb{1} \pm g_j). \tag{1.33}$$

For each S there exists an one-dimensional subspace V_S, which is the common eigenspace to eigenvalue $+1$ for all $s \in S$. The projector onto the common positive eigenspace $V_{++\cdots+}$ is then given by the product of the projectors onto the individual positive eigenspaces

$$\Pi^{++\cdots+} = \frac{1}{2^n} \prod_{j=1}^{n} (\mathbb{1} + g_j). \tag{1.34}$$

Similarly one can define projectors onto the other eigenspaces, which can be labelled by the n signs as done in Eq. (1.34) by k times $+$. One defines the stabiliser state to be the up to a global phase uniquely determined eigenvector $|\psi_S\rangle$ to eigenvalue $+1$ for all g_j

$$g_j |\psi_S\rangle = |\psi_S\rangle. \tag{1.35}$$

A special class of stabilising groups is given by the so-called graph states, which are relevant in many applications in quantum information processing [61].

As the name indicates, the corresponding maximal stabiliser groups are closely connected to mathematical graphs in their definition. Therefore let us consider a graph G, which is a set of n vertices with edges

[10] A subgroup is said to be Abelian, iff all of its elements commute with each other. Furthermore, an Abelian subgroup $S \subseteq G_n$ is said to be maximal if there is no other Abelian subgroup $\tilde{S} \subseteq G_n$, such that $S \subseteq \tilde{S}$.

[11] I.e., no generator can be written as a product of the remaining.

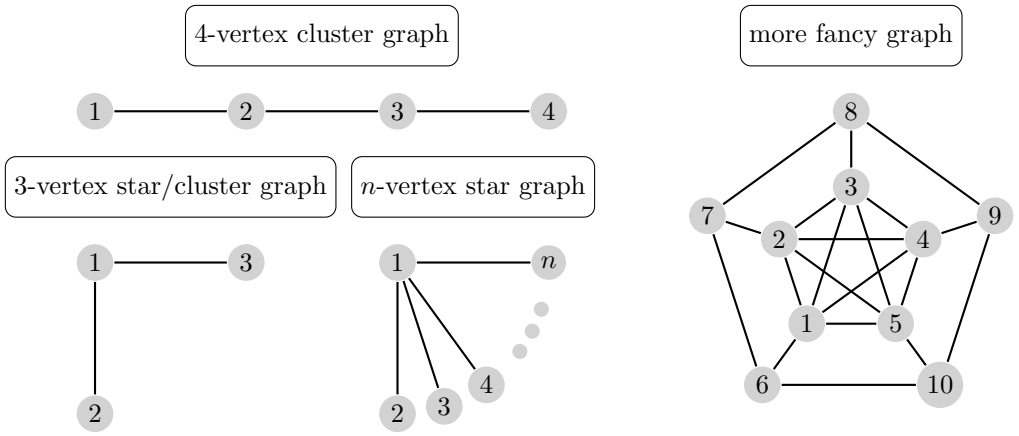


Figure 1.3: In this figure the 4-vertex cluster graph, the 3-vertex star graph, the general n-vertex star graph and a more fancy graph with labelled vertices are shown.

connecting them (see Fig. 1.3). For each vertex i we define its neighbourhood $\mathfrak{N}(i)$ by the set of all vertices, which are connected to i.

The graph state formalism then assigns a stabilising operator g_i to each vertex of the graph. Denote by $X^{(i)}$, $Y^{(i)}$ and $Z^{(i)}$ the Pauli matrices σ_x, σ_y and σ_z on the i-th particle with the identity on all the other, then

$$g_i := X^{(i)} \bigotimes_{j\in\mathfrak{N}(i)} Z^{(j)}. \tag{1.36}$$

These operators commute pairwise and define the n independent generators $\{g_1, g_2, \ldots, g_n\}$ of the stabilising group S corresponding to the graph G. The graph state $|G\rangle$ is then the unique n-qubit eigenstate to eigenvalue $+1$ to all g_i.

$$g_i|G\rangle = |G\rangle, \text{for all } i = 1, 2, \ldots, n. \tag{1.37}$$

Example 1.2. To illustrate the stabiliser and graph state formalism consider the following two examples:

1. First, consider the stabiliser group generated by $g_1 = XXX$, $g_2 = ZZ\mathbb{1}$ and $g_3 = Z\mathbb{1}Z$[12] then the stabiliser group generated by g_i is given by

$$G = \{\mathbb{1}\mathbb{1}\mathbb{1}, XXX, ZZ\mathbb{1}, Z\mathbb{1}Z, YYX, YXY, \mathbb{1}ZZ, XYY\}\,. \tag{1.38}$$

   The corresponding stabiliser state is the well known GHZ-state [12]. In the computational basis it reads

$$|GHZ\rangle = \frac{1}{\sqrt{2}}\left(|000\rangle + |111\rangle\right). \tag{1.39}$$

2. Second, consider the stabiliser group corresponding to the star graph with with three vertices (see Fig. 1.3). Its generators are given by $g_1 = XZZ$, $g_2 = ZX\mathbb{1}$ and $g_3 = Z\mathbb{1}X$ according to Eq. (1.36). The generated group is given by

$$G = \{\mathbb{1}\mathbb{1}\mathbb{1}, XZZ, ZX\mathbb{1}, Z\mathbb{1}X, YYZ, YZY, \mathbb{1}XX, XYY\}\,. \tag{1.40}$$

   Then the corresponding graph state can be expressed in the computational basis as

$$|G\rangle = \frac{1}{\sqrt{8}}\left(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle - |101\rangle - |110\rangle + |111\rangle\right). \tag{1.41}$$

[12]These operators do not correspond to a graph, but are equivalent to $g_1 = XZZ$, $g_2 = ZX\mathbb{1}$ and $g_3 = Z\mathbb{1}X$ (these arise from the three vertex linear graph) by a local basis change. By a slight abuse of notation we omit the tensor product signs to improve readability ($XXX \equiv X \otimes X \otimes X$).

1.4.2 Local Clifford equivalence

It was already pointed out that each stabiliser state $|\psi_S\rangle$ uniquely corresponds to a maximal Abelian subgroup $S \subset G_n$ of the Pauli group. However, it might still occur that two stabiliser states are equal up to some local unitary operation, which corresponds to a simple basis change, i.e., given two stabiliser subsets $S, S' \subset G_n$ there exists a $U \in U(2)^{\otimes n}$, such that

$$U|\psi_S\rangle = |\psi_{S'}\rangle. \tag{1.42}$$

Note, however, that even though the stabiliser states are LU equivalent the corresponding stabiliser groups are in general not

$$USU^\dagger \neq S'., \tag{1.43}$$

since in general $USU^\dagger$ is no longer a subgroup of G_n [29][13].

Hence to study the equivalence of stabiliser states within the stabiliser formalism one usually considers the local Clifford (LC) group [20], which is the set of unitaries that map G_n onto itself

$$C_1^{\otimes n} = \left\{ U \in U(2)^{\otimes n} | UG_nU^\dagger = G_n \right\}. \tag{1.44}$$

It is given by the n-fold tensor product of the single-qubit Clifford group, which is generated by the Hadamard gate H and the single-qubit phase gate Φ

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad \text{and} \quad \Phi = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}. \tag{1.45}$$

Similar to local unitary equivalence we denote two stabiliser states LC-equivalent if there is an element of the local Clifford group mapping the states onto each other. I.e., given two stabiliser subsets $S, S' \subset G_n$ there exists a $U \in C_1^{\otimes n}$, such that

$$U|\psi_S\rangle = |\psi_{S'}\rangle. \tag{1.46}$$

In this case, the corresponding stabiliser groups are also equivalent

$$USU^\dagger = S'. \tag{1.47}$$

In fact, it has been shown that any general stabiliser state is LC-equivalent to some graph state [78, 79]. Hence it often suffices to consider graph states only.

Example 1.3. To illustrate the result recall the example of the last paragraph with the GHZ-state

$$|GHZ\rangle = \frac{1}{\sqrt{2}} \left(|000\rangle + |111\rangle \right) \tag{1.48}$$

corresponding to a general stabiliser group and the graph state

$$|G\rangle = \frac{1}{\sqrt{8}} \left(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle - |101\rangle - |110\rangle + |111\rangle \right) \tag{1.49}$$

corresponding to the star graph. It turns out that these two graph states are local unitary equivalent

$$\mathbb{1} \otimes H \otimes H |GHZ\rangle = |G\rangle. \tag{1.50}$$

For this reason, one often by a slight abuse of notation identifies certain graph states by the most simple[14] LC-equivalent counterpart.

Remark 1.3. Often, one wants to remove the ambiguity of a basis change and wants to consider LC-inequivalent stabiliser states only. In this case one usually restricts first to graph states, since each stabiliser state is LC-equivalent to some graph state. In a second step one then characterises graph states with respect to LC-equivalence. This can be done solely on the basis of the corresponding graphs since it has been shown that two graph states are LC-equivalent, iff their graphs can be transformed into each other by a sequence of local complementations[15] [26].

[13] This result is also known as a disproof of the conjecture that local unitary equivalence equals local Clifford equivalence.

[14] Here simple refers to the number of terms the state has in the computational basis. This simplification is often possible, since any graph state always has the maximal number of 2^n terms in the computational basis.

[15] A local complementation is an operation performed at a vertex, where all existing edges are deleted and edges to prior unconnected nodes are created.

Chapter 2

Analytical characterisation of the genuine multiparticle negativity

Science is built up with facts, as a house is with stones. But a collection of facts is no more a science than a heap of stones is a house.
Henri Poincaré

In this chapter we present an analytical study of the genuine multiparticle negativity (GMN). First, we show that a renormalised version of the GMN can be expressed as mixed convex roof of the minimum of bipartite negativities [50, 51]. These mixed convex roofs were already studied in the context of entanglement quantification in the bipartite setting in Refs. [52–55]. In our case and contrary to the usual pure state convex roof constructions the renormalised GMN can be efficiently computed using semidefinite programming. Second, we derive analytic expressions of the GMN for two different state families. These are the GHZ-diagonal n-qubit and the cluster-diagonal four-qubit states. These analytic formulas for the GMN in terms of the fidelities of the GHZ and cluster states also provide lower bounds on the genuine multiparticle entanglement of general mixed quantum states.

This chapter is organised as follows: First, we introduce the renormalised GMN and show that it can be expressed as a mixed convex roof. We then compare the original GMN and the renormalised GMN and provide the naturally arising upper and lower bounds to the latter. In the next section we derive an analytic formula for the original and renormalised GMN for n-qubit GHZ-diagonal states and compare our results to the GME-concurrence [76, 80]. We also show that an exact expression can also be obtained for cluster-diagonal four-qubit states, where only lower bounds are known for the GME-concurrence [80–82]. We conclude this chapter with a brief discussion of our results and a short outlook on possible future directions.

2.1 The GMN as a convex roof measure

In this section we introduce a renormalised version of the GMN by changing the normalisation of the witness operator. Our main motivation is the following. So far, we have a good understanding of the GMN in terms of witnesses. In state space, however, there is no satisfying interpretation. By slightly altering the definition the GMN has a direct simple interpretation in the witness *and* in the state space picture.

2.1.1 Modifying the definition of the GMN

Consider any finite-dimensional multiparticle system and recall the definition of the genuine multiparticle negativity [15] as introduced in Paragraph 1.3.4, which is a genuine multiparticle entanglement monotone.

For any state ϱ the renormalised GMN $N_g(\varrho)$ is given by

$$\begin{aligned} N_g(\varrho) = &- \inf \mathrm{tr}\,(\varrho \mathcal{W}) \\ &\text{subjected to: } \mathcal{W} = P_m + Q_m^{T_m}, \\ &\qquad\qquad 0 \leq P_m \\ &\qquad\qquad 0 \leq Q_m \leq \mathbb{1} \quad \text{for all partitions } m|\bar{m}. \end{aligned} \tag{2.1}$$

Compared to the original definition in Eq. (1.30), the only difference is a relaxation in the constraints on the positive operators P_m, which is not bounded by $\mathbb{1}$ anymore. Note that this definition was already used in Ref. [83] to quantify genuine multiparticle entanglement in a device independent manner.

The interesting point is that the renormalised GMN has an interpretation in state space as coming from an optimisation over decompositions of the density matrix ϱ. This is known as the mixed convex roof construction [52–55] and many entanglement measures are defined via such an optimisation. In the present case, one deals with such a mixed convex roof, and this can be derived from the the dual problem [68] to the semidefinite problem in Eq. (2.1). We have:

Theorem 2.1. *Let N_m be the bipartite negativity given by $N_m(\varrho) = \sum_i |\lambda_i^-(\varrho^{T_m})|$, where $\lambda_i^-(\varrho^{T_m})$ are the negative eigenvalues of ϱ^{T_m}. Then the genuine multiparticle negativity equals a mixed convex roof of bipartite negativities. That is*

$$N_g(\varrho) = \min_{\varrho = \sum_m p_m \varrho_m} \sum_m p_m N_m(\varrho_m), \tag{2.2}$$

where the summation runs over all inequivalent partitions $m|\bar{m}$ of the system and the minimisation is performed over all mixed state decompositions of the state $\varrho = \sum_m p_m \varrho_m$.

The proof of this Theorem can be found in A.1.

Note that the optimisation in Eq. (2.2) can also be written in a different way: If one defines for an arbitrary multiparticle quantum state the quantity $\mu(\varrho) = \min_m N_m(\varrho)$ as the bipartite negativity, minimised over all bipartitions, then the multiparticle negativity can be written as

$$N_g(\varrho) = \min_{\varrho = \sum_k p_k \varrho_k} \sum_k p_k \mu(\varrho_k), \tag{2.3}$$

where now the minimisation is over all decompositions $\varrho = \sum_k p_k \varrho_k$ into mixed states and k does not label the bipartitions anymore. In this way, the connection to the usual convex roof construction (see Ref. [73] and also Eq. (2.21) below) becomes more transparent. In general, however, mixed and pure state convex roofs are extremely difficult to compute. In this respect it is important to highlight that the renormalised GMN can be computed using semidefinite programming (SDP).

2.1.2 Comparison with the original definition of the GMN

First, we can state that all the properties of the GMN also hold for the renormalised definition and one additional property is new.

Lemma 2.2. *For the renormalised multiparticle negativity N_g as defined in Eq. (2.1) all the properties (i) to (vi) listed in Lemma 1.2 hold. Additionally, it has the following property:*

vii If $|\psi\rangle$ is a pure state, then

$$N_g(|\psi\rangle) = \min_m N_m(|\psi\rangle), \tag{2.4}$$

where the minimisation is performed over all bipartite splittings $m|\bar{m}$ of the system.

Proof. The first properties from Lemma 2.2 can be proved directly as in Lemma 1.2 by modification of the respective proofs in Refs. [15, 24]. Concerning statement (vii), note that for a pure state $\varrho = |\psi\rangle\langle\psi|$ there is only a single (and trivial) decomposition, namely $\varrho = 1 \cdot |\psi\rangle\langle\psi|$. □

Note that due to property (ii) of Lemmata 1.2 and 2.2 both versions of the GMN are entanglement monotones. Naturally, the question arises how the renormalised GMN performs in detecting genuine multiparticle

entangled states. If one is interested in the quantification of genuine multiparticle entanglement with the help of an entanglement monotone then either of the two monotones can be used as they are non-zero on the same set of states.[1] One directly has:

Corollary. *For all* ϱ, $\tilde{N}_g(\varrho) \leq N_g(\varrho)$ *and* $\tilde{N}_g(\varrho) = 0 \Leftrightarrow N_g(\varrho) = 0$.

Another feature of the renormalised GMN are the natural upper *and* lower bounds, arising from Eq. (2.1) and Eq. (2.2) in Theorem 2.1. First, as for the original GMN, every witness $\mathcal{W}$, which satisfies the constraints in Eq. (2.1) provides a lower bound on the renormalised GMN

$$- tr\,(\mathcal{W}\varrho) \leq N_g(\varrho). \tag{2.5}$$

Second, every mixed state decomposition of a state $\varrho = \sum_m p_m \varrho_m$, $p_m \geq 0$, $\sum_m p_m = 1$ provides an upper bound on the renormalised GMN

$$\sum_m p_m N_m(\varrho_m) \geq N_g(\varrho). \tag{2.6}$$

This property makes the renormalised GMN easier to compute analytically. Note that the upper bounds of the renormalised GMN also provide upper bounds for $\tilde{N}_g$ since $\tilde{N}_g \leq N_g$.

Finally, note that for pure states the renormalised GMN can directly be computed with the help of Lemma 2.2. Sometimes it coincides with the original GMN for pure states, and sometimes not. An example is the three-qubit GHZ state $|GHZ\rangle = 1/\sqrt{2}(|000\rangle + |111\rangle$, where $N_g(|GHZ\rangle) = \tilde{N}_g(|GHZ\rangle) = 1/2$. On the other hand, for the three-qubit W state $|W\rangle = 1/\sqrt{3}(|001\rangle + |010\rangle + |100\rangle)$, $N_g(|W\rangle) = \sqrt{2}/3 \approx 0.47$ and $\tilde{N}_g(|W\rangle) \approx 0.43$.

2.2 Analytic computation of the GMN

In this section we use our previous results to provide analytic formulas of the GMN for two important families of multi-qubit states. These are the n-qubit GHZ-diagonal and four-qubit cluster-diagonal states. The idea in both cases is to construct for each family of states a family of witnesses lower bounding the GMN and a family of decompositions, which results in upper bounds. Since the bounds coincide and hold true for the original and the renormalised GMN they provide closed formulas for both monotones.

2.2.1 Graph-diagonal states

Both state families are connected to so-called graph states, which were introduced in Section 1.4.1. Recall that the main idea was to introduce a set of commuting stabiliser operators g_i corresponding to the vertices of a graph, which then give rise to an unique common eigenstate, called graph state.

One can extend this framework by considering all common eigenstates of the stabilising operators. We label those 2^n different states by their eigenvalues of ± 1 on the stabilising operators g_k, such that $g_i|a_1 a_2 \ldots a_n\rangle = a_i|a_1 a_2 \ldots a_n\rangle$ with $a_i = \pm$. Note that these states are all orthogonal $\langle a_1 \ldots a_n|b_1 \ldots b_n\rangle = \prod_{i=1}^N \delta_{a_i b_i}$ and thus form a basis in the n-qubit Hilbert space, the so-called graph state basis. Mixed states, which are diagonal in the graph state basis are determined by their fidelities

$$F_{a_1 a_2 \ldots a_n} = \langle a_1 a_2 \ldots a_n|\varrho|a_1 a_2 \ldots a_n\rangle, \tag{2.7}$$

i.e.,

$$\varrho = \sum_{a_1, a_2, \ldots, a_n} F_{a_1 a_2 \ldots a_n}|a_1 a_2 \ldots a_n\rangle\langle a_1 a_2 \ldots a_n|. \tag{2.8}$$

They are called graph-diagonal and have the property that they are invariant under the group generated by the stabilising operators, i.e., $g_i \varrho g_i^\dagger = \varrho$ for all i.

Note that an arbitrary state ϱ can be transformed into a graph-diagonal state by the symmetrisation operation

$$\varrho_{graph-diag} = \frac{1}{2^n} \sum_{g \in G} g \varrho g^\dagger, \tag{2.9}$$

[1] One can modify the original MATLAB implementation of the PPT mixer [69] to get an implementation of the renormalised GMN, by simply commenting line 90 in the "entmon.m" file.

where the summation runs over all group elements of the group generated by the g_i. Since all $g \in G$ consist of local Pauli operators only this symmetrisation does not increase entanglement and hence $N_g(\varrho_{graph-diag}) \leq N_g(\varrho)$.

2.2.2 n-qubit GHZ-diagonal states

In this paragraph we consider generalised Greenberger-Horne-Zeilinger (GHZ) states. These states are diagonal in the n-qubit GHZ-basis consisting of 2^n states $|\psi_i\rangle = 1/\sqrt{2}\,(|x_1 x_2 \dots x_n\rangle \pm |\bar{x}_1 \bar{x}_2 \dots \bar{x}_n\rangle)$, where $x_j, \bar{x}_j \in \{0,1\}$ and $x_j \neq \bar{x}_j$. In the three-qubit case this basis consists of the states $1/\sqrt{2}\,(|000\rangle \pm |111\rangle)$, $1/\sqrt{2}\,(|001\rangle \pm |110\rangle)$, $1/\sqrt{2}\,(|010\rangle \pm |101\rangle)$ and $1/\sqrt{2}\,(|011\rangle \pm |100\rangle)$. Note that these states are invariant under the group generated by $g_1 = X^{(1)}X^{(2)}\dots X^{(n)}$ and $g_i = X^{(1)}X^{(i)}$ for $2 \leq i \leq n$. This group is local unitary equivalent to the stabiliser group corresponding to the star graph as shown in Fig. 1.3, where each node is connected to node one.

A three-qubit state diagonal in the GHZ basis is of the form

$$\varrho = \begin{pmatrix} \lambda_0 & & & & & & & \mu_0 \\ & \lambda_1 & & & & & \mu_1 & \\ & & \lambda_2 & & & \mu_2 & & \\ & & & \lambda_3 & \mu_3 & & & \\ & & & \mu_3 & \lambda_3 & & & \\ & & \mu_2 & & & \lambda_2 & & \\ & \mu_1 & & & & & \lambda_1 & \\ \mu_0 & & & & & & & \lambda_0, \end{pmatrix}, \tag{2.10}$$

with $\lambda_i, \mu_i \in \mathbb{R}$.[2] A general n-qubit GHZ-diagonal state would have the same shape, with 2^{n-1} independent real λ_i on the diagonal and corresponding real μ_i on the anti-diagonal. The eigenvalues of these states are $\lambda_i \pm \mu_i$, $0 \leq i < 2^{n-1}$. Hence to be a valid density matrix one needs $\lambda_i \geq 0$ and $|\mu_i| \leq \lambda_i$ for all $0 \leq i < 2^{n-1}$.

We now make use of the special structure of this class of states to construct explicit upper and lower bounds, which are valid for both versions of the GMN.

Lemma 2.3. *For all GHZ-diagonal n-qubit states*

$$N_g(\varrho) \leq \max_i \{0, |\mu_i| - w_i\} = \max_i \left\{0, F_i - \frac{1}{2}\right\}, \tag{2.11}$$

where $w_i = \sum_{k \neq i} \lambda_k$ and $F_i = \langle \psi_i | \varrho | \psi_i \rangle$ denotes the fidelity with the GHZ-basis state $|\psi_i\rangle$. $N_g(\varrho) \leq \max_i 0, |\mu_i| - w_i$ also holds for the slightly more general case with complex μ_i on the anti-diagonal.

Proof. We will prove the statement for the three-qubit case, a generalisation is straightforward. First, consider the case where the right-hand-side of Eq. (2.11) is non-zero. Without loss of generality one can assume that the maximum is achieved for $i = 0$ and thus we have $|\mu_0| \geq \sum_{k=1}^{3} \lambda_k$. Let $p_k = \lambda_k / (\sum_{k=1}^{3} \lambda_k)$ for $1 \leq k \leq 3$, then $\sum_i p_i = 1$ and

$$p_k |\mu_0| \geq \lambda_k. \tag{2.12}$$

From the positivity of ϱ it follows that $|\mu_i| \leq \lambda_i$ and so

$$p_k \lambda_0 \geq p_k |\mu_0| \geq \lambda_k \geq |\mu_k|. \tag{2.13}$$

Using these weights one decomposes ϱ into a convex combination $\varrho = \sum_{k \neq 0} \tilde{p}_k \varrho_k$ with

$$\varrho_1 = \frac{1}{\tilde{p}_1} \begin{pmatrix} p_1\lambda_0 & & & & & & & p_1\mu_0 \\ & \lambda_1 & & & & & \mu_1 & \\ & & 0 & & & 0 & & \\ & & & 0 & 0 & & & \\ & & & 0 & 0 & & & \\ & & 0 & & & 0 & & \\ & \mu_1^* & & & & & \lambda_1 & \\ p_1\mu_0^* & & & & & & & p_1\lambda_0, \end{pmatrix}, \tag{2.14}$$

[2] GHZ-diagonal states have real μ_i, but we stress that all of our results hold true for states with complex μ_i as well.

$$\varrho_2 = \frac{1}{\tilde{p}_k}\begin{pmatrix} p_2\lambda_0 & & & & & & & p_2\mu_0 \\ & 0 & & & & & 0 & \\ & & \lambda_2 & & & \mu_2 & & \\ & & & 0 & 0 & & & \\ & & & 0 & 0 & & & \\ & & \mu_2^* & & & \lambda_2 & & \\ & 0 & & & & & 0 & \\ p_2\mu_0^* & & & & & & & p_2\lambda_0, \end{pmatrix}, \tag{2.15}$$

$$\varrho_3 = \frac{1}{\tilde{p}_k}\begin{pmatrix} p_3\lambda_0 & & & & & & & p_3\mu_0 \\ & 0 & & & & & 0 & \\ & & 0 & & & 0 & & \\ & & & \lambda_3 & \mu_3 & & & \\ & & & \mu_3^* & \lambda_3 & & & \\ & & 0 & & & 0 & & \\ & 0 & & & & & 0 & \\ p_3\mu_0^* & & & & & & & p_3\lambda_0, \end{pmatrix} \tag{2.16}$$

and $\tilde{p}_k = 2(p_k\lambda_0 + \lambda_k)$.

To calculate the upper bound resulting from this decomposition we first have to compute the action of partial transposition with respect to subsystem A, B and C on three-qubit GHZ-diagonal state. One directly sees that the partial transposition permutes the anti-diagonal elements. In the three-qubit case transposition of the first qubit exchanges $\mu_0 \leftrightarrow \mu_1$, $\mu_2 \leftrightarrow \mu_3$ and the corresponding conjugate pairs. The partial transposition of the second qubit exchanges $\mu_0 \leftrightarrow \mu_2$, $\mu_1 \leftrightarrow \mu_3$ and conjugate pairs and the partial transposition on the last qubit exchanges $\mu_0 \leftrightarrow \mu_3^*$, $\mu_1 \leftrightarrow \mu_2^*$ and conjugate pairs. So $\tilde{p}_k \varrho_k^{T_k}$ has the following four non-zero eigenvalues

$$\{p_k\lambda_0 + |\mu_k|, p_k\lambda_0 - |\mu_k|, \lambda_k + p_k|\mu_0|, \lambda_k - p_k|\mu_0|\}\,. \tag{2.17}$$

Taking into account Eqs. (2.12) and (2.13) the only non-positive eigenvalue is $\lambda_k - p_k|\mu_0|$ and thus $\tilde{p}_k N_k(\varrho_k) = p_k|\mu_0| - \lambda_k$. This results in the conjectured upper bound

$$N_g(\varrho) \leq \sum_{k=1}^{3} p_k N_k(\varrho_k) = \sum_{k=1}^{3} p_k|\mu_0| - \lambda_k = |\mu_0| - \sum_{k=1}^{3}\lambda_k. \tag{2.18}$$

This bound can be rewritten as $|\mu_0| - \sum_{k=1}^{3}\lambda_k = |\mu_0| + \lambda_0 - \frac{1}{2}$, since $\mathrm{tr}\varrho = 2\sum_{k=0}^{3}\lambda_k = 1$. If we then use the fidelities $F_0 = \langle\psi_0|\varrho|\psi_0\rangle = \lambda_0 + \mu_0$ and $F_1 = \langle\psi_1|\varrho|\psi_1\rangle = \lambda_0 - \mu_0$ with $|\psi_0\rangle = 1/\sqrt{2}(|000\rangle + |111\rangle)$ and $|\psi_1\rangle = 1/\sqrt{2}(|000\rangle - |111\rangle)$, then $|\mu_0| + \lambda_0 = \max\{F_0, F_1\}$ for μ_0 real, which proves the alternative expression $N_g(\varrho) \leq \max_i\left\{0, F_i - \frac{1}{2}\right\}$ in Eq. (2.11).

If, on the other hand, the right-hand-side of Eq. (2.11) is zero, then $|\mu_0| \leq \sum_{k\neq 0}\lambda_k$, which is known to be a necessary and sufficient criterion for biseparability [23] and for all biseparable states ϱ, $N_g(\varrho) = 0$. □

Lemma 2.4. *Consider a n-qubit GHZ-diagonal state ϱ then there exists a fully decomposable witness $\mathcal{W}$ satisfying the properties in Eq. (2.1), such that*

$$N_g(\varrho) \geq -\mathrm{tr}(\mathcal{W}\varrho) = \min_i\{0, |\mu_i| - w_i\} = \max_i\left\{0, F_i - \frac{1}{2}\right\}, \tag{2.19}$$

where $w_i = \sum_{k\neq i}\lambda_k$. $N_g(\varrho) \geq \max_i 0, |\mu_i| - w_i$ also holds for the slightly more general case with complex μ_i on the antidiagonal.

Proof. As in the last proof we consider the three-qubit case. Without loss of generality the minimum in inequality (2.19) is achieved for $i = 0$. Then the position of μ_0 in ϱ in the computational basis is given by the tuple $(000, 111)$. From this tuple we construct the witness $\mathcal{W} = \frac{1}{2}\mathbb{1} - |\phi\rangle\langle\phi|$, with $|\phi\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$. In the more general case, where the $\mu_i \in \mathbb{C}$ one would insert an additional phase $e^{i\arg\mu_0}$ in front of $|111\rangle$.

The witness is of the form of ϱ as in Eq. (2.10). From the discussion in the proof of Lemma 2.3 it follows that $\mathcal{W}^{T_m} \geq 0$. Hence, $\mathcal{W}$ is fully decomposable with $P_m = 0$ and $Q_m = W^{T_m}$. In case the minimum equals zero the witness is given by $\mathcal{W} = 0$, $P_m = 0$ and $Q_m = 0$. This proves the claim. □

In summary, we have:

Corollary. *For all GHZ-diagonal n-qubit states ϱ*

$$N_g(\varrho) = \max_i \{0, |\mu_i| - w_i\} = \max_i \left\{0, F_i - \frac{1}{2}\right\}, \tag{2.20}$$

where $F_i = \langle\psi_i|\varrho|\psi_i\rangle$ denotes the fidelity with the GHZ-basis state ψ_i. $N_g(\varrho) = \max_i 0, |\mu_i| - w_i$ holds also true for the slightly more general case with complex μ_i on the anti-diagonal.

Three remarks are in order at this point. First, for general states the right hand side of Eq. (2.20) still gives a lower bound on the renormalised GMN since every state can be transformed into a GHZ-diagonal by means of local operations only. Thus the analytic expression in Eq. (2.20) might be used to estimate genuine multiparticle entanglement also for a general state. Second, we calculated the value for the renormalised GMN $N_g(\varrho)$ as defined in Eq. (2.1), but the same result holds for the GMN $\tilde{N}_g(\varrho)$ according to Eq. (1.30). This is because the witness constructed in the proof of Lemma 2.4 fulfils also the conditions of Eq. (1.30) and Lemma 2.3 delivers an upper bound due to Corollary 2.1.2. Third, note that the analytic formula we found coincides with the maximal violation of the biseparability criteria derived in Ref. [23].

Further, it is interesting to compare our expression for the GMN to the analytic formula of the genuine multiparticle concurrence (GME-concurrence) [80] for GHZ-diagonal n-qubit states [76]. The GME-concurrence is defined as follows: For a bipartite pure state $|\psi\rangle$ the concurrence is given by $C^{A|B}(\psi) = \sqrt{2[1 - Tr(\varrho_A^2)]}$ where ϱ_A is the reduced state on Alice's system. For a pure multiparticle state $|\phi\rangle$, the genuine multiparticle concurrence is defined as the minimum of the bipartite concurrences $C^{\mathrm{gme}}(\phi) = \min_{m|\bar{m}} C^{m|\bar{m}}(\phi)$, minimised over all bipartitions. Finally, for mixed states the measure is given by the convex roof construction

$$C^{\mathrm{gme}}(\varrho) = \min_{\varrho=\sum_k p_k|\phi_k\rangle\langle\phi_k|} \sum_k p_k C^{\mathrm{gme}}(\phi_k). \tag{2.21}$$

Note that contrary to the mixed convex roof optimisation in Eq. (2.2) here only pure state decompositions are involved. This pure state convex roof, however, is in general nearly impossible to compute and therefore one relies on lower bounds for practical applications.

The GME-concurrence has been computed for GHZ-diagonal states [76] and one observes that up to a factor of two both expressions coincide. There are, however, deeper connections: For a pure multi-qubit state one has

$$N_g(\phi) \leq C^{\mathrm{gme}}(\phi). \tag{2.22}$$

This follows directly from known relations between the bipartite negativity and the bipartite concurrence [84]. Since the GMN can be defined via the *mixed* convex roof, which is an optimisation over a larger set than the pure convex roof of the GME-concurrence, the general bound

$$N_g(\varrho) \leq C^{\mathrm{gme}}(\varrho). \tag{2.23}$$

holds for all mixed states. Therefore, Theorem 2.1 provides a way to obtain lower bounds on the GME-concurrence via semidefinite programming or analytical calculations. We will see in the next section that the value of the $N_g(\varrho)$ is much more sensitive to entanglement than the known lower bounds on the GME-concurrence.

2.2.3 Four-qubit cluster-diagonal states

In this section we consider the linear graph having four vertices as shown in Fig. 1.3. The corresponding stabilising operators as defined in Eq. (1.36) are given by

$$\begin{aligned} g_1 &= XZ\mathbb{1}\mathbb{1}, \qquad g_2 = ZXZ\mathbb{1}, \\ g_3 &= \mathbb{1}ZXZ \text{ and } g_4 = \mathbb{1}\mathbb{1}ZX. \end{aligned} \tag{2.24}$$

As already discussed, the corresponding graph state basis is given by $|++++\rangle, |+++-\rangle, \ldots, |----\rangle$. We will also write this states as $|ijlk\rangle$, with $i, j, l, k \in \{+, -\}$ and denote by $\bar{k}$ the complement of k. Then we consider the graph-diagonal state

$$\varrho = \sum_{i,j,k,l} F_{ijkl}|ijkl\rangle\langle ijkl| \tag{2.25}$$

and wish to compute the GMN for these states. Note that in the literature the four-qubit cluster state is often defined via the local unitary equivalent stabilising operators $\tilde{g}_1 = ZZ\mathbb{1}\mathbb{1}$, $\tilde{g}_2 = XXZ\mathbb{1}$, $\tilde{g}_3 = \mathbb{1}ZXX$ and $\tilde{g}_4 = \mathbb{1}\mathbb{1}ZZ$. Then the corresponding eigenstate to eigenvalue $+1$ on all $\tilde{g}_i$ is the familiar cluster state

$$|CL\rangle = \frac{1}{2}(|0000\rangle + |1100\rangle + |0011\rangle + |1111\rangle) \tag{2.26}$$

in the computational basis.

For discussing genuine multiparticle entanglement of cluster-diagonal states, the following two classes of witnesses

$$\mathcal{W}_{\alpha\beta\gamma\delta} = \frac{\mathbb{1}}{2} - |\alpha\beta\gamma\delta\rangle\langle\alpha\beta\gamma\delta| - \frac{1}{2}\sum_{i,j} |\bar{\alpha}ij\bar{\delta}\rangle\langle\bar{\alpha}ij\bar{\delta}|, \tag{2.27}$$

$$\mathcal{W}_{\alpha\beta\gamma\delta\mu\nu} = \frac{\mathbb{1}}{2} - |\alpha\beta\gamma\delta\rangle\langle\alpha\beta\gamma\delta| - |\bar{\alpha}\mu\nu\bar{\delta}\rangle\langle\bar{\alpha}\mu\nu\bar{\delta}|, \tag{2.28}$$

have turned our to be useful. It has been shown that these witnesses provide necessary and sufficient criteria to detect genuinely multiparticle entanglement in these states [85]. Based on these, Chen et al. [86] provided a closed formula for the genuine multiparticle relative entropy of entanglement as entanglement monotone. Here we provide a closed formula for the GMN for four-qubit cluster-diagonal states.

In terms of the fidelity the expectation values of the witnesses in Eqs. (2.27) and (2.28) read

$$\begin{aligned} \mathrm{tr}(\mathcal{W}_{\alpha\beta\gamma\delta}\varrho) &= -F_{\alpha\beta\gamma\delta} + \frac{1}{2}\sum_{ij} F_{\alpha ij\bar{\delta}} + F_{\bar{\alpha}ij\delta} + F_{\alpha ij\delta}, \\ \mathrm{tr}(\mathcal{W}_{\alpha\beta\gamma\delta\mu\nu}\varrho) &= -F_{\alpha\beta\gamma\delta} - F_{\bar{\alpha}\mu\nu\bar{\delta}} + \frac{1}{2}. \end{aligned} \tag{2.29}$$

First we note that all of these witnesses can be used to bound the GMN from below:

Lemma 2.5. *For all partitions $m|\bar{m}$ of the four partied system $ABCD$ there exists $0 \leq Q_m, \tilde{Q}_m \leq \mathbb{1}$, such that*

$$\mathcal{W}_{\alpha\beta\gamma\delta} = Q_m^{T_m} \quad \text{and} \quad \mathcal{W}_{\alpha\beta\gamma\delta\mu\nu} = \tilde{Q}_m^{T_m}. \tag{2.30}$$

Proof. One can easily compute the eigenvalues of $Q_m = \mathcal{W}_{++++}^{T_m}$ and $\tilde{Q}_m = \mathcal{W}_{++++ij}^{T_m}$ for all $i, j \in \{+, -\}$ to be 0 and $\frac{1}{2}$. For any other witness $\mathcal{W}_{\alpha\beta\gamma\delta}$ ($\mathcal{W}_{\alpha\beta\gamma\delta\mu\nu}$) there exists a local unitary transformations, such that $\mathcal{W}_{++++}$ ($\mathcal{W}_{++++ij}$) transforms into it. □

Taking into account that each of the above witnesses gives a lower bound on the GMN [see Eq. (2.5)] we have that

$$-\min_{\alpha,\beta,\gamma,\delta,\mu,\nu} \{\mathrm{tr}(\mathcal{W}_{\alpha\beta\gamma\delta}\varrho)\} \cup \{\mathrm{tr}(\mathcal{W}_{\alpha\beta\gamma\delta\mu\nu}\varrho)\} \cup \{0\} \leq N_g(\rho), \tag{2.31}$$

for all four-qubit cluster-diagonal states. As done for the GHZ-diagonal states we can use specific decompositions [see Eq. (2.2)] to construct upper bounds on the GMN, which result in an analytic formula for all four-qubit cluster states.

Theorem 2.6. *Let $\varrho = \sum_{\alpha\beta\gamma\delta} F_{\alpha\beta\gamma\delta}|\alpha\beta\gamma\delta\rangle\langle\alpha\beta\gamma\delta|$ be diagonal in the cluster graph basis, then the GMN of that state is given by*

$$N_g(\varrho) = -\min_{\alpha,\beta,\gamma,\delta,\mu,\nu} \{\mathrm{tr}(\mathcal{W}_{\alpha\beta\gamma\delta}\varrho)\} \cup \{\mathrm{tr}(\mathcal{W}_{\alpha\beta\gamma\delta\mu\nu}\varrho)\} \cup \{0\}\,. \tag{2.32}$$

This means that effectively the largest violation of the witnesses in Eqs. (2.27) and (2.28) gives the value of the GMN.

The proof of this Theorem is given in Appendix A.2.

Let us discuss some examples. For the graph state mixed with white noise $\varrho = p|++++\rangle\langle++++| + (1-p)\mathbb{1}/16$ we obtain $N_g(\varrho) = \max(\frac{13p-5}{16}, 0)$, which gives the exact threshold $p > 5/13$ for genuine multiparticle entanglement [85]. In Fig. 2.1 a two-parameter family is shown, which is is genuine multiparticle entangled in three regions and biseparable in one. In each of the regions a different optimal witness gives the GMN.

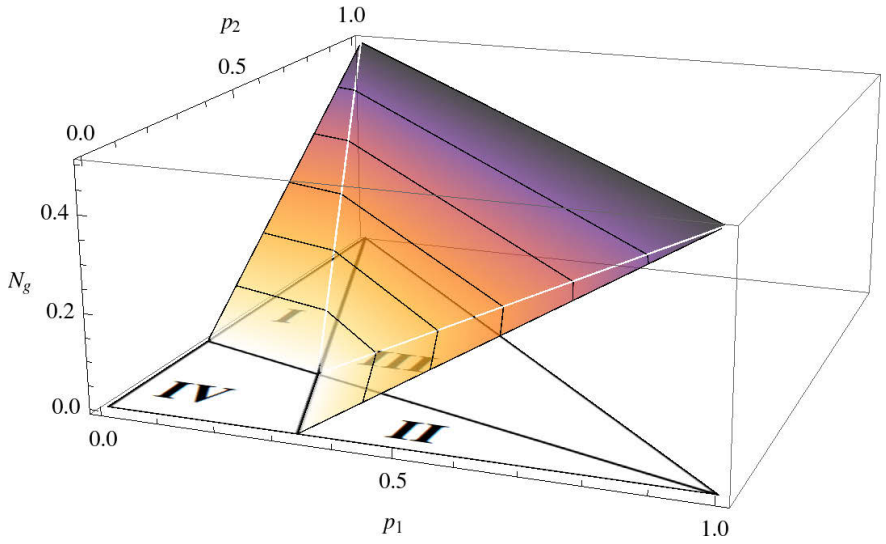


Figure 2.1: The GMN for a two-parameter family of cluster-diagonal states, given by $\varrho = p_1|++++\rangle\langle++++| + p_2|-++-\rangle\langle-++-| + (1-p_1-p_2)\frac{1}{2}(\sigma_1+\sigma_2)$, with biseparable $\sigma_1 = \frac{1}{2}(|++-+\rangle\langle++-+| + |+-++\rangle\langle+-++|)$ and $\sigma_2 = \frac{1}{2}(|-+--\rangle\langle-+--| + |--+-\rangle\langle--+-|)$. In the regions I to III the GMN N_g is given by the negative expectation value of different witnesses [see Eq. (2.32)]. In I it corresponds to $-\mathrm{tr}(\mathcal{W}_{-++-}\varrho) = \frac{1}{4}(p_1+3p_2-1)$, in II it is given by $-\mathrm{tr}(\mathcal{W}_{++++++}\varrho) = p_1 + p_2 - \frac{1}{2}$ and in region III it is $-\mathrm{tr}(\mathcal{W}_{++++}\varrho) = \frac{1}{4}(3p_1 + p_2 - 1)$. In the remaining region IV the state is biseparable.

Finally, we compare our results to computable lower bounds on the GME-concurrence introduced in Refs. [80, 82] and general lower bounds on the linear entropy based genuine multiparticle entanglement measure in Ref. [81]. To compare the performance we calculate the value p, down to which $\varrho = p|++++\rangle\langle++++| + (1-p)\mathbb{1}/16$ is still detected as genuine multiparticle entangled. Using the general bound of the GME-concurrence in Ref. [80] we found that even the pure four-qubit cluster-diagonal state is not detected. Using instead a set of inequalities build to detect genuine multiparticle entanglement in n-qubit Dicke states [82] we found ϱ to be detected for $p > 0.982$.[3] A better detection was achieved with the general lower bound on the genuine multiparticle entanglement measure given by Theorem 1 of Ref. [81]. We found that the state ϱ was detected as genuine multiparticle entangled for $p > 7/15 \approx 0.47$, which is closer to the exact threshold $p > 5/13 \approx 0.38$ but not the exact value. We can therefore conclude that although the analytic formula for the GME-concurrence is equivalent to ours for n-qubit GHZ-diagonal states, the lower bounds for four-qubit cluster-diagonal states do not match our analytic results.

2.3 Conclusions

In conclusion we have shown that the renormalised genuine multiparticle negativity can be expressed in two equivalent ways: as an optimisation over suitable normalised fully decomposable witnesses as given by Eq. (2.1) and as mixed convex roof of the minimal bipartite negativity as given by Eq. (2.2). As a direct consequence of these equivalent definitions there are naturally arising lower and upper bounds, which we used to obtain an exact algebraic prescription of the genuine multiparticle negativity for the n-qubit GHZ-diagonal and four-qubit cluster-diagonal states. These analytic expressions can also be used to obtain lower bounds on the genuine multiparticle negativity for arbitrary n-qubit states.

There are several questions arising, which one might investigate in the future. First, since the scheme to obtain the analytic expression is quite general it should be possible to find closed expressions for other highly symmetric state families such as other graph-diagonal states [61] or states with $U \otimes U \otimes U$ symmetry [87].

[3] Note that we applied local filters to the state $\varrho \mapsto N(F_A^\dagger \otimes F_B^\dagger \otimes F_C^\dagger \otimes F_D^\dagger \varrho F_A \otimes F_B \otimes F_C \otimes F_D)$ to enhance its detectability, where N is a normalisation and the F_i are linear maps on the single qubit systems.

Second, it would be desirable to obtain an operational interpretation for the genuine multiparticle negativity. As the bipartite logarithmic negativity is the upper bound for distillable entanglement [51] one may speculate that our monotone is connected to the distillation rate of genuine multiparticle entangled states. Also, the multiparticle negativity may be related to different entanglement classes in the multiparticle case and the dimensionality of multiparticle entanglement [88, 89].

Finally, recall that the shareability of quantum correlations among many parties is limited and these restrictions are known as monogamy relations [90–93]. For example, for a three-qubit system the bipartite entanglement of the splitting $A|BC$ as measured by the concurrence is given by the entanglement in the reduced marginals plus the three tangle τ_3 as a genuine tripartite contribution, $C^2_{A|BC} = C^2_{AB} + C^2_{AC} + \tau_3$ [90]. It would be very interesting to derive similar relations for the genuine multiparticle negativity.

Chapter 3

Genuine multiparticle entanglement in spin chains

After climbing a great hill, one only finds that there are many more hills to climb.
Nelson Mandela

In this chapter we study the scaling and spatial distribution of genuine multiparticle entanglement at a quantum phase transition in one-dimensional spin models. Our results are enabled by recent progress in the theory of multiparticle entanglement [15, 56, 57] in combination with an explicit determination of reduced k-particle states in the transverse XY-model [94–97]. We consider the genuine multiparticle negativity [15] as a measure genuine multiparticle entanglement for the reduced three- and four-particle states and demonstrate that its derivative diverges at the critical point. For both cases we show that the entanglement obeys finite-size scaling, which can be used to compute the critical exponent for the infinite system from finite-size data.

This chapter is organised as follows: First, we review the transverse XY-model, the different phases of its ground state and the method to explicitly determine the reduced k-particle states [94–97]. Second, we investigate spatial distribution and strength of genuine multiparticle entanglement in three- and four-spin reduced states of the transverse Ising model, which is a special case of the XY-model. Finally, we study the scaling of genuine multiparticle entanglement same reduced states close to the quantum phase transition.

3.1 The model

We consider the one-dimensional XY-model with transverse magnetic field on L particles and periodic boundary conditions [94–97]. The Hamiltonian of this model is given by

$$H = -\sum_{i=1}^{L} \frac{\lambda}{4}[(1+\gamma)\sigma_x^{(i)}\sigma_x^{(i+1)} + (1-\gamma)\sigma_y^{(i)}\sigma_y^{(i+1)}] + \frac{1}{2}\sigma_z^{(i)}, \tag{3.1}$$

where the coupling constant $\lambda \geq 0$ tunes the strength of the nearest neighbour coupling with respect to the external magnetic field. The parameter γ sets the anisotropy of the system and connects the Ising model ($\gamma = 1$) with the isotropic XY-model ($\gamma = 0$). In the thermodynamic limit and for $0 < \gamma \leq 1$ the ground state of the model undergoes a quantum phase transition at the critical point $\lambda_c = 1$. For $\lambda = 0$ there is an unique ground state, where all spins are aligned in the direction of the magnetic field and there is no magnetisation in the XY-plane. For $\lambda \to \infty$ the ground state is two-fold degenerate and hence is an equal mixture of these two states. At the quantum phase transition the systems ground state changes from being non-degenerate to degenerate, accompanied by an abrupt change of the magnetisation in the x-direction, which is zero for $\lambda < 1$ and finite for $\lambda \geq 1$. We use the XY-model as a paradigm for our approach since it allows one to study a phase transition with analytical rigor [94–97].

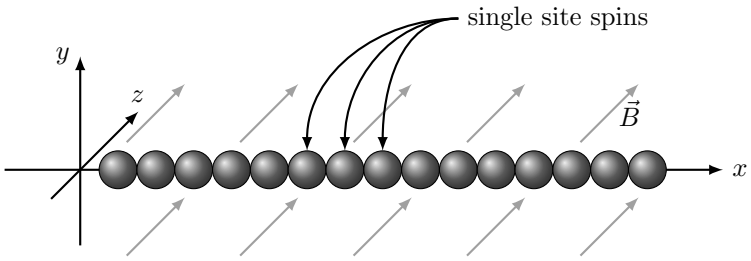


Figure 3.1: In this figure, a finite length one-dimensional spin chain with open boundary condition in an external magnetic field is shown. Solid state models of these kind usually consist of a chain of interacting spins in a external magnetic field, where the first and last spins are considered to be neighbours (periodic boundary conditions) or open ends (open boundary conditions). Analytic models such as the one-dimensional XY-model are often used to study the behaviour of observables close to critical points and phase transitions. In this chapter, we use the XY-model as paradigm to study the scaling and distribution of the 3- and 4-site genuine multiparticle entanglement of the ground state in the vicinity of a quantum phase transition. Recall that a quantum phase transition is not driven by temperature but by the proportion of the strengths of external magnetic field and internal interaction.

3.1.1 Diagonalizing the XY-model

We give a brief review of the analytical diagonalisation of

$$H = -\sum_{i=1}^{L} \frac{\lambda}{4}[(1+\gamma)\sigma_x^{(i)}\sigma_x^{(i+1)} + (1-\gamma)\sigma_y^{(i)}\sigma_y^{(i+1)}] + \frac{1}{2}\sigma_z^{(i)}. \tag{3.2}$$

The strategy is as follows [94–97].

- First, we apply the so-called Jordan-Wigner transformation, which maps the model onto a fermionic Fock space with creation and annihilation operators.
- Second, the transformed system decouples into a direct sum of four dimensional Hilbert spaces by using discrete Fourier transformation.
- Finally, these subspaces can be diagonalised, giving access to the ground state, the energy spectrum and expectation values of finite products of one-site Pauli operators.

First, apply the Jordan-Wigner transformation, which transforms the spin operators S_j^α, $j = 1, \ldots, L$, $\alpha = x, y, z$ into fermionic creators $c_j^\dagger$ and annihilators c_j. It is composed of two intermediate transformations

$$a_j^\dagger = S_x^j + iS_y^j \text{ and } a_j = S_x^j - iS_y^j, \tag{3.3}$$

where $a_j^\dagger$ and a_j are hard-core bosonic creation and annihilation operators. The fermionic operators are then obtained by

$$c_j = \exp\left\{\pi i \sum_{k=1}^{j-1} a_k^\dagger a_k\right\} a_j, \quad c_j^\dagger = a_j^\dagger \exp\left\{-\pi i \sum_{k=1}^{j-1} a_k^\dagger a_k\right\}. \tag{3.4}$$

The new operators obey the fermionic anti-commutation relation

$$\left\{c_i^\dagger, c_j\right\} = \delta_{ij} \text{ and } \{c_i, c_j\} = \left\{c_i^\dagger, c_j^\dagger\right\} = 0. \tag{3.5}$$

Applying the Jordan-Wigner transformation to the Hamiltonian (3.2) yields

$$H = \frac{L}{2} + \frac{\lambda}{2}\left[\left(c_L^\dagger c_1 + \gamma c_L^\dagger c_1\right) + H.c.\right]\left(\exp\left\{i\pi \sum_{j=1}^{L} c_j^\dagger c_j\right\} + 1\right) - \frac{\lambda}{2}\sum_{i=1}^{L}\left(c_i^\dagger c_{i+1} + \gamma c_i^\dagger c_{i+1}^\dagger\right) + H.c. + c_i^\dagger c_i, \tag{3.6}$$

where $H.c.$ denotes the Hermitian conjugate of the parenthesised expression in front. For L large the second term can be neglected. For odd L the second term vanishes. Thus by restriction to an odd number of particles, one can proceed the diagonalisation procedure with the Hamiltonian

$$H = \frac{L}{2} - \sum_{i=1}^{L} \frac{\lambda}{2} \left(c_i^\dagger c_{i+1} + \gamma c_i^\dagger c_{i+1}^\dagger \right) + H.c. + c_i^\dagger c_i. \tag{3.7}$$

Next, one performs a Fourier transform to decouple the Hamiltonian into a direct product of Hilbert spaces preserving the fermionic anti-commutation relations. Let $\phi_p = 2\pi p/L$, then the operators c_i are given in the new operators b_p as

$$c_j = \frac{1}{\sqrt{L}} \sum_{p=-L/2}^{L/2} \exp(-ij\phi_p) b_p, \tag{3.8}$$

where the summation runs over all possible momenta. Applying the Fourier transform (3.8) to the Hamiltonian (3.7) results in

$$H = \frac{L}{2} - (\lambda + 1) b_0^\dagger b_0 - \sum_{p=1}^{L/2} (\lambda \cos\phi_p + 1)(b_p^\dagger b_p + b_{-p}^\dagger b_{-p}) - i\gamma\lambda \sin\phi_p (b_p^\dagger b_{-p}^\dagger + b_p b_{-p}). \tag{3.9}$$

The Hilbert space decomposes into non-interacting subspaces. The space of zero momentum is two dimensional and already diagonal, whereas the other subspaces are four dimensional and non-diagonal.

In order to diagonalise the remaining subspaces in the Hamiltonian (3.9) let $\alpha_p = (\lambda \cos\phi_p + 1)$, $\beta_p = \lambda\gamma \sin\phi_p$ and use then the canonically transformed operators

$$\eta_k = \tilde{\alpha}_k b_{-k} - i\tilde{\beta}_k b_k^\dagger, \tag{3.10}$$

with

$$\begin{aligned} \tilde{\alpha}_k &= \frac{\Lambda_k - \alpha_k}{\sqrt{2\left(\Lambda_k^2 - \Lambda_k \alpha_k\right)}}, \\ \tilde{\beta}_k &= \frac{\beta_k}{\sqrt{2\left(\Lambda_k^2 - \Lambda_k \alpha_k\right)}} \text{ and} \\ \Lambda_k &= \sqrt{\alpha_k^2 + \beta_k^2}. \end{aligned} \tag{3.11}$$

In the new creation and annihilation operators $\eta_k^\dagger$ and η_k our Hamiltonian reads

$$H = \sum_{k=-L/2}^{L/2} \Lambda_k \eta_k^\dagger \eta_k - \frac{1}{2} \sum_k \Lambda_k. \tag{3.12}$$

Clearly, the ground state of the system is given by the Fock vacuum in the fermionic basis. Although the vacuum state is separable in the fermionic basis this does not have to hold for the computational basis, since the Jordan-Wigner transformation is a global unitary transformation. To draw conclusions about the entanglement properties of the ground state one has to study the ground state in the computational basis.

3.1.2 Expectation values of Pauli operators

In order to calculate the three-particle [98] and four-particle reduced density matrices of the ground state $|0\rangle$ of our system one has to trace out all particles but three and four, respectively. Alternatively one can choose a local operator basis on each system, which is not to be traced out and calculate the one-point up to four-point correlators to recover the reduced density matrix. For the reduced states on the three particles i, j and k and the four particles i, j, k and l this would yield

$$\varrho_{ijk} = \frac{1}{8} \sum_{m,n,o} \langle \sigma_i^m \sigma_j^n \sigma_k^o \rangle_{|0\rangle} \sigma_i^m \sigma_j^n \sigma_k^o \text{ and} \tag{3.13}$$

$$\varrho_{ijkl} = \frac{1}{16} \sum_{m,n,o,p} \langle \sigma_i^m \sigma_j^n \sigma_k^o \sigma_l^p \rangle_{|0\rangle} \sigma_i^m \sigma_j^n \sigma_k^o \sigma_l^p, \tag{3.14}$$

where each of the summations m, n, o and p run over $\{x, y, z, 0\}$. In order to calculate the expectation values $\langle \sigma_{i_1}^{\alpha_1} \dots \sigma_{i_k}^{\alpha_k} \rangle_{|0\rangle}$ one needs to express the Pauli operators in terms of the fermionic operators $\eta_k^\dagger$ and η_k. It is

$$\begin{aligned}
\sigma_l^x &= (c_l + c_l^\dagger) \prod_{i=1}^{l-1} (c_i^\dagger + c_i)(c_i^\dagger - c_i), \\
\sigma_l^z &= -(c_l^\dagger + c_l)(c_l^\dagger - c_l), \\
\sigma_l^y &= -i(c_l^\dagger - c_l) \prod_{i=1}^{l-1} (c_i^\dagger + c_i)(c_i^\dagger - c_i)
\end{aligned} \tag{3.15}$$

and thus the Pauli operators are products of

$$A_l = c_l + c_l^\dagger = \frac{1}{\sqrt{L}} \sum_q \left(\eta_{-q}^\dagger + \eta_q \right) (\alpha_q + i\beta_q) e^{i\phi_q l}, \tag{3.16}$$

$$B_l = c_l - c_l^\dagger = \frac{1}{\sqrt{L}} \sum_q \left(-\eta_{-q}^\dagger + \eta_q \right) (\alpha_q - i\beta_q) e^{i\phi_q l}, \tag{3.17}$$

which are linear in the creation and annihilation operators $\eta_i^\dagger$ and η_i, respectively. The expectation values of arbitrary tensor products of

$$\begin{aligned}
\sigma_l^x &= A_l \prod_{i=1}^{l-1} A_i B_i, \\
\sigma_l^z &= -A_i B_i, \\
\sigma_l^y &= -i B_l \prod_{i=1}^{l-1} A_i B_i,
\end{aligned} \tag{3.18}$$

are then monomials in the fermionic creation and annihilation operators. These can be evaluated using the Wick theorem, which states the equality

$$\langle \mathcal{O}_1 \cdots \mathcal{O}_n \rangle_{|0\rangle} = \langle \mathcal{O}_1 \mathcal{O}_2 \rangle_{|0\rangle} \langle \mathcal{O}_3 \cdots \mathcal{O}_n \rangle_{|0\rangle} - \langle \mathcal{O}_1 \mathcal{O}_3 \rangle_{|0\rangle} \langle \mathcal{O}_2 \mathcal{O}_4 \cdots \mathcal{O}_n \rangle_{|0\rangle} + \langle \mathcal{O}_1 \mathcal{O}_4 \rangle_{|0\rangle} \cdots , \tag{3.19}$$

where $\mathcal{O}_i$ can be any operator A_j or B_k. Successive application of the theorem reduces the expectation values in (3.13) and its four-particle counterpart (3.14) to products of the two-point expectation values $\langle A_l A_k \rangle_{|0\rangle}$, $\langle A_l B_k \rangle_{|0\rangle}$ and $\langle B_l B_k \rangle_{|0\rangle}$. Using Eqs. (3.16) and (3.17) and setting $r = k - l$ one can calculate these to be

$$\langle A_l A_k \rangle_{|0\rangle} = \delta_{lk} \tag{3.20}$$

$$\langle A_l B_k \rangle_{|0\rangle} = \frac{2}{\pi} \int_0^\pi d\phi (\cos \phi r \, (1 + \lambda \cos \phi) - \gamma \lambda \sin \phi \sin \phi r) \frac{1}{\Lambda_\phi} \tag{3.21}$$

$$\langle B_l B_k \rangle_{|0\rangle} = -\delta_{lk} \tag{3.22}$$

in the thermodynamic limit. For finite L Eq. (3.21) is given by

$$\langle A_l B_k \rangle_{|0\rangle} = \frac{1}{L} \sum_q \frac{1}{\Lambda_q} (\cos r\phi_q (1 + \lambda \cos \phi_q) - \gamma \lambda \sin \phi_q \sin r\phi_q). \tag{3.23}$$

To evaluate $\langle \sigma_i^m \sigma_j^n \sigma_k^o \rangle_{|0\rangle}$ and $\langle \sigma_i^m \sigma_j^n \sigma_k^o \sigma_l^p \rangle_{|0\rangle}$ we express the Pauli operators in terms of A_j or B_k. Then proceed with the general scheme by Ref. [99]. Reorder the operators with respect to the fermionic commutation relations, such that all A_l are in front of the B_l and both are in ascending order. After this step the expressions look like

$$\pm \langle A_{i_1} ... A_{i_k} B_{j_1} ... B_{j_k} \rangle_{|0\rangle}. \tag{3.24}$$

Applying the Wick theorem iteratively one obtains the following Pfaffian

$$\pm pf\begin{bmatrix} 0 & \langle A_{i_1}A_{i_2}\rangle_{|0\rangle} & \dots & \langle A_{i_1}A_{i_k}\rangle_{|0\rangle} & \langle A_{i_1}B_{j_1}\rangle_{|0\rangle} & \dots & \langle A_{i_1}B_{j_k}\rangle_{|0\rangle} \\ -\langle A_{i_1}A_{i_2}\rangle_{|0\rangle} & 0 & & \langle A_{i_2}A_{i_k}\rangle_{|0\rangle} & \langle A_{i_2}B_{j_1}\rangle_{|0\rangle} & & \langle A_{i_2}B_{j_k}\rangle_{|0\rangle} \\ & & & \vdots & \vdots & & \vdots \\ & & \ddots & \langle A_{i_{k-1}}A_{i_k}\rangle_{|0\rangle} & \langle A_{i_{k-1}}B_{j_1}\rangle_{|0\rangle} & & \langle A_{i_{k-1}}B_{j_k}\rangle_{|0\rangle} \\ & & & 0 & \langle A_{i_k}B_{j_1}\rangle_{|0\rangle} & \dots & \langle A_{i_k}B_{j_k}\rangle_{|0\rangle} \\ & & & & 0 & \dots & \langle B_{j_1}B_{j_k}\rangle_{|0\rangle} \\ & & & & & & \vdots \\ & & & & & \ddots & \langle B_{j_{k-1}}B_{j_k}\rangle_{|0\rangle} \\ & & & & & & 0 \end{bmatrix}. \tag{3.25}$$

Note that the upper left and the lower right $k \times k$ block of this Pfaffian are zero, since we have $i_1 < i_2 < \cdots < i_k$ and $j_1 < j_2 < \cdots < j_k$ together with (3.20) and (3.22). Such a Pfaffian can be expressed as

$$pf\begin{bmatrix} 0 & M \\ -M^T & 0 \end{bmatrix} = (-1)^{\frac{k(k-1)}{2}} \det M \tag{3.26}$$

and therefore we have

$$\langle A_{i_1}...A_{i_k}B_{j_1}...B_{j_k}\rangle_{|0\rangle} = (-1)^{\frac{k(k-1)}{2}} \begin{vmatrix} G_{j_1-i_1} & \dots & G_{j_k-i_1} \\ \vdots & & \vdots \\ G_{j_1-i_k} & \dots & G_{j_k-i_k} \end{vmatrix}. \tag{3.27}$$

where $G_{k-l} = \langle A_l B_k\rangle_{|0\rangle}$.

A list of all non-zero expectation values of the four-site Pauli operators $\langle \sigma_i^m \sigma_j^n \sigma_k^o \sigma_l^p\rangle_{|0\rangle}$ can be found in Appendix B.

3.2 Genuine multiparticle entanglement in reduced states

Now that we have the three and four particle reduced density matrices of the ground state of the transverse XY-model available, we can investigate these with respect to the question whether these states are genuine multiparticle entangled. That is especially the question, how the spatial distribution of the three, respectively particles within the chain affects the amount of genuine multiparticle entanglement they share. To measure the genuine multiparticle entanglement we use the genuine multiparticle negativity [15] given by Eq. (1.30) as entanglement monotone. Furthermore, we apply a separability algorithm to three-particle states not detected by our entanglement monotone to prove their biseparability. That way we obtain a complete picture of the distribution of genuine multiparticle entanglement in the chain.

In the following, we focus on the transverse Ising model ($\gamma = 1$), but our approach can be straightforwardly extended to the XY-model.

3.2.1 Separability verification

In our work we supplement the genuine multiparticle negativity [15] with a separability algorithm [56, 57]. The idea of the algorithm is to decompose ϱ into two biseparable parts

$$\varrho = (1-p)\varrho_a + p\varrho_b, \tag{3.28}$$

such that ϱ_b is a statistical mixture of pure separable states and ϱ_a is within the ball of separable states around the completely mixed state [100]. The algorithm is performed iteratively such that the purity of ϱ_a is decreased with each successful step until ϱ_a lies within the set of separable states proving the separability of ϱ.

Given an input state ϱ, we define $\varrho = \varrho_0$ and iterate the following procedure:

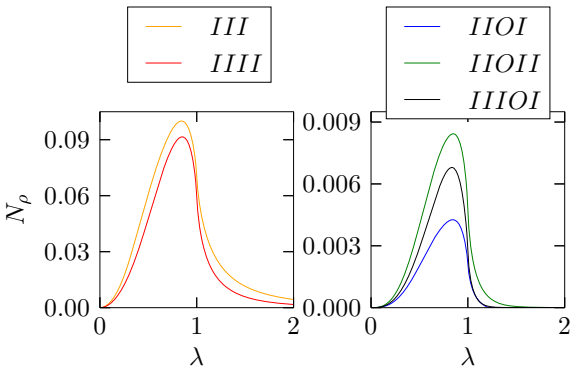


Figure 3.2: The genuine negativity N_ρ (for $L \to \infty$) as a function of λ for three- and four-particle reduced states in different spatial arrangements. On the left the values for three (III) or four (IIII) consecutive particles are shown. On the right hand side, spatial arrangements for three and four particles, which contain a single vacancy (denoted by 0) are shown. Here, II0I denotes a three-particle reduced state in the qubits $\{i, i+1, i+3\}$ and II0II denotes a four-particle reduced state in the qubits $\{i, i+1, i+3, i+4\}$ etc. (see text for further details).

- Find a pure biseparable state $|\psi_k\rangle$, which has large overlap with $\sqrt{\varrho_k}$. It will add to the part ϱ_b in the decomposition (3.28). We take random states and choose the one with the largest overlap.
- Choose $0 \leq \varepsilon_k \leq 1$, such that $\varrho_{k+1} = \frac{1}{1-\varepsilon_k}(\varrho_k - \varepsilon_k |\psi_k\rangle\langle\psi_k|)$ is positive semidefinite. This ensures that ϱ is indeed a convex combination of all $|\psi_k\rangle\langle\psi_k|$ and ϱ_{k+1}. Note that the algorithm seems to be more reliable if ε_k is significantly smaller than the smallest eigenvalue of ϱ.
- Check whether $tr({\varrho_{k+1}}^2) < \frac{1}{7}$ and thus if ϱ_{k+1} is separable for some bipartition [100].
- If this is the case, than the algorithm finishes and ϱ is separable, else continue until some maximal iteration number is reached.

In practice the algorithm performs good on all full rank states if, however, the state has eigenvalues close to zero it fails to detect separable states as such. This is due to the fact that ρ_0 is initially close to the border of separable states and hence during iterations ρ_k might leave the set of separable states. In such a case the algorithm cannot bring back ρ_k into the separable ball around the completely mixed state and the algorithm does not detect the state as separable.

Using local filtering one can significantly improve the algorithms ability to detect separable states with small eigenvalues. The idea is to apply invertible local matrices to the state $\varrho \mapsto \mathcal{F}_1 \otimes \mathcal{F}_2 \otimes \mathcal{F}_3 \varrho \mathcal{F}_1^\dagger \otimes \mathcal{F}_2^\dagger \otimes \mathcal{F}_3^\dagger$, such that the smallest eigenvalue of the normalised new state increases. This new states is then more likely to be detected by the algorithm if it is separable. Since both states are connected by invertible local operations the separability of one state implies the separability of the other. Using this modification we can on the one hand decrease the number of iterations the algorithm needs to show the separability of a separable state. On the other hand we are able to detect separable states, which are not detected by the unmodified algorithm.

3.2.2 Entanglement in three-qubit and four-qubit states

Let us start with the three-particle marginals of the ground state of the Ising model. Consider the particles i, j and k in the systems ground state. For the Ising model we can always set i to be zero, since the XY-Hamiltonian is translationally invariant and has periodic boundary conditions. The spatial arrangement of the particles is then completely described by $(\alpha, \beta) = (j-i, k-j)$, where (α, β) and (β, α) lead to the same reduced states due to mirror symmetry and hence one can choose $\alpha \leq \beta$ without loss of generality.

As a first task, we determined explicitly the reduced three-qubit states using the methods of Ref. [94–97], detailed expressions are given in Section 3.1.2. Evaluating the genuine multiparticle negativity for different spatial constellations in the thermodynamic limit $L \to \infty$ we find that for $(1,1)$ and $(1,2)$ the reduced marginals of the ground state are genuine multiparticle entangled in the vicinity of $\lambda = 1$ (see Fig. 3.2) and

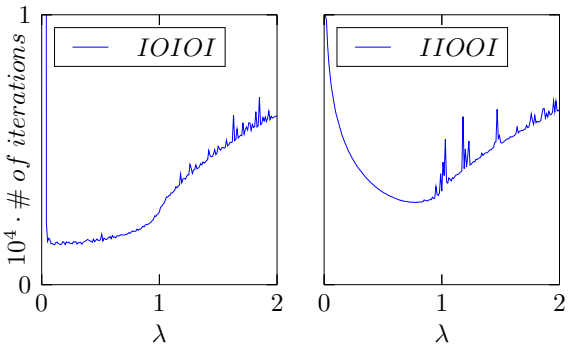


Figure 3.3: The number of iteration steps of the separability algorithm needed to show separability is shown with respect to the coupling parameter λ for the particle constellations $(1,3)$ (right) as well as for $(2,2)$ (left). For most cases the algorithm converges in less than 10^4 steps proving the separability of the respective states. For $\lambda = 0$ we know that the reduced state is pure and separable. For $\lambda \to 0$ the underlying state has small eigenvalues. Here the algorithm does not prove separability, even though these states are separable as well.

no entanglement is found if one separates the particles further. As the criterion of PPT mixtures is in general only a sufficient criterion for entanglement, the question arises whether for the separated configurations the reduced states are indeed biseparable. Concerning this point, it it first worth mentioning that so far no example of a genuinely entangled three-qubit state, which cannot be detected by the PPT mixture approach is known. In our case, we can show even explicitly, using the algorithm for proving separability from Ref. [56, 57] that the states are separable if the qubits are separated further (see Fig. 3.3). Using these novel results we also conclude that the genuine three-particle entanglement in the ground state stays short ranged and falls off to zero. It answers a discussion recently raised in Ref. [48]:In this reference, lower bounds on the entanglement were computed and no entanglement in the configuration $(1,2)$ was found, so it remained open, whether this or other configurations were separable. This makes us confident that the genuine multiparticle negativity is a well-suited tool for our analysis.

As in the three-particle case, the density matrices of four particles $i<j<k<l$ depend on the spacing between the particles $\alpha = j-i$, $\beta = k-j$ and $\delta = l-k$ and the coupling parameter λ only. Here we may choose $\alpha \leq \delta$ due to symmetry.

We find that there are three spatial arrangements with non-zero genuine negativity. These are the tightest packed constellations $(1,1,1)$ and the constellations $(1,1,2)$ and $(1,2,1)$. Further separated constellations yield a zero genuine negativity. We observe that with increasing separation the four-particle genuine negativity decreases with increasing separation. This is similar to the three particle case. On the other hand, the genuine negativity for the four-particle arrangements with one intermediate particle $(1,1,2)$ and $(1,2,1)$ is much larger than the comparable three-particle case $(1,2)$ (see Fig. 3.2). Taking into account that for the tightest conformations $(1,1)$ and $(1,1,1)$, respectively, the values are quite close to each other, the four-particle genuine multiparticle entanglement seems to be more uniformly distributed throughout the system.

3.3 Finite-size scaling

If one instead of the genuine multiparticle negativity plots its first derivative with respect to the coupling parameter λ (see Fig. 3.4), one observes a divergence, which indicates that the system undergoes a phase transition. We study it in more detail using the finite size scaling analysis [101, 102]. This analysis is based on the idea that close to the phase transition at λ_c the behaviour of a diverging quantity $P^{(L)}(\lambda)$ for finite system sizes L is governed by the system size L and a rescaled variable L/ξ only, with ξ being the correlation length. In the case of a logarithmic singularity of $P^{(L)}(\lambda)$, the finite-size scaling ansatz asserts the existence of a function Q, such that for finite L and λ close to the critical value [102]

$$P^{(L)}(\lambda) - P^{(L)}(\lambda_0) \sim Q(L^{\frac{1}{\nu}}|\lambda - \lambda_c|) - Q(L^{\frac{1}{\nu}}|\lambda_0 - \lambda_c|). \tag{3.29}$$

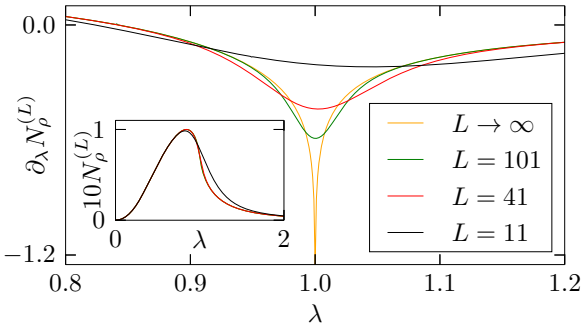


Figure 3.4: In the thermodynamic limit the first derivative of the genuine negativity with respect to λ diverges at the quantum phase transition. For finite chain length $L = 11$ (dotted line), $L = 41$ (dashed dotted line) and $L = 101$ in (dashed line) there is a minimum in the vicinity of the critical point $\lambda_c = 1$, which gets smaller for increasing L, diverging in the limit $L \to \infty$. The subplot shows the genuine negativity depending on λ for different chain lengths in a larger region around the critical point.

Here ν is the critical exponent, which governs the divergence of the correlation length $\xi \sim |\lambda - \lambda_c|^{-\nu}$ close to the critical value as $L \to \infty$. For the ansatz to consistently recover

$$P^{(\infty)}(\lambda) \sim C_\infty \ln|\lambda - \lambda_c| \text{ as } \lambda \to \lambda_c \tag{3.30}$$

in the thermodynamic limit, one sets $Q(z) \sim C_\infty \ln z$ for $z \to \infty$. Provided that $Q(z) = const.$ as $z \to 0$

$$P^{(L)}(\lambda_c(L)) \sim -\frac{C_\infty}{\nu} \ln L + const. \tag{3.31}$$

the minimum in $P^{(L)}$ at the pseudo-critical value diverges with the system size L, such that the Eqs. (3.30) and (3.31) allow to determine the critical exponent ν.

3.3.1 Scaling for three and four particles

For the two-particle entanglement, it was already shown that finite-size scaling holds [30]. In the multiparticle case, the quantity of interest is the derivative of the multiparticle negativity. So we vary the particle number L, keeping the arrangement $(1,1)$ fixed, and study the three-particle genuine negativity $N_\rho^{(L)}$ together with its first derivative $\partial_\lambda N_\rho^{(L)}$ (see Fig. 3.4). One observes a logarithmic divergence of $\partial_\lambda N_\rho^{(\infty)}$ at $\lambda_c = 1$. This is where the quantum phase transition occurs. There is distinct minimum in $\partial\lambda N_\rho^{(L)}$ for finite system sizes L at $\lambda_c(L)$, which we take to be the pseudo-critical value. We find that it approaches the critical value like $\lambda_c(L) - \lambda_c \sim L^{-\kappa}$ with a shift exponent $\kappa = 2.19$ [1].

Fitting the expected behaviour to our data as done in Fig. 3.5, one recovers the critical exponent, which is known to be $\nu = 1$, since

$$\partial_\lambda N_\varrho^{(\infty)} = 0.170 \ln(|\lambda - \lambda_c|) + 0.267 \tag{3.32}$$

$$\partial_\lambda N_\varrho^{(L)}(\lambda_c(L)) = -0.170 \ln L + 0.191\ , \tag{3.33}$$

for sufficiently big L. We performed the same analysis on the genuine negativity for the arrangements $(1,2)$ and observed similar qualitative results. Note that the numerical accuracy of the fitting procedure is higher than the displayed accuracy, a more detailed discussion is given in the next parapraph.

The finite-size scaling analysis shows that the behaviour of the genuine multiparticle entanglement close to the critical point is governed by the quantum phase transition. A divergent behaviour alone may be expected from the results of Ref. [33], as the reduced two-particle density matrix of the ground state itself is

[1] In many systems the shift exponent κ equals the inverse of critical exponent ν of the diverging correlation length [102]. In our case, however, this turns not out to be the case.

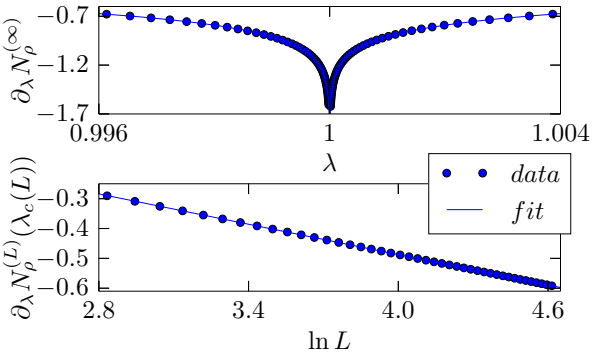


Figure 3.5: Evaluation of the minima for different chain lengths and the divergence in the thermodynamic limit (circles) of the first derivative of the genuine negativity for three consecutive particles shows the behaviour expected (lines) from the scaling ansatz for a logarithmic divergence. In the upper graph the divergence of the genuine negativity is plotted with respect to the coupling parameter λ close to the critical point $\lambda_c = 1$. The lower plot shows that minimum at the pseudo-critical value $\lambda_c(L)$ scales linear with the logarithm of the chain length L.

non-analytical [2]. Indeed many investigations of lower bounds on genuine multiparticle entanglement show this behaviour [43–49]. On the flip side none of these investigations allowed one to draw conclusions with respect to the critical attributes of the system. The finite-size scaling, however, shows that multiparticle entanglement faithfully represents important properties of the spin system at the critical point, moreover, it may be used for the extrapolation of critical exponents from finite-size numerical simulations.

The finite-size scaling analysis for four consecutive particles $(1, 1, 1)$ yields results similar to the three-particle case but is more subjected to numerical errors due to error propagation. For separations $(1, 2, 1)$ and $(1, 1, 2)$ the first derivative of the genuine negativity shows a qualitatively similar scaling behaviour as in the case where all four particles are in succession.

The scaling analysis in the case of four closely packed particles $(1, 1, 1)$ is along the line of the three-particle case. It yields similar results as in the three-particle case (see Fig. 3.6)

$$\partial_\lambda N_\varrho^{(\infty)} = 0.20 \ln(|\lambda - \lambda_c|) + 0.36 \tag{3.34}$$

$$\partial_\lambda L_\varrho^{(L)}(\lambda_c(L)) = -0.20 \ln L + 0.27. \tag{3.35}$$

The negative quotient of the logarithmic prefactors again give way to the expected critical exponent of the diverging correlation length $\nu = 1$. Due to numerical precision, however, the determination of the position and value of the minima was more subtle than in the three particle case. Specifically, a faithful determination of the position of the minima was impossible for $L > 33$ due to numerical inaccuracies, which we discuss in the next section in this Appendix. The determination of the absolute value of the minima, however, suffered less from these problems and made it possible to confirm the expected scaling.

3.3.2 Discussion of numerical precision

For the derivation of our results we make use of the genuine negativity [15], which is implemented via semidefinite programming. It provides an upper bound of how far the numerical value obtained and the global optimum are apart. One can take this bound as a conservative error for the genuine negativity. In our calculations the absolute error of the genuine negativity is of the order of 10^{-14}. This error is negligible if one considers the genuine negativity itself. For the finite-size scaling analysis, however, this error has to be taken into account. Recall that in the scaling analysis one has to extract several minima of the first

[2] But note that the argument of Ref. [33] in the strict sense concerns only entanglement in k-particle marginals of a ground state of an k-body Hamiltonian.

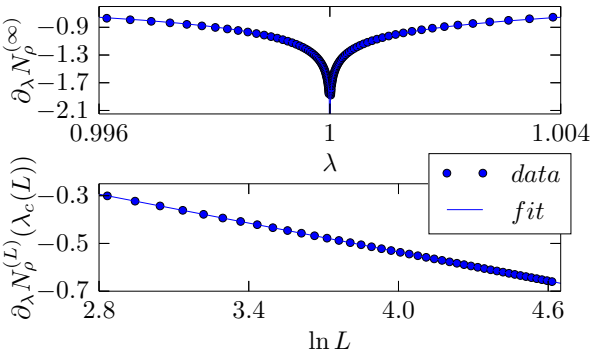


Figure 3.6: Evaluation of the minima for different chain lengths and the divergence in the thermodynamic limit (circles) of the first derivative of the genuine negativity for four consecutive particles shows the scaling behaviour expected (lines) from the scaling ansatz for a logarithmic divergence. On top the genuine negativity is plotted with respect to λ diverging logarithmically at the critical point. The mid plot confirms the ansatz made with respect the L-dependence of position of the minima and the lower plot shows, the predicted behaviour of the value of the minima with respect to different chain length.

derivative of the genuine negativity. The first derivative is approximated with the central finite difference method

$$f' \approx \frac{-f(x+2h) + 8f(x+h) - 8f(x-h) + f(x-2h)}{12h} \tag{3.36}$$

and the forward and backward finite differences close to the divergence

$$\begin{aligned} f' &\approx \frac{-25f(x) + 48f(x+h) - 36f(x+2h) + 16f(x+3h) - 3f(x+4h)}{12h} \\ f' &\approx \frac{+25f(x) - 48f(x-h) + 36f(x-2h) - 16f(x-3h) + 3f(x-4h)}{12h}. \end{aligned} \tag{3.37}$$

We found $h = 10^{-7}$ to give us optimal results. The absolute error of the genuine negativity, results in an absolute leading error in the first derivative. It is of the order of 10^{-7} and hence the overall precision in estimating the position and absolute value of the minimum in the first derivative of the genuine negativity is limited. This directly influences how well one can perform the finite-size scaling analysis. In the case of three $(1,1)$ and four $(1,1,1)$ consecutive particles one can find the position of these minima with sufficient high precision up to $L = 33$ particles and their values up to $L = 100$ particles, which is sufficient for the finite-size scaling analysis. For a larger number of particles, however, the errors of the positions and values of the minima caused growing errors in the coefficients of the fitting functions. This finally makes the extraction of the critical exponent ν impossible. Hence, we omitted a quantitative finite-size scaling analysis in these cases.

3.4 Conclusions

Using the Ising model in a transverse magnetic field, we investigated the connection between genuine multiparticle entanglement and quantum phase transitions. We identified the configurations of three and four particles where entanglement is present and showed that the derivative of the genuine multiparticle negativity diverges logarithmically at the critical points. We further confirmed that this quantity obeys a finite-size scaling behaviour close to the quantum phase transition.

Besides its fundamental interest, there are several consequences and applications of our work. First, as our method allows the direct study of multiparticle entanglement, it can be used to complete existing indirect results on multiparticle entanglement. To give an example, in Ref. [34] the existence of multiparticle entanglement in the one-dimensional XYZ-model was concluded indirectly from monogamy relations, and

possible connections to phase transitions were found. Our approach allows one to verify these results in a direct manner. Similarly in Ref. [98] the two-partite negativities for different splittings of three particles in the XY-model were studied, our methods can now decide whether these bipartite quantities are connected to genuine multiparticle entanglement. Further, in Ref. [49] certain three-particle reduced states could neither be detected as genuine multiparticle entangled nor shown to be biseparable. We expect that an application of our method would solve this issue. Second, our results demonstrate the usefulness of the genuine multiparticle negativity to study many-body systems. This makes it applicable to further systems, such as dynamical phase transitions [103], quenching dynamics, or the study of symmetry breaking [49].

Chapter 4

Generalised stabiliser formalism

It is possible to commit no mistakes and still lose. That is not a weakness; that is life.
David Kemper

In this chapter a generalisation to the stabiliser formalism defined in Section 1.4.1 will be discussed. Therefore, stabiliser groups and subgroups thereof will be considered as symmetry groups and stabilised states, which are invariant under the action of these symmetry groups shall be introduced. Finally, a method will be provided to identify and classify these symmetry groups with respect to LC-equivalence.

The chapter is organised as follows: First, we introduce the notion of stabilised states and discuss relations to the stabiliser formalism. Second, we provide an efficient method to generate subgroups of a stabiliser group. Finally, we classify the symmetry groups with respect to LC-equivalence and introduce a method to obtain all classes.

4.1 Stabilised states

Recall from Paragraph 1.4.1 that the main idea of the stabiliser formalism was to use a maximal commuting subgroup S of the n-qubit Pauli group

$$G_n = \{\alpha\sigma_{j_1} \otimes \sigma_{j_2} \otimes \cdots \otimes \sigma_{j_n} | j_k \in \{0,1,2,3\}, \alpha \in \{1,-1,i,-i\}\}\,, \tag{4.1}$$

to define the corresponding stabiliser state $|\psi_S\rangle$ as common eigenstate to eigenvalue $+1$ for all elements of the subgroup.

Contrary to an unique eigenstate stabilised states define a family of mixed states: the stabilised states, which share S as a symmetry group. ϱ_S is stabilised with respect to S, iff

$$s\varrho_S s\dagger = \varrho_S \tag{4.2}$$

for all $s \in S$.

For a given stabiliser group S let $\{g_1, g_2, \ldots, g_n\}$ be a set of generators. Corresponding to this set there are common eigenstates

$$g_i|a_1a_2\ldots a_n\rangle = a_i|a_1a_2\ldots a_n\rangle, \quad a_i \in \{+,-\}\,, \tag{4.3}$$

which form the stabiliser basis in the n-qubit Hilbert space. In this case the following holds true:

Lemma 4.1. *ϱ_S is diagonal in the basis $\{|a_1a_2\ldots a_n\rangle\}$.*

Proof. Each operator can be expressed with respect to the stabiliser basis as

$$\varrho_S = \sum_{a_1,\ldots,a_n,b_1\ldots,b_n \in \{\pm\}} F_{a_1\ldots a_n,b_1\ldots b_n}|a_1\ldots a_n\rangle\langle b_1\ldots b_n|, \tag{4.4}$$

where $F_{a_1\ldots a_n,b_1\ldots b_n} = \langle a_1\ldots a_n|\varrho_S b_1\ldots b_n\rangle$ are the coefficients of ϱ_S in the given basis. Due to the symmetry $g_i\varrho_S g_i\dagger = \varrho_S$ one has

$$a_ib_iF_{a_1\ldots a_n,b_1\ldots b_n}|a_1\ldots a_n\rangle\langle b_1\ldots b_n| = F_{a_1\ldots a_n,b_1\ldots b_n}|a_1\ldots a_n\rangle\langle b_1\ldots b_n|. \tag{4.5}$$

This equation is satisfied if $a_i = b_i = \pm 1$ for all i or $F_{a_1 \dots a_n, b_1 \dots b_n} = 0$ else. That is equivalent to: all off-diagonal elements $F_{a_1 \dots a_n, b_1 \dots b_n}$ vanish. □

In this case stabilised states are simply diagonal in the basis of common eigenvectors of the stabiliser group. This also includes graph-diagonal states, which have been introduced in Section 2.2.1.

4.1.1 Stabilised states of stabiliser subgroups

If one sticks to the idea to use symmetries to define state families, then there is a quite natural extension to the formalism introduced above.

Consider subgroups of stabiliser groups, then the definition of stabilised states can be extended to these subgroups. Let S be a stabiliser group and T be some subgroup of S then ϱ_T is stabilised by T, iff

$$t \varrho_T t^\dagger = \varrho_T \tag{4.6}$$

for all $t \in T$. Henceforth, non-maximal commuting subgroups of the Pauli group will be referred to as stabiliser subgroups.

If we denote $\mathcal{S}_S$ the set of all state stabilised by S and by $\mathcal{S}_T$ the set of states stabilised by T, then $\mathcal{S}_S \subset \mathcal{S}_T$. Or in terms of symmetry one might say that the symmetry group S contains the symmetry group T. In other words: the symmetry group T puts fewer constraints onto the set of T symmetric states $\mathcal{S}_T$ than S.

Note that since $T \subset S \subset G_n$, T is an Abelian subgroup of the n-qubit Pauli group. Furthermore, one can still find a set of algebraic independent generators $\{g_1, \dots, g_k\}$ with $k < n$, such that the generators generate T. Since all the generators commute the subgroup has $|T| = 2^k$ elements.

Lemma 4.2. *For each subgroup T of S the set of generators $\{g_1, \dots, g_k\}$ of T can be extended to a generator set $\{g_1, \dots, g_k, g_{k+1}, \dots, g_n\}$ of S.*

Proof. Let T be a subgroup of the stabiliser group S and $\{g_1, \dots, g_k\}$ be a generator set of T.

One can find generators $\{g_{k+1}, \dots, g_n\}$ by the following itterative procedure.

1. Choose $g_{k+1} \in S/T$ arbitrary. Then the generators in $\{g_1, \dots, g_{k+1}\}$ generate a new subgroup T' of S with 2^{k+1} elements.
2. If $k + 1 = n$, then $T' \subset S$ and $|T'| = |S|$. That is $T' = S$ hence the generator set $\{g_1, \dots, g_{k+1}\}$ generates S. If $k + 1 < n$ then denote $T' \equiv T$ and proceed the iteration.

□

Let T be a subgroup of the stabiliser group S and ϱ_T be stabilised by T. In this case we cannot use the Projectors onto the stabiliser states of S to describe the state as done in Lemma 4.1. There is, however, another canonical way to describe the set of T-stabilised states as a multi-parameter family.

Therefore, consider as basis in operator space the n-fold tensor product $\mathfrak{B} = \mathfrak{P}^{\otimes n}$ of the single qubit operator basis $\mathfrak{P} = \{\sigma_i\}_{i=0}^3$. Then every state ϱ can be expressed in terms of this basis as

$$\varrho = \frac{1}{2^n} \sum_{i_1 \dots i_n} \mathrm{tr}(\varrho \sigma_{i_1} \otimes \cdots \otimes \sigma_{i_n}) \sigma_{i_1} \otimes \cdots \otimes \sigma_{i_n}. \tag{4.7}$$

Similar to the proof of Lemma 4.1 one can show that

$$\mathrm{tr}(\varrho b) = 0 \tag{4.8}$$

for all $b \in \mathfrak{B}/\mathfrak{B}_T$, where

$$\mathfrak{B}_T = \{b \in \mathfrak{B} |\, [b, t] = 0 \text{ for all } t \in T\}. \tag{4.9}$$

Corollary. *Let ϱ_T be stabilised by T, then ϱ_T can be written as*

$$\varrho_T = \frac{1}{2^n} \sum_{b \in \mathfrak{B}_T} p_b b, \tag{4.10}$$

where the p_b are real and $\mathfrak{B}_T = \{b \in \mathfrak{B} |\, [b, t] = 0$ for all $t \in T\}$.

Example 4.1. A prominent example of such a state family are the well known two-qubit X-states [104–106]. The density matrices of these states have non-vanishing entries only on the diagonal and anti-diagonal. That is the only non-vanishing coefficients in Eq. (4.10) appear in front of $\mathbb{1}\otimes\mathbb{1}$, $\mathbb{1}\otimes\sigma_z$, $\sigma_z\otimes\mathbb{1}$, $\sigma_z\otimes\sigma_z$, $\sigma_x\otimes\sigma_x$, $\sigma_x\otimes\sigma_y$, $\sigma_y\otimes\sigma_x$ and $\sigma_y\otimes\sigma_y$.

Now consider the two-qubit stabiliser subgroup T generated by the single generator $t_1=\sigma_z\otimes\sigma_z$. The states stabilised by T are exactly the two-qubit X-states, since the non-vanishing coefficients in Eq. 4.10 are given by

$$\mathfrak{B}_T=\{\mathbb{1}\otimes\mathbb{1},\mathbb{1}\otimes\sigma_z,\sigma_z\otimes\mathbb{1},\sigma_z\otimes\sigma_z,\sigma_x\otimes\sigma_x,\sigma_x\otimes\sigma_y,\sigma_y\otimes\sigma_x,\sigma_y\otimes\sigma_y\}\,. \tag{4.11}$$

Note that X-states first have been algebraically described using the set $\mathfrak{B}_T$ by Rau [107]. Moreover, he extended this description to generalised n-qubit X-states [108]. In the framework of stabilised states these generalised X-states are stabilised by the stabiliser subgroup generated by $\left\{g_i=Z^{(1)}Z^{(i+1)}\right\}_{i=1}^{n-1}$.

The novelty of our approach is that certain widely used state classes such as graph-diagonal states or X-states arise naturally as stabilised states, offering a single framework to characterise and study these states.

4.2 Efficient generation of subgroups

As a direct consequence of Lemma 4.2 every possible subgroup $T\subset S$ arises from a suitable generator set of S by removing some of its generators.

This procedure is quite inefficient, since many generator sets obtained this way will generate the same subgroup. Hence, a method to obtain generator sets, which uniquely correspond to a subgroup will be developed.

At first, let $\{g_1,\ldots,g_n\}$ be a fixed set of generators of S. Since the g_i generate S all other possible generator sets $\{\tilde{g}_1,\ldots,\tilde{g}_n\}$ can be expressed with respect to $\{g_i\}_{i=1}^n$ as

$$\begin{aligned}\tilde{g}_1&=g_1^{a_{11}}g_2^{a_{12}}\ldots g_n^{a_{1n}}\\ \tilde{g}_2&=g_1^{a_{21}}g_2^{a_{22}}\ldots g_n^{a_{2n}}\\ &\vdots\\ \tilde{g}_n&=g_1^{a_{n1}}g_2^{a_{n2}}\ldots g_n^{a_{nn}},\end{aligned} \tag{4.12}$$

where the matrix $A=(a_{ij})\in GF(2)^{n\times n}$ is a matrix over the Galois field of two elements. Note that the $\tilde{g}_i$ generate S, iff A is invertible. If on the other hand A is not invertible, then the generators $\{\tilde{g}_i\}_{i=1}^n$ are algebraic dependent and generates a subgroup T of S.

If one identifies the generator change with the corresponding matrices, then one can write any generator change as

$$\{g_i\}^n\overset{A}{\to}\{\tilde{g}_i\}^n\,, \tag{4.13}$$

where $\{g_i\}^n\equiv\{g_i\}_{i=1}^n$.

Using this notation a composition of multiple generator changes correspond to left-multiplication of the corresponding matrices. That is

$$\{g_i\}^n\overset{A}{\to}\{\tilde{g}_i\}^n\overset{B}{\to}\{g_i'\}^n=\{g_i\}^n\overset{BA}{\to}\{g_i'\}^n\,. \tag{4.14}$$

Before we proceed we review the notion of the row-reduced echelon form of a matrix [109]. To each matrix $A\in\mathbb{F}^{m\times n}$ over a finite field there corresponds a matrix in row-reduced echelon form (RREF). For each matrix there is only one matrix in RREF, which can be attained by a sequence of elementary matrix operations. These are:

1. Interchange of two rows.
2. Multiplication of a row by a nonzero scalar.
3. Addition of a scalar multiple of one row to another one.

Note that in the case of the field $GF(2)$ the second operation can be omitted, since the only non-zero scalar is unity. The RREF is then defined by the following properties:

1. Each non-zero row has 1 as first non-zero entry.
2. All other entries in the column of such a leading 1 are zero.
3. Any row consisting of zeros only occurs at the bottom of the matrix.
4. A leading 1 in a lower row must occur to the right of its above counterpart.

As an example

$$\begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{4.15}$$

is in RREF. Furthermore, a matrix A is invertible, iff its RREF is the identity matrix.

Let A be any matrix corresponding to a generator change $\{g_i\}^n \xrightarrow{A} \{\tilde{g}_i\}^n$. In this case the interchange of two rows i and j corresponds to an exchange of $\tilde{g}_i \leftrightarrow \tilde{g}_j$ in the new generator set. Moreover, an addition of a scalar multiple of row i to row j either leaves the generators as they are if the scalar was 0 or transforms $\tilde{g}_j \to \tilde{g}_i\tilde{g}_j$.

An important observation is:

Corollary. *Neither of the elementary operations applied to a matrix, which corresponds to a generator change, changes the set generated by the corresponding generators.*

We can now come back to the problem to find an efficient method to generate all possible subgroups of S. The original method to do so was the following:

1. Consider all possible generator sets of S. As we have shown that corresponds to the set of all possible invertible matrices $A \in GF(2)^{n\times n}$.
2. Consider all possible ways to remove a generator. That corresponds to to the set of all possible non-invertible diagonal matrices $B \in GF(2)^{n\times n}$.

Combining these two steps yield the set of all non-invertible matrices in $GF(2)^{n\times n}$. For each of these matrices there is a subgroup T of S generated by the corresponding generators.

Still there are many generator sets generating the same subgroup. To circumvent this ambiguity we use the fact that each of the corresponding matrices can be transformed into the RREF without changing the generated subgroup. As a direct consequence two generators generate the same subgroup, if the corresponding matrices have the same RREF. Furthermore, if two generators generate the same subgroup, then the corresponding generator matrices have the same RREF. In conclusion:

Corollary. *Each subgroup T of S is in one to one correspondence with a matrix $A \in GF(2)^{n\times n}$ in RREF. If $A = \mathbb{1}_n$, then $T = S$. For all other A in RREF the corresponding subgroup T is a proper subgroup of S.*

We can summarise our results in the following efficient method to find all possible subgroups of S and corresponding generator sets:

1. First, we fix a generator set $\{g_i\}^n$ of S.
2. Second, we write down the set $\mathfrak{R} = \{A \in GF(2)^{n\times n} | A \text{ is in RREF}\}$ of all possible row reduced echelon matrices.
3. Last, each $A \in \mathfrak{R}$ labels a subgroup T_A of S, which is generated by

$$\begin{aligned} t_1 &= g_1^{a_{11}} g_2^{a_{12}} \dots g_n^{a_{1n}} \\ &\vdots \\ t_k &= g_1^{a_{k1}} g_2^{a_{k2}} \dots g_n^{a_{kn}}, \end{aligned} \tag{4.16}$$

   where k is the number of non-zero rows in A.

Example 4.2. To illustrate the method consider the example of the 3-qubit star graph stabiliser, which we discussed in Section 1.4.1. There $n = 3$ and we fix the generators to be $g_1 = XZZ$, $g_2 = ZX\mathbb{1}$ and $g_3 = Z\mathbb{1}X$. Next, we list all possible 3×3 matrices in $GF(2)$, which are in RREF:

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

2. $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$

3. $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix}$

4. $\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$

5. $\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix}$

6. $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$

7. $\begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$

8. $\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$

9. $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$

10. $\begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$

11. $\begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$

12. $\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$

13. $\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$

14. $\begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$

15. $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$

According to Eq. (4.16) the generators of the corresponding subgroups read:

1. $t_1 = XZZ$, $t_2 = ZX\mathbb{1}$, $t_3 = Z\mathbb{1}X$

2. $t_1 = XZZ$, $t_2 = ZX\mathbb{1}$

3. $t_1 = XZZ$, $t_2 = \mathbb{1}XX$

4. $t_1 = YZY$, $t_2 = ZX\mathbb{1}$

5. $t_1 = YZY$, $t_2 = \mathbb{1}XX$

6. $t_1 = XZZ$, $t_2 = Z\mathbb{1}X$

7. $t_1 = YYZ$, $t_2 = Z\mathbb{1}X$

8. $t_1 = ZX\mathbb{1}$, $t_2 = Z\mathbb{1}X$

9. $t_1 = XZZ$

10. $t_1 = YZY$

11. $t_1 = YYZ$

12. $t_1 = XYY$

13. $t_1 = ZX\mathbb{1}$

14. $t_1 = \mathbb{1}XX$

15. $t_1 = Z\mathbb{1}X$

4.3 LC-equivalent subgroups

As already pointed out for stabiliser states in Section 1.4.2 many stabiliser states are equivalent with respect to a local Clifford (LC) operation, which can also be seen as a local basis change[1]. Furthermore, each stabiliser group is LC-equivalent to a group corresponding to a graph state [61].

[1] Note, however, that if two stabiliser groups are LU equivalent (local basis change), they are not necessarily LC equivalent [29].

Here the LC-classification of stabiliser groups using invariants as discussed in Ref. [27] is extended to stabiliser subgroups.

Let T and T' be two stabiliser subgroups. They are said to be LC-equivalent, iff there exists an element of the Local Clifford group $U \in C_1^{\bigotimes n}$, such that

$$UTU^\dagger = T'. \tag{4.17}$$

In that case the corresponding sets of stabilised states are also equivalent. That is

$$U^\dagger \mathcal{S}_T U = \mathcal{S}_{T'}. \tag{4.18}$$

To find all possible LC-equivalence classes within the stabiliser subgroups one starts by considering the set of LC-inequivalent graphs[2], since LC-equivalent stabiliser groups would yield LC-equivalent stabiliser subgroups.

The classification of graphs with respect to LC-equivalence has been achieved up to $n = 7$ [61] and $n = 8$ [110].

Let the number of qubits n be fixed, $\mathcal{G}_n$ be the set of all LC-inequivalent n-vertex graphs and consider the corresponding LC-inequivalent stabiliser groups S_i. From these one generates all possible subgroups $\mathcal{T}^{(k)} = \left\{T_i^{(k)}\right\}$ using the method introduced in the last paragraph. Here the superscript (k) denotes the number of algebraic independent generators generating the subgroup.

Let T and T' be two subgroups in $\mathcal{T}^{(k)}$, then there are four possibilities concerning the LC-equivalence of these subgroups:

1. Both subgroups originate from the same stabiliser group and are LC-inequivalent.
2. Both subgroups originate from the same stabiliser group and are LC-equivalent.
3. Both subgroups originate from different stabiliser groups and are LC-inequivalent.
4. Both subgroups originate from different stabiliser groups and are LC-equivalent.

In fact for the LC-classification it is not of relevance, if the subgroups originate from the same or different subgroups.

To classify the subgroups in $\mathcal{T}^{(k)}$ with respect to LC-equivalence we adapt Theorem 1 of Ref. [27]. In this theorem invariants are provided, which characterise LC-equivalence classes of stabiliser groups. These invariants are based on the notion of the support of group elements. Let $g = \pm \bigotimes_{i=1}^n \sigma_{\alpha_i}$, $\alpha_i \in \{0, \ldots, 3\}$ be any group element of the n-qubit Pauli group, then the support of this element is given by

$$supp(g) := \left\{ i \in \{1, \ldots, n\} | \sigma_{\alpha_i}^{(i)} = \mathbb{1} \right\}. \tag{4.19}$$

Lemma 4.3. *Let $T, T' \in \mathcal{T}^{(k)}$ be two stabiliser subgroups. Then T and T' are LC-equivalent, iff there are generator sets $\{g_1, \ldots, g_k\}$ for T and $\{g'_1, \ldots, g'_k\}$ for T', such that*

$$\mathrm{supp}(g_i) = \mathrm{supp}(g'_i) \quad \textit{and} \quad \mathrm{supp}(g_l g_m) = \mathrm{supp}(g'_l g'_m) \tag{4.20}$$

for all $i = 1, \ldots, k$, $1 \leq l < m \leq k$.

Proof. To prove this lemma we adopt the proof of Theorem 1 of Ref. [27].

Denote by $g^{(l)}$ the l-th tensor component of g. From Eqs. (4.19) and (4.20) it follows that

$$\begin{aligned} g_i^{(l)} = \mathbb{1} &\Leftrightarrow g_i'^{(l)} = \mathbb{1} \\ g_i^{(l)} = g_j^{(l)} &\Leftrightarrow g_i'^{(l)} = g_j'^{(l)}. \end{aligned} \tag{4.21}$$

This implies the existence of some permutation π_l of $\{x, y, z\}$, such that

$$\begin{aligned} g_i^{(l)} = \mathbb{1} &\Leftrightarrow g_i'^{(l)} = \mathbb{1} \\ g_i^{(l)} = \sigma_\alpha &\Leftrightarrow g_i'^{(l)} = \sigma_{\pi_l(\alpha)} \quad \text{for all } \alpha \in \{x, y, z\}. \end{aligned} \tag{4.22}$$

[2] By slight abuse of notation we use the phrase graph and sometimes refer to the corresponding stabiliser group.

Recall that the Clifford group C_1 is the group of all operations, which map σ_α to $\sigma_{\pi(\alpha)}$, where $\alpha \in \{x, y, z\}$, $u = \pm 1$ and π is some permutation of $\{x, y, z\}$.

Hence there is for each tensor component l a local Clifford operation mapping $g_i^{(l)}$ to $g_i'^{(l)}$ for all i. The tensor product of these one-qubit Clifford operations then map g_i to g_i' for all i. Furthermore, the invertability of these operation ensures the existence of a Clifford operation mapping g_i' to g_i for all i.

We have just shown that $\{g_1, \ldots, g_k\}$ and $\{g_1', \ldots, g_k'\}$ are LC-equivalent. The LC-equivalence of the generated groups directly follows. □

Example 4.3. As an example consider the two subgroups generated by

$$t_1 = XZZ \quad \text{and} \quad t_2 = ZX\mathbb{1} \tag{4.23}$$

and

$$t_1' = YZY \quad \text{and} \quad t_2' = ZX\mathbb{1} \tag{4.24}$$

respectively. For these we have

$$supp(t_1) = \emptyset,\ supp(t_2) = \{3\} \ \text{and} \ \ supp(t_1 t_2) = \emptyset \tag{4.25}$$

and

$$supp(t_1') = \emptyset,\ supp(t_2') = \{3\} \ \text{and} \ \ supp(t_1' t_2') = \emptyset \tag{4.26}$$

respectively. According to Lemma 4.3 these two generator sets are LC-equivalent and indeed the permutation $X \leftrightarrow Y$ on the first and $Z \leftrightarrow Y$ on the third qubit maps the generator set $\{t_1, t_2\}$ to $\{t_1', t_2'\}$.

4.3.1 LC-classification of subgroups

To provide an unambiguous LC-classification of all possible stabilising subgroups it is convenient to remove qubit-permutational equivalent subgroups[3]. That is, two stabilising subgroups T and T' are considered qubit-permutational equivalent if there exists a qubit-permutation Π, such that $T = \Pi(T') = \{\Pi(t) | t \in T'\}$.

Taking this into account one finds all possible LC-equivalent[4] stabilising subgroups on n qubits as follows:

1. Let $\mathcal{G}_n$ be the set of all LC inequivalent n-qubit graphs.
2. For a fixed $k < n$ generate all subgroups $\mathcal{T}^{(k)}$ with k generators using the method introduced in Section 4.2.
3. Within the set $\mathcal{T}^{(k)}$ find all qubit-permutational and LC-equivalent subgroups:
   (a) Consider all possible generator sets $\mathfrak{G}$ of $T_1^{(k)} \in \mathcal{T}^{(k)}$. These correspond to invertible $k \times k$ matrices over the finite field $GF(2)$ as showed in Section 4.2.
   (b) Add to $\mathfrak{G}$ all generator sets, which can be achieved by permutation of qubits.
   (c) Choose an arbitrary generator set $\{t_1', \ldots, t_k'\}$ of T'.
   (d) T and T' are LC-equivalent, iff for at least one generator set $\{t_1, \ldots, t_k\} \in \mathfrak{G}$

$$\text{supp}(t_i) = \text{supp}(t_i') \ \ \text{and} \ \ \text{supp}(t_l t_m) = \text{supp}(t_l' t_m'). \tag{4.27}$$

   (e) Remove T_1 and all T', which are LC-equivalent from $\mathcal{T}^{(k)}$ and start anew until $\{T_i\} = \emptyset$.

Example 4.4. Consider the case of $n = 3$ and $k = 2$. For $n = 3$ there is just one LC-equivalence class. The graph representing this group is the three-vertex linear graph and the corresponding stabiliser group is generated by $g_1 = XZ\mathbb{1}$, $g_2 = ZXZ$ and $g_3 = \mathbb{1}ZX$.

All possible subgroups in $\mathcal{T}^{(2)}$ are then given by:

1. $\{\mathbb{1}\mathbb{1}\mathbb{1}, XZ\mathbb{1}, YYZ, ZXZ\}$,
2. $\{\mathbb{1}\mathbb{1}\mathbb{1}, X\mathbb{1}X, YXY, ZXZ\}$,

[3] This is done since qubit-permutations corresponds merely to a relabelling of the qubits.

[4] By a slight abuse of notation LC-equivalent also refers to qubit-permutational equivalent subgroups.

3. $\{\mathbb{1}\mathbb{1}\mathbb{1}, XZ\mathbb{1}, YXY, ZYY\}$,
4. $\{\mathbb{1}\mathbb{1}\mathbb{1}, X\mathbb{1}X, YYZ, ZYY\}$,
5. $\{\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}ZX, YXY, YYZ\}$,
6. $\{\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}ZX, ZXZ, ZYY\}$,
7. $\{\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}ZX, X\mathbb{1}X, XZ\mathbb{1}\}$.

Performing the third step of the procedure above one finds two qubit-permutational and LC-inequivalent classes: the first contains the subgroups 1-6 and the second class contains solely the subgroup 7.

Using the method above gives a full characterisation of the qubit-permutational and LC-equivalent symmetry groups. In Appendix C we listed all LC-inequivalent subgroups for $n \leq 6$ and $k \leq 4$. Each class is thereby represented by one of its members.

Note that the symmetry group LC-equivalent to the symmetry group of the generalised X-states also appear in this list. For $n = 2$, $n = 3$, $n = 4$ and $n = 5$ the classes are represented by

$$\{\mathbb{1}\mathbb{1}, XZ\}\,, \tag{4.28}$$

$$\{\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}ZX, X\mathbb{1}X, XZ\mathbb{1}\}\,, \tag{4.29}$$

$$\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}XX, \mathbb{1}X\mathbb{1}X, \mathbb{1}XX\mathbb{1}, Z\mathbb{1}\mathbb{1}X, Z\mathbb{1}X\mathbb{1}, ZX\mathbb{1}\mathbb{1}, ZXXX\} \text{ and} \tag{4.30}$$

$$\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}\mathbb{1}XX, \mathbb{1}\mathbb{1}X\mathbb{1}X, \mathbb{1}\mathbb{1}XX\mathbb{1}, \mathbb{1}X\mathbb{1}\mathbb{1}X, \mathbb{1}X\mathbb{1}X\mathbb{1}, \mathbb{1}XX\mathbb{1}\mathbb{1}, \mathbb{1}XXXX,$$
$$Z\mathbb{1}\mathbb{1}\mathbb{1}X, Z\mathbb{1}\mathbb{1}X\mathbb{1}, Z\mathbb{1}X\mathbb{1}\mathbb{1}, Z\mathbb{1}XXX, ZX\mathbb{1}\mathbb{1}\mathbb{1}, ZX\mathbb{1}XX, ZXX\mathbb{1}X, ZXXX\mathbb{1}\} \tag{4.31}$$

respectively.

4.4 Conclusions

In summary we have generalised the stabiliser formalism bidirectionally. First, arbitrary Abelian subgroups of the n-qubit Pauli group have been considered as symmetry group instead of maximal Abelian ones. Second, the notion of stabilised states have been introduced. These include states such as those diagonal in a stabiliser basis [61] and generalised n-qubit X-states [108]. Furthermore, an efficient method to obtain symmetry groups from maximal Abelian subgroups of the n-qubit Pauli group was provided and the subgroups were classified with respect to equivalence under qubit-permutation and local Clifford operations.

Still there are many open questions, which might be worth studying in the future. Recall that within the graph state formalism stabiliser groups of two graphs are LC-equivalent if the corresponding graphs can be transformed into each other by local complementation [26, 61, 111]. Naturally, the question arises, if there is a way to relate graphs or operations thereof to stabiliser subgroups and whether this correspondence can be used to obtain the LC-equivalence classes. Furthermore, it would be interesting to extend the LC-classification to a LU-classification to completely remove the ambiguity of basis change.

It is worth to note that stabilised states are a useful theoretical laboratory. They provide multi-parameter state families, which are described by there symmetries. These symmetries restrict the state space, in which the stabilised states can live in. It would be interesting to investigate certain aspects such as genuine multiparticle entanglement on this restricted space, which are hard to grasp on the full state space. Finally, the question remains whether stabilised states can be used for quantum tasks beyond those, where graph states have already proven to be useful.

Chapter 5

Semidefinite programming

The aim of science is not to open the door to infinite wisdom, but to set some limit on infinite error.
Bertold Brecht

This chapter discusses a numerical and theoretical tool of convex optimisation theory for a problem known as semidefinite programming (SDP) [68]. In semidefinite programming problems one minimises a linear function subjected to positive semidefinite constraints: an optimisation, which appears quite naturally in quantum information processing.

To mention but a few examples semidefinite programming is used to obtain the maximum fidelity of a positive partial transpose distillation protocol [112]. In Ref. [113], semidefinite programming is applied to design operational criteria that distinguish entangled from separable quantum states. Another approach uses it to optimise over a certain set of entanglement witnesses [114] to create sufficient separability tests. In the framework of quantum state discrimination semidefinite programming is used to design optimal strategies and measurements to discriminate between non-orthogonal quantum states [115, 116]. Further, it was shown in Ref. [117] that finding a completely positive trace-preserving linear map that maximises the fidelity between the map itself and certain desired transformations can be cast into a SDP.

This chapter is organised as follows: First, we will review the basic definition of the primal and dual semidefinite optimisation problem and the main result of duality theory connecting both of them. Second, we show in detail how to implement the (renormalised) genuine multiparticle negativity with the semidefinite programming framework CVXOPT and provide some technical remarks on how semidefinite programming can be used in quantum mechanics. Finally, we discuss how one can exploit the symmetries of a problem to reduce the computational effort of a semidefinite problem and therefore the time a solver needs to find the optimal solution.

5.1 Semidefinite program

We start by recalling the notion of a semidefinite programming problem as introduced in Ref. [68]. Note that in the literature one often refers to a semidefinite programming problem as semidefinite program or semidefinite problem. Let $\vec{x} \in \mathbb{R}^m$ be the variable and $\vec{c} \in \mathbb{R}^m$ be the problem vector. In a semidefinite program (SDP) one wants to minimise the linear function $f(\vec{x}) = \vec{c}^T \vec{x}$ with respect to the constraint that $F(\vec{x}) = F_0 + \sum_i x_i F_i$ is positive semidefinite[1], where $F_i = F_i^T \in \mathbb{R}^{n \times n}$ are symmetric matrices. In a more compact notion one writes the semidefinite program or primal problem[2] as

$$\min_{\vec{x}} \vec{c}^T \vec{x} \qquad (5.1)$$
$$\text{s.t. } F(\vec{x}) = F_0 + \sum_{i=1}^{m} x_i F_i \geq 0.$$

[1] One also writes $F(\vec{x}) \geq 0$ for this condition.

[2] The phrase primal problem might be used if it is clear, which kind of convex optimisation problem is meant. In our case, primal problem refers to a semidefinite programming problem. It may also indicate that the corresponding dual problem is of importance.

This optimisation is called convex, since the feasible region $\{\vec{x}|F(\vec{x}) \geq 0\}$, the set of all feasible points, $\{\vec{x} \in \mathbb{R}^m|F(\vec{x}) \geq 0\}$, is convex and the function being minimised is linear and hence convex. That is for all feasible $\vec{x}$ and $\vec{y}$ and for all $0 \leq \lambda \leq 1$, the point $\lambda\vec{x} + (1-\lambda)\vec{y}$ is again feasible

$$F(\lambda\vec{x} + (1-\lambda)\vec{y}) = \lambda F(\vec{x}) + (1-\lambda)F(\vec{y}) \geq 0. \tag{5.2}$$

Using the primal problem, one derives the dual semidefinite program, also referred to as dual problem[3] by

$$\begin{aligned} &\max_Z \left[-\mathrm{tr}\left(F_0 Z\right)\right] \\ &\quad \text{s.t. } \mathrm{tr}\left(F_i Z\right) = c_i \quad \text{for all } i = 1, \ldots, m, \\ &\quad Z \geq 0, \end{aligned} \tag{5.3}$$

where $Z = Z^T \in \mathbb{R}^{n\times n}$ is subjected to m equality constraints $\mathrm{tr}(F_i Z) = c_i$ and the positive semidefiniteness condition $Z \geq 0$. As in the primal case, a point Z is called feasible if it satisfies all the constraints of the dual optimisation, i.e., $\mathrm{tr}\left(F_i Z\right) = c_i$ and $Z \geq 0$.

Note at this point that the dual problem is also a semidefinite program[4]. Moreover, the dual problem lower bounds the primal problem and conversely the primal problem upper bounds the dual one, i.e., $\vec{c}^T\vec{x} \geq -\mathrm{tr}(F_0 Z)$ for all feasible points $\vec{x}$ and Z. To see this consider the feasible points $\vec{x}$ and Z. Since the product of two positive symmetric matrices is also positive

$$\mathrm{tr}\left(ZF(\vec{x})\right) \geq 0. \tag{5.4}$$

Using $\mathrm{tr}\left(F_i Z\right) = c_i$ the left hand side of Eq. (5.4) is then given by the difference of the primal an dual objective

$$0 \leq \mathrm{tr}\left(ZF(\vec{x})\right) = \sum_{i=1}^{m} \mathrm{tr}(ZF_i x_i) + \mathrm{tr}(ZF_0) = \vec{c}^T\vec{x} + \mathrm{tr}(F_0 Z). \tag{5.5}$$

Then, the duality gap $\eta := \mathrm{tr}(ZF(\vec{x}))$ denotes the violation of this inequality.

Consider now the optimal value of the primal and dual semidefinite program

$$p^* = \inf\left\{\vec{c}^T\vec{x}|F(\vec{x}) \geq 0\right\} \tag{5.6}$$

$$d^* = \sup\left\{-\mathrm{tr}(F_0 Z)|Z = Z^T \geq 0, \mathrm{tr}(F_i Z) = c_i, i = 1, \ldots, m\right\}. \tag{5.7}$$

Then, for any feasible point Z and $\vec{x}$

$$-\mathrm{tr}(ZF_0) \leq p^* \tag{5.8}$$

$$d^* \leq \vec{c}^T\vec{x} \tag{5.9}$$

$$\Rightarrow d^* \leq p^*, \tag{5.10}$$

respectively. That is, every feasible point $\vec{x}$ of the primal problem provides a natural upper bound, whereas every feasible dual point Z provides a natural lower bound to the optimal values of both problems.

In convex optimisation theory two questions arise. The first one is, under which conditions the primal and dual objectives are equal, $p^* = d^*$, and whether the primal and dual optimal sets

$$X_{opt} = \left\{\vec{x}|F(\vec{x}) \geq 0 \text{ and } \vec{c}^T\vec{x} = p^*\right\} \text{ and} \tag{5.11}$$

$$Z_{opt} = \{Z|Z \geq 0, \mathrm{tr}(ZF_i) = c_i, i = 1, \ldots, m \text{ and } -\mathrm{tr}F_0 Z = d^*\} \tag{5.12}$$

are non-empty. The following theorem answers these questions:

Theorem 5.1 (Strong duality theorem)**.** *The primal and dual objectives are equal, $p^* = d^*$, if either:*

1. *The primal problem as given by Eq. (5.1) is strictly feasible, i.e., there exists an $\vec{x}$ with $F(\vec{x}) > 0$.*
2. *The dual problem as given by Eq. (5.3) is strictly feasible, i.e., there exists a Z with $Z = Z^T > 0$ and $\mathrm{tr}(F_i Z) = c_i$ for all $i = 1, \ldots, m$.*

The optimal sets X_{opt} and Z_{opt} are non-empty if both conditions hold.

A proof can be found in [118].

[3]Technically, the primal and dual problem should be defined using inf and sup, respectively. Here, however, it is assumed, that strictly feasible points $\vec{x}$ and Z satisfying $F(\vec{x}) > 0$, $Z > 0$ and $\mathrm{tr}(F_i Z) = c_i$ exist. In this case inf and sup can be replaced by min and max, since Theorem 5.1 ensures the non-emptiness of the optimal sets. Hence, inf and sup will be used only if it cannot be avoided.

[4]It can be transformed into the form given by Eq. (5.1) as shown in Ref. [68].

5.2 Central path following algorithm

One of the main features of semidefinite programming next to Theorem 5.1 is the fact that there exit numerical methods to approximately solve the semidefinite program as given by Eq. (5.1) with certified precision. Most of the solvers are based on so-called interior point methods, the main ideas of which we sketch here. For a complete discussion on the subject we refer the reader to Refs. [68, 119] and references therein.

For the rest of this section, the case, where the feasible sets of the primal and dual problem are non-empty and the primal and dual objectives are equal $p^* = d^*$, is considered. The main idea of interior point methods is to generate a sequence of feasible points $\vec{x}^{(k)}$ and $Z^{(k)}$, where $x^{(k)}$ is said to be a suboptimal point and the dual point $Z^{(k)}$ upper bounds the distance to the unknown primal objective in terms of the duality gap $\eta^{(k)} = \vec{c}^T\vec{x}^{(k)} + \mathrm{tr}(F_0 Z^{(k)})$ by

$$\vec{c}^T\vec{x}^{(k)} - p^* \leq \eta^{(k)}. \tag{5.13}$$

The points are chosen in such a way that with each step of iteration the duality gap $\eta^{(k)}$ decreases and the primal and dual objective converge to the optimum as shown in Fig. 5.1. The iteration terminates if some pre-specified fault tolerance $\eta^{(l)} < \epsilon$ is reached. The duality gap then not only signals the exit condition but also guarantees that $\vec{c}^T\vec{x}^{(l)} - p^*$ is not greater than ϵ.

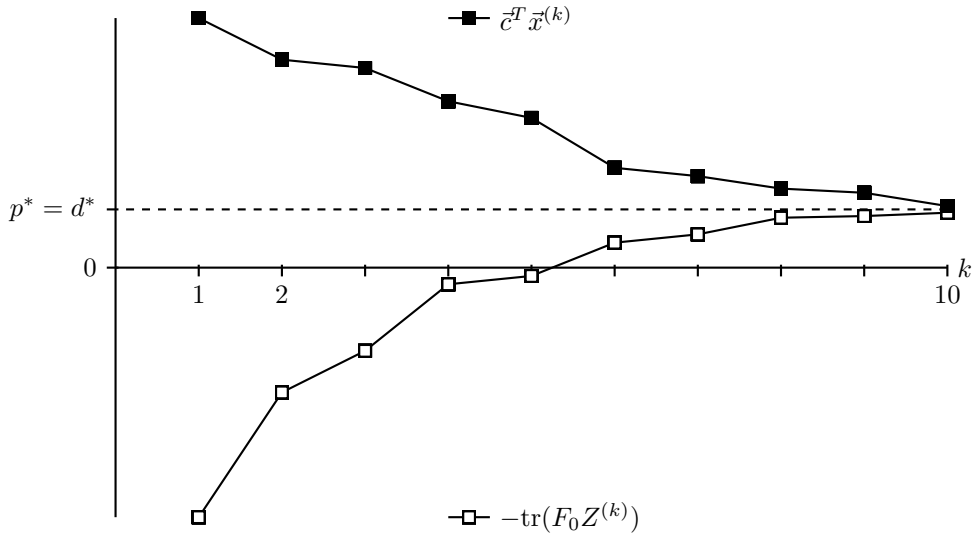


Figure 5.1: Primal-dual numerical algorithms generate a sequence of feasible primal $\vec{x}^{(k)}$ and dual $Z^{(k)}$ points, such that the primal and dual objective converge to the optimal value and the duality gap decreases.

A canonical way of generating a sequence of points is to stay close to the so-called central path [68]. That is a special path running through the feasible set[5], which leads to the optimal set. Note that most algorithms that solve semidefinite problems generate a sequence of points following this path. In practice these algorithms need between 5 and 50 steps to converge quite independent of the problem size. Note that the time needed to compute one step of the iteration scales with the number of variables m and the dimension of F_i and hence directly influences the overall computationally effort.

To introduce the central path, we first need to consider the barrier function

$$\phi(\vec{x}) = \begin{cases} \log\det F(\vec{x})^{-1} & \text{if } F(\vec{x}) > 0 \\ \infty & \text{else.} \end{cases} \tag{5.14}$$

This function is finite inside the strictly feasible region and diverges as it approaches the boundary of $\{\vec{x}|F(\vec{x}) \geq 0\}$. Furthermore, it is strictly convex and has an unique minimiser $\vec{x}^*$, which one refers to as

[5] In that sense, "central" refers to all points but the optimal on the path being strictly feasible.

analytic center

$$\vec{x}^* = \underset{F(\vec{x})\geq 0}{\operatorname{argmin}} \phi(\vec{x}). \tag{5.15}$$

Using the function in Eq. (5.14), one defines the central path $\vec{x}^*(\gamma)$ by

$$\begin{aligned} \vec{x}^*(\gamma) = \operatorname{argmin}\ & \log(\det[F(\vec{x})^{-1}]) && (5.16)\\ \text{s.t. } & F(\vec{x}) > 0, \\ & \vec{c}^T\vec{x} = \gamma, && (5.17) \end{aligned}$$

where $p^* < \gamma < \bar{p} = \sup\left\{\vec{c}^T\vec{x} | F(\vec{x}) > 0\right\}$. This path passes through the analytic center $\vec{x}^*$ and converges to an optimal point as $\gamma \to p^*$. The idea of so-called path-following algorithms is then to generate points on or close to the central path converging towards a sub-optimal point.

5.3 Implementing the renormalised genuine multiparticle negativity using CVXOPT

Up to this point, the basic notions of semidefinite programming have been reviewed. A brief introduction on the usage of two common convex optimisation frameworks can be found in the Appendix D.1. In this section, the steps needed to parse the convex optimisation problem defining the renormalised genuine multiparticle negativity as given by Eq. (2.1) using CVXOPT shall be discussed. Furthermore, we discuss the possibility to include symmetries of a quantum state to further improve the performance of the (renormalised) genuine multiparticle negativity.

Recall that for any quantum state ϱ, the renormalised genuine multiparticle negativity is given by

$$\begin{aligned} N_g(\varrho) = & -\inf \operatorname{tr}(\varrho\mathcal{W}) && (5.18)\\ & \text{subjected to: } \mathcal{W} = P_m + Q_m^{T_m}, \\ & \qquad 0 \leq P_m \\ & \qquad 0 \leq Q_m \leq \mathbb{1} \quad \text{for all partitions } m|\bar{m}. \end{aligned}$$

As opposed to the YALMIP implementation, which was done by Jungnitsch in Ref. [69], we have to manually parse the SDP and therefore have to transform it into the standard form given by Eq. (5.1). That is, the semidefinite constraints have to be provided in the form of real symmetric matrices F_i instead of complex Hermitian ones. Furthermore, we have to bring the optimisation goal into the form $\vec{c}^T\vec{x}$ and preferably resolve the equality $\mathcal{W} = P_m + Q_m^{T_m}$ for speed up.

To reformulate the SDP in Eq. (5.18) accordingly one has to perform two steps. First, one has to find a suitable operator basis, such that one can identify the optimisation variable $\vec{x}$ and the problem vector $\vec{c}$ of the problem. Second, one has to reformulate the constraints in terms of symmetric real matrices.

Before these steps are performed, rewrite Eq. (5.18) by explicitly using the equality $Q_m = (\mathcal{W} - P_m)^{T_m}$ and hence eliminating Q_m as free variable. That is

$$\begin{aligned} N_g(\varrho) = & -\inf \operatorname{tr}(\varrho\mathcal{W}) && (5.19)\\ & \text{s.t. } 0 \leq P_m, \\ & \qquad 0 \leq (\mathcal{W} - P_m)^{T_m} \leq \mathbb{1} \quad \text{for all partitions } m. \end{aligned}$$

The first step is done by choosing a Hermitian operator basis σ_i, $i = 1, \ldots, K$, such that $\operatorname{tr}(\sigma_i\sigma_j) = \delta_{ij}$. In this basis $\varrho = \sum_i \varrho^{(i)}\sigma_i$ with $\varrho^{(i)} = \operatorname{tr}(\varrho\sigma_i)$, $\mathcal{W} = \sum_i w^{(i)}\sigma_i$ and $P_m = \sum_i p_m^{(i)}\sigma_i$.

In this representation the optimisation variables are given by $w^{(i)}$ and $p_m^{(i)}$ for all partitions m and $i = 1, \ldots, K$. Writing these variables into the separate vectors $\vec{x}_w = \left(w^{(1)}, \ldots., w^{(K)}\right)$ and $\vec{x}_m = \left(p_m^{(1)}, \ldots., p_m^{(K)}\right)$ for all partitions m, one can define the problem vector $\vec{x}$ by concatenating all these vectors into a single one. That is for three particles A, B and C

$$\vec{x} = \left(\vec{x}_w, \vec{x}_A, \vec{x}_B, \vec{x}_C\right), \tag{5.20}$$

where the subscripts A, B and C denote the three inequivalent partitions $A|BC$, $B|AC$ and $C|AB$. Given $\vec{x}$ in the form above, it is possible to identify the problem vector from the given optimisation objective

$$\mathrm{tr}(\mathcal{W}\varrho) = \sum_i w^{(i)} \varrho^{(i)}. \tag{5.21}$$

That is the problem vector $\vec{c}$ is given by the concatenation of $\vec{c}_w = \left(\varrho^{(1)}, \ldots, \varrho^{(K)}\right)$ and $\vec{c}_m = \vec{0}$ for all partitions m

$$\vec{c} = \left(\vec{c}_w, \vec{c}_A, \vec{c}_B, \vec{c}_C\right). \tag{5.22}$$

Using $\vec{x}$ and $\vec{c}$ induced by the choice of the basis above yields

$$\mathrm{tr}(\mathcal{W}\varrho) = \sum_i w^{(i)} \varrho^{(i)} = \vec{c}^T \vec{x}. \tag{5.23}$$

In the second step the positivity constraints of the Hermitian complex matrices are replaced by positivity constraints of real symmetric ones. Note that the complex Hermitian operators $\mathcal{W}$ and P_m forming the semidefinite constraints are linear functions of $\vec{x}$. To highlight this dependence we also denote these operators by $\mathcal{W}(\vec{x})$ and $P_m(\vec{x})$.

Consider an arbitrary complex Hermitian operator $A \in \mathbb{C}^{k\times k}$. Every such operator can be decomposed into a real symmetric and an imaginary anti-symmetric part $A = R + iI$, with $R, I \in \mathbb{C}^{k\times k}$.

Recall that A is positive semidefinite if and only if for all $\vec{x} = \vec{x}_R + i\vec{x}_I \in \mathbb{C}^k$

$$\vec{x}^\dagger A \vec{x} = \vec{x}_R^T R \vec{x}_R + 2\vec{x}_I^T I \vec{x}_R + \vec{x}_I^T R \vec{x}_I \geq 0, \tag{5.24}$$

where we have used $\vec{x}_I^T I \vec{x}_R = -\vec{x}_R^T I \vec{x}_I$. One can define a linear map, mapping the real and imaginary part of any Hermitian A to a real symmetric operator $\phi(A) \in \mathbb{R}^{2k\times 2k}$. That is

$$\phi(A) := \begin{pmatrix} R & -I \\ I & R \end{pmatrix}. \tag{5.25}$$

If one considers $\vec{r} \in \mathbb{R}^{2k}$ with the first k entries labeled by $\vec{r_R} \in \mathbb{R}^k$ and the second k by $\vec{r_I} \in \mathbb{R}^k$, then $\phi(A)$ is positive if and only if

$$\vec{r}^T \phi(A) \vec{r} = \vec{r}_R^T R \vec{r}_R + 2\vec{r}_I^T I \vec{r}_R + \vec{r}_I^T R \vec{r}_I \geq 0. \tag{5.26}$$

Hence, the positivity constraints of any Hermitian operator can be reformulated as positivity constraint of a real symmetric operator.

Corollary. *A Hermitian operator $A \in \mathbb{C}^{k\times k}$ is positive (semi)definite if and only if the real symmetric operator $\phi(A) \in \mathbb{R}^{2k\times 2k}$ is positive (semi)definite.*

Further, since ϕ is a linear mapping and $A(\vec{x})$ is linear in $\vec{x}$, $\phi(A(\vec{x}))$ is also linear in $\vec{x}$.

Summarizing the last two steps, the SDP given in Eq. (5.18) can be reformulated in the standard form as given by Eq. (5.1). That is

$$\begin{aligned} N_g(\varrho) = -\inf & \vec{c}^T \vec{x} \\ \text{s.t. } & 0 \leq \phi(P_m(\vec{x})), \\ & 0 \leq \phi\left(\left[\mathcal{W}(\vec{x}) - P_m(\vec{x})\right]^{T_m}\right) \leq \mathbb{1} \quad \text{for } m \in \{A, B, C\}, \end{aligned} \tag{5.27}$$

where the constraints are given by real symmetric matrices, which are linear in the variable vector $\vec{x}$.

According to the discussion in this section the complete CVXOPT code implementing the renormalised genuine multiparticle negativity can be found in Listing D.7. The main part of the program is given by the function `gmn` in lines 76 to 208, which parses and solves the SDP given by Eq. (5.27).

5.3.1 Exploiting symmetries

This paragraph discusses, how a general class of symmetries of a state can be used to reduce the number of variables in the semidefinite program given in Eq. (5.27).

In Ref. [15], the authors noted that for any graph-diagonal state[6] the semidefinite program defining the genuine multiparticle negativity (GMN) can be reformulated as linear program, which drastically reduces the computational effort necessary to solve the optimisation, which also holds true for the renormalised GMN. Furthermore, for states, which are permutationally invariant the computational effort can be drastically reduced as shown in Ref. [120].

The gain in exploiting such symmetries is twofold. The obvious reason is to speed up the run-time needed to compute the (renormalised) genuine multiparticle negativity for a given state. The other less obvious reason is that by reducing the problem to the necessary minimum, it in general requires less memory during computation than the standard implementation. This might even allow to compute the (renormalised) genuine multiparticle negativity for large systems, which usually do not fit into the computer memory and hence would require a computer cluster to be computed or would be practically not computable.

Let $\mathcal{H} = \mathbb{C}^k = \bigotimes_{i=1}^n C^{k_i}$, $\sum_{i=1}^n k_i = k$ be the finite dimensional multiparticle Hilbert space of the whole system. Any mixed state is given by a positive semidefinite operator $\varrho \in \mathbb{C}^{k\times k}$ with $\mathrm{tr}\varrho = 1$. Further, consider the set of linear operations $\varphi : \mathbb{C}^{k\times k} \mapsto \mathbb{C}^{k\times k}$, which are positive[7], i.e., $\varphi(A) \geq 0$ if $A \geq 0$, and trace class preserving, i.e., $\mathrm{tr}(\varphi(\varrho)) \leq 1$ for any mixed state ϱ. Then the dual φ^* of φ is given by $\mathrm{tr}(\phi(A^\dagger)B) = \mathrm{tr}(A^\dagger\varphi^*(B))$ for all $A, B \in \mathbb{C}^{k\times k}$.

Define such an operation φ to be a symmetry of a state ϱ iff $\varphi(\varrho) = \varrho$. Such a symmetry may be used to further restrict the set of operators P_m and Q_m in the definition of the genuine multiparticle negativity as given by Eq. (5.18).

Lemma 5.2. *Let ϱ be a mixed state, φ be a symmetry of ϱ and let $\mathcal{O} \subset \mathbb{C}^{k\times k}$ denote the set of all positive Hermitian operators.*

If φ commutes with all partial transpositions T_m of the system $[\varphi, T_m] = 0$, i.e., $\varphi(A^{T_m}) = \varphi(A)^{T_m}$ for all $A \in \mathbb{C}^{k\times k}$, then the renormalised genuine multiparticle negativity as given in Eq. (5.18) equals

$$\begin{aligned} N_g(\varrho) = -\inf \mathrm{tr}\,(\varrho\mathcal{W}) \qquad & (5.28)\\ \textit{subjected to: } & \mathcal{W} = P_m + Q_m^{T_m}\\ & 0 \leq P_m,\ P_m \in \varphi^*(\mathcal{O})\\ & 0 \leq Q_m \leq \mathbb{1},\ Q_m \in \varphi^*(\mathcal{O}) \ \textit{ for all partitions } m|\bar{m}. \end{aligned}$$

Proof. To prove this statement it suffices to show that the optima in Eqs. (5.18) and (5.28) are equal.

Let $W = P_m + Q_m^{T_m}$ be an optimal point of Eq. (5.18). Now map P_m and Q_m to $\tilde{P}_m = \varphi^*(P_m)$ and $\tilde{Q}_m = \varphi^*(Q_m)$. φ is assumed to be positive. By definition of the dual, φ^* is positive as well. Hence, $\tilde{P}_m \geq 0$ and $\tilde{Q}_m \geq 0$. Since $\mathrm{tr}(\varphi(\varrho)) \leq 1$ and $Q_m \leq \mathbb{1}$,

$$\mathrm{tr}\,(\varrho(\mathbb{1} - \varphi^*(Q_m))) = \mathrm{tr}(\varrho) - \mathrm{tr}(\varphi(\varrho)Q_m) \geq \mathrm{tr}(\varrho) - \mathrm{tr}(\varphi(\varrho)) \geq 0 \qquad (5.29)$$

for all density operators ϱ. That is $\tilde{Q}_m \leq \mathbb{1}$. Using $\varphi^*(Q_m^{T_m}) = \varphi^*(Q_m)^{T_m}$ it follows that $\tilde{W} = \tilde{P}_m + \tilde{Q}_m^{T_m} = \varphi^*(W)$ is a witness of the optimisation given in Eq. (5.28). Furthermore, it yields the same value as the renormalised genuine negativity as given in Eq. (5.18)

$$\mathrm{tr}\,(\varrho W) = \mathrm{tr}\,(\varphi(\varrho)W) = \mathrm{tr}\,(\varrho\varphi^*(W)) = \mathrm{tr}(\varrho\tilde{W}). \qquad (5.30)$$

Conversely, every optimal witness of Eq. (5.28) naturally satisfies the constraints of the optimisation given in Eq. (5.18) and hence provides a witness of the original optimisation with the same value. It follows that both semidefinite programs yield the same optimum. □

Remark 5.1. In case the image $\varphi^*(\mathcal{O})$ is a proper subspace of $\mathcal{O}$, any orthogonal operator basis in $\varphi^*(\mathcal{O})$ has less elements than any other orthogonal operator basis in $\mathcal{O}$. Hence the optimisation given by Eq. (5.28)

[6] At this point it suffices to note that a graph-diagonal state ϱ is invariant under a certain commutative group G consisting of local unitary transformations. I.e., $g^\dagger\varrho g = \varrho$ for all $g \in G$.

[7] For the most general type of quantum operations one demands complete positivity. That is $\varphi\otimes\mathbb{1}_n$ is positive for all identity maps $\mathbb{1}_n \in \mathbb{C}^{n\times n}$. Here we do not need this stronger assumption and therefore demand positivity only.

has less independent variables to optimise over than the original problem given by Eq. (5.27). Given a suitably designed implementation this can lead to a considerable speed up of the numerical calculation of the (renormalised) genuine multiparticle negativity.

Moreover, if a state has several symmetries φ_i such that Lemma 5.2 can be applied, then the operators P_m and Q_m can be assumed to lie in the intersection of all subspaces $\varphi_i^*(\mathcal{O})$. That is

$$P_m, Q_m \in \bigcap_i \varphi_i^*(\mathcal{O}). \tag{5.31}$$

Example 5.1. As an example consider the n-qubit Hilbert space $\mathcal{H} = \otimes_{i=1}^n \mathbb{C}^2$ and an arbitrary stabiliser subgroup $T = \{\mathbb{1}, g\}$ generated by a single generator g[8] as introduced in Paragraph 4.1.1.

We use g to define a projection onto the T-symmetric subspace, the space of operators satisfying $A = g^\dagger A g$ for all $A \in \mathbb{C}^{k\times k}$, which includes the set of T-stabilised states. This projection is given by the linear mapping

$$\varphi_T(A) := \frac{1}{2}(A + g^\dagger A g). \tag{5.32}$$

Its positivity follows from the fact that $g^\dagger A g$ is positive for all $A \geq 0$ and it is trace class preserving since the trace is cyclic and $g^\dagger g = \mathbb{1}$. Furthermore, due to the tensor product structure of g, φ_g commutes with all possible transpositions T_m and hence satisfies all requirements of the Lemma above.

Lemma 5.2 then states that given a state ϱ_T, with $\varrho_T = \varphi_T(\varrho_T)$, i.e., a T-stabilised state, we can perform the optimisation according to Eq. (5.28). That is $P_m, Q_m \in \varphi_T^*(\mathcal{O}) \subset \mathcal{O}$ can be chosen to be Hermitian operators, such that $A = \varphi_T^*(A) = \frac{1}{2}(A + gAg^\dagger)$ for all P_m, Q_m. To determine a suitable basis of $\varphi_T^*(\mathcal{O})$ consider the orthogonal Hermitian operator basis

$$\mathcal{B} = \{a_1 \otimes a_2 \otimes \cdots \otimes a_n | a_i \in \{\mathbb{1}_2, \sigma_x, \sigma_y, \sigma_z\}\}, \tag{5.33}$$

which we used already in Paragraph 4.1.1. Since our mapping φ_T is linear, $\varphi_T^*(\mathcal{O})$ is given by the span of $\varphi_T^*(\mathcal{B})$.

Recall that for each $A \in \mathcal{B}$, either $g^\dagger A g = A$ and thus $\varphi_T^*(A) = A$ or $g^\dagger A g = -A$[9] and thus $\varphi_T^*(A) = 0$. Then $\mathcal{B}_T = \{A \in \mathcal{B} | [A, g] = 0 \text{ for all } g \in T\}$ is a Hermitian operator basis of $\varphi_T^*(\mathcal{O})$. Rephrased: The operators P_m and Q_m can be chosen to be invariant under the symmetry group T.

In case $g \neq \mathbb{1}$ the basis $\mathcal{B}_T$ has strictly fewer elements than $\mathcal{B}$ and hence the semidefinite program given by Eq. (5.28) has fewer variables than the original optimisation problem given by Eq. (5.18), which results in a noticeable speed up.

Example 5.2. Recall the two-qubit X-states, which have been introduced in Example 4.1. These states are stabilised by the stabiliser subgroup T generated by the single generator $t_1 = \sigma_z \otimes \sigma_z$. In the two qubit case there are only two operators P and Q to optimise to compute the genuine multiparticle negativity, which coincides with the bipartite negativity in this case. According to our discussion the operators P and Q can be expressed using the set

$$\mathfrak{B}_T = \{\mathbb{1}\otimes\mathbb{1}, \mathbb{1}\otimes\sigma_z, \sigma_z\otimes\mathbb{1}, \sigma_z\otimes\sigma_z, \sigma_x\otimes\sigma_x, \sigma_x\otimes\sigma_y, \sigma_y\otimes\sigma_x, \sigma_y\otimes\sigma_y\}. \tag{5.34}$$

That is $P = \sum_{b\in\mathfrak{B}_T} p_b b$ and $Q = \sum_{b\in\mathfrak{B}_T} q_b b$. Hence, one has to optimise over 8 free parameters per operator, which is half the number of variables one would usually have.

Remark 5.2. In a more general scenario, we consider the case, where a state ϱ is invariant under an arbitrary stabiliser subgroup T. I.e., $\varphi_g(\varrho) = \varrho$ for all $t \in T$. In this case the projection onto the T-symmetric subspace is given by

$$\varphi_T(A) := \frac{1}{|T|}\sum_{t\in T} t^\dagger A t, \tag{5.35}$$

where $|T|$ is the cardinality of the subgroup. In this case by a similar reasoning as in the example above one can show, that φ_T satisfies all conditions of Lemma 5.2. As a consequence one may without loss of generality chose the operators P_m and Q_m from the subspace spanned by

$$\mathcal{B}_T = \{A \in \mathcal{B} | [A, t] = 0 \text{ for all } t \in T\}. \tag{5.36}$$

[8] In fact the following discussion holds true for all finite groups generated by a local unitary transformation g of $\mathcal{H}$, which satisfies $gg = \mathbb{1}$. Note that the arguments may be modified, such that the reasoning holds also true if $g^k = \mathbb{1}$.

[9] For the single Pauli matrices σ_x, σ_y and σ_z we have $\sigma_a\sigma_b\sigma_a = \sigma_a$ if $a = b$ and $\sigma_a\sigma_b\sigma_a = -\sigma_b$ otherwise.

It follows that for all stabilised states one may speed up the calculation of the (renormalised) genuine multiparticle negativity by choosing a suitable operator basis and hence reducing the total number of parameters in the optimisation.

Example 5.3. Similar to the line of thoughts in the last example it follows that if a state ϱ is invariant under transposition $\varrho = \varrho^T$, the map

$$\varphi(A) := \frac{1}{2}(A + A^T) \tag{5.37}$$

induces a restriction of P_m and Q_m to be real symmetric instead of Hermitian complex. This effectively reduces the number of parameters by a factor of two.

5.4 Conclusions

In this chapter the basic concepts and notions of semidefinite programming were reviewed: a special problem in convex optimisation theory, where a linear target function is optimised over a convex set defined by the positivity of a linear matrix. We outlined the main idea of the so-called central path following algorithms, which aim to numerically compute suboptimal points, which are certified to be only ε-suboptimal.

In semidefinite programming one is originally restricted to real symmetric matrices, whereas in quantum mechanics we usually deal with Hermitian operators. Therefore a way to set the positivity condition of a Hermitian complex matrix in one to one correspondence with the positivity condition of a specific real symmetric matrix on an enlarged real matrix space was provided. Following, this correspondence was used to implement the renormalised genuine multiparticle negativity using CVXOPT. Finally, a method how symmetries of a state can be used to possibly reduce the number of parameters in the semidefinite program defining the (renormalised) genuine multiparticle negativity was shown. We then showed that this method can be applied to reduce the computational effort to compute the genuine multiparticle negativity for stabilised states.

Conclusions

In this thesis, we pointed out the usefulness of the genuine multiparticle negativity to quantify entanglement. We showed that it can be characterised in an analytical way and used to study genuine multiparticle entanglement in many-body systems. Furthermore, we generalised the stabiliser formalism and pointed out a connection between its inherent symmetries and the optimisation defining the genuine multiparticle negativity. In detail, the following results were obtained:

A) We have shown that the renormalised genuine multiparticle negativity can be expressed in two equivalent ways: as an optimisation over suitable normalised fully decomposable witnesses and as a mixed convex roof of the minimal bipartite negativity. Using this duality, we obtained an exact algebraic prescription of the genuine multiparticle negativity for the n-qubit GHZ-diagonal and four-qubit cluster-diagonal states.

B) Furthermore, we investigated the connection between genuine multiparticle entanglement and quantum phase transitions in the one-dimensional Ising model in a transverse magnetic field. We identified spatial distributions of three and four particles, where entanglement is present, showed that the derivative of the genuine multiparticle negativity diverges logarithmically at the critical points and confirmed that finite-size scaling holds close to the quantum phase transition.

C) Finally, we generalised the stabiliser formalism by introducing stabilised states. These include states, which are diagonal in a stabiliser basis [61] and the generalised n-qubit X-states [108]. Further, the stabiliser subgroups were characterised with respect to equivalence under qubit-permutations and local Clifford operations. Subsequently, we provided a general method to use symmetries to simplify the optimisation defining the genuine multiparticle negativity. Applying this method to the stabilised states, we showed how the quantification of entanglement using the genuine multiparticle negativity can be speed up.

There are several questions arising, which might be investigated in the future.

A) Since the method to obtain the analytic expression of the genuine multiparticle negativity is quite general, it should be possible to find closed expressions for other highly symmetric state families such as other graph-diagonal states [61] or states with $U \otimes U \otimes U$ symmetry [87]. Additionally, it would be desirable to obtain an operational interpretation for the genuine multiparticle negativity: Because the bipartite logarithmic negativity is the upper bound for distillable entanglement [51], one may speculate that the genuine multiparticle negativity is connected to the distillation rate of genuine multiparticle entangled states. Also, the measure may be related to SLOCC-classes in the multiparticle case and the dimensionality of multiparticle entanglement [88, 89]. Furthermore, recall that the shareability of quantum correlations among many parties is limited and these restrictions are known as monogamy relations [90–93]. For example, for a three-qubit system the bipartite entanglement of the splitting $A|BC$ as measured by the concurrence is given by the entanglement in the reduced marginals plus the three-tangle τ_3 as a genuine tripartite contribution, $C^2_{A|BC} = C^2_{AB} + C^2_{AC} + \tau_3$ [90]. As a future project it would be desirable to derive relations, which correlate the genuine multiparticle negativity with the bipartite negativities of the system and its parts.

B) In many-body systems, our methods allow the direct study of genuine multiparticle entanglement. Hence, it can be used to complete existing indirect results on multiparticle entanglement. To give an example, in Ref. [34] the existence of multiparticle entanglement in the one-dimensional XYZ-model was concluded indirectly from monogamy relations and possible connections to phase transitions were found.

Our approach allows to verify these results in a direct manner. Similarly, in Ref. [98] the two-partite negativities for different splittings of three particles in the XY-model were studied. Our methods can now decide, whether these bipartite quantities are connected to genuine multiparticle entanglement. Further, in Ref. [49] certain three-particle reduced states could neither be detected as genuine multiparticle entangled nor shown to be biseparable. We expect that an application of our method would solve this issue. Additionally, our results demonstrate the usefulness of the genuine multiparticle negativity to study many-body systems. This makes it applicable to further systems, such as dynamical phase transitions [103], quenching dynamics or the study of symmetry breaking [49].

C) Since stabilised states are easily characterised by their symmetries, many properties of these states can be studied solely on the basis of the related symmetry groups. Recall that within the graph state formalism stabiliser groups of two graphs are LC-equivalent if the corresponding graphs can be transformed into each other by local complementation [26, 61, 111]. Naturally, the question arises if there is a way to relate graphs or operations thereof to stabiliser subgroups and whether this correspondence can be used to obtain the LC-equivalence classes. Furthermore, it would be interesting to extend the LC-classification to a LU-classification to completely remove the ambiguity of local basis change. It would be also interesting to investigate certain aspects such as genuine multiparticle entanglement of stabilised states, which are hard to grasp on the full state space. Finally, the question remains whether stabilised states can be used for quantum tasks other than those, for which graph states have already proven to be useful.

Appendix A

Characterizing the GMN

I thought up an ending for my book. "And he lives happily ever after, till the end of his days."
Bilbo Baggins

A.1 Proof of Theorem 2.1

To start, we recall some notions of semidefinite programing [68]. The primal problem of an SDP reads

$$\begin{aligned} &\inf_{\vec{x}} \ \vec{c}^T \vec{x} \\ &\quad \text{s.t. } F(\vec{x}) = F_0 + \sum_i x_i F_i \geq 0, \end{aligned} \tag{A.1}$$

where $\vec{c}, \vec{x} \in \mathbb{R}^n$ and $F_i = F_i^\dagger \in \mathbb{C}^{m\times m}$. The scalar product $\vec{c}^T\vec{x}$ is the linear function to minimise and the linear matrix inequality $F(\vec{x}) \geq 0$, understood in terms of positive semi-definiteness holds all the constraints of the optimisation. The dual problem to this primal problem is given by

$$\begin{aligned} &\sup_Z \left[-\mathrm{tr}\left(F_0 Z\right)\right] \\ &\quad \text{s.t. } \mathrm{tr}\left(F_i Z\right) = c_i \ \text{ for all } i = 1, \ldots, n, \\ &\qquad Z \geq 0. \end{aligned} \tag{A.2}$$

A point $\vec{x}$ or Z is called feasible if it meets the constraints of the primal $F(\vec{x}) \geq 0$ or dual, respectively $Z \geq 0$ and $\mathrm{tr}(F_i Z) = c_i$. For any pair of feasible points both problems are connected to each other via $-\mathrm{tr}F_0 Z \leq \vec{c}^T\vec{x}$. Moreover, if at least one of the problems is strictly feasible, i.e., that either a primal point $\vec{x}$ satisfying $F(\vec{x}) > 0$ or a dual point Z satisfying $Z > 0$ and $\mathrm{tr}(F_i Z) = c_i$ exists, Theorem 3.1 of Ref. [68] ensures that both problems yield the same optimum $\sup_Z \{-\mathrm{tr}(F_0 Z) | Z \geq 0 \text{ and } \mathrm{tr}(F_i Z) = c_i\} = \inf_{\vec{x}} \left\{\vec{c}^T\vec{x} | F(\vec{x}) \geq 0\right\}$.

The idea of the proof goes as follows. Using our renormalised GMN as given by Eq. (2.1) as the primal problem of a SDP we show that the corresponding dual problem is given by the left-hand-side of Eq. (2.2). Equality then follows by showing strict feasibility of the primal problem. We start the proof by rewriting the semidefinite program (2.1) as

$$\begin{aligned} N_g(\varrho) = -\inf \ &\mathrm{tr}\left(\varrho \mathcal{W}\right) \\ &\text{s.t. } 0 \leq P_m, \\ &\quad 0 \leq \left(\mathcal{W} - P_m\right)^{T_m} \leq \mathbb{1} \ \text{ for all partitions } m. \end{aligned} \tag{A.3}$$

For the sake of readability we write down the proof by assuming a quantum system composed of three parts A, B and C. A generalisation to larger particle numbers is straightforward.

We choose a Hermitian operator basis σ_i, $i = 1, \ldots, K$, such that $\mathrm{tr}\left(\sigma_i \sigma_j\right) = \delta_{ij}$. In this basis $\varrho = \sum_i \varrho^{(i)} \sigma_i$, $\mathcal{W} = \sum_i w^{(i)} \sigma_i$ and $P_m = \sum_i p_m^{(i)} \sigma_i$ for $m \in \{A, B, C\}$. We gather the components of this

decompositions in the vectors

$$\begin{aligned} \vec{x}_w &= \left(w^{(1)}, \ldots., w^{(K)}\right), \qquad \vec{x}_m = \left(p_m^{(1)}, \ldots., p_m^{(K)}\right), \\ \vec{c}_w &= \left(\varrho^{(1)}, \ldots., \varrho^{(K)}\right), \qquad \vec{c}_m = \vec{0}, \end{aligned} \tag{A.4}$$

where $\vec{x}_w$ are the coefficients of $\mathcal{W}$, $\vec{x}_m$ are the coefficients of P_m and $\vec{c}_w$ as well as the $\vec{c}_m$ characterise parts of the optimisation goal. If we merge these vectors into

$$\vec{x} = (\vec{x}_w, \vec{x}_A, \vec{x}_B, \vec{x}_C) \text{ and } \vec{c} = (\vec{c}_w, \vec{c}_A, \vec{c}_B, \vec{c}_C), \tag{A.5}$$

we can rewrite the SDP (A.3) as

$$\begin{aligned} &- \inf_{\vec{x}} \vec{c}^T \vec{x} \\ &\text{s.t. } F(x) = F_0 + \sum_i x_i F_i \geq 0, \end{aligned} \tag{A.6}$$

where $F(\vec{x})$ has the following block-diagonal form

$$\begin{aligned} F(\vec{x}) \;=\; & diag(P_A, P_B, P_C \mid (\mathcal{W} - P_A)^{T_A}, (\mathcal{W} - P_B)^{T_B}, (\mathcal{W} - P_C)^{T_C} \mid \\ & \mathbb{1} - (\mathcal{W} - P_A)^{T_A}, \mathbb{1} - (\mathcal{W} - P_B)^{T_B}, \mathbb{1} - (\mathcal{W} - P_C)^{T_C}). \end{aligned} \tag{A.7}$$

Here the vertical lines "|" are introduced for notational convenience in order to to split the block-diagonal matrix $F(\vec{x})$ into three parts, each of which consists of three sub blocks. The first represents the constraint $0 \leq P_m$, the second ensures $0 \leq (\mathcal{W} - P_m)^{T_m}$ (equivalent to $0 \leq Q_m$) and the last one bounds $(\mathcal{W} - P_m)^{T_m} \leq \mathbb{1}$ (equivalent to $Q_M \leq \mathbb{1}$) for all $m \in \{A, B, C\}$. According to $F(\vec{x}) = F_0 + \sum_i x_i F_i = F_0 + \sum_j (\vec{x}_w)_j F_{w,j} + \sum_m \sum_j (\vec{x}_m)_j F_{m,j}$, we have:

$$\begin{aligned} F_0 &= diag\,(0,0,0 \mid 0,0,0 \mid \mathbb{1}, \mathbb{1}, \mathbb{1}), \\ F_{w,j} &= diag\left(0,0,0 \mid \sigma_j^{T_A}, \sigma_j^{T_B}, \sigma_j^{T_C} \mid -\sigma_j^{T_B}, -\sigma_j^{T_B}, -\sigma_j^{T_C}\right), \\ F_{A,j} &= diag\left(\sigma_j, 0, 0 \mid -\sigma_j^{T_A}, 0, 0 \mid \sigma_j^{T_A}, 0, 0\right), \\ F_{B,j} &= diag\left(0, \sigma_j, 0 \mid 0, -\sigma_j^{T_B}, 0 \mid 0, \sigma_j^{T_B}, 0\right), \\ F_{C,j} &= diag\left(0,0,\sigma_j \mid 0,0,-\sigma_j^{T_C} \mid 0,0,\sigma_j^{T_C}\right). \end{aligned} \tag{A.8}$$

The dual problem as given in Eq. (A.2) involves the calculation of $\mathrm{tr}\,(F_0 Z)$ and $\mathrm{tr}\,(F_i Z)$. To do so we can make use of the block-diagonal structure of the F_i and write the corresponding diagonal blocks of Z into a new block-diagonal matrix

$$Z_{bd} = diag\left(Z_A, Z_B, Z_C \mid Z_A^+, Z_B^+, Z_C^+ \mid Z_A^-, Z_B^-, Z_C^-\right). \tag{A.9}$$

Note that the positivity of Z ensures the positivity of each block in Z_{bd}. On the other hand if the blocks in Z_{bd} are positive so is Z_{bd}. We can now evaluate $-\mathrm{tr}\,(F_0 Z) = -\mathrm{tr}\,(F_0 Z_{bd})$ to write down the dual objective

$$-\sup_{Z \geq 0} \left[-\mathrm{tr}\left(Z_A^-\right) - \mathrm{tr}\left(Z_B^-\right) - \mathrm{tr}\left(Z_C^-\right)\right] = \inf_{Z \geq 0} \sum_m \mathrm{tr}\left(Z_m^-\right). \tag{A.10}$$

The constraints $\mathrm{tr}\,(F_i Z) = c_i$ can be evaluated similarly and split up into two types $\mathrm{tr}\,(F_{w,j} Z) = \varrho_j$ and $\mathrm{tr}\,(F_{m,j} Z) = 0$. In detail, they read

$$\sum_m \mathrm{tr}\left(\sigma_j^{T_m} Z_m^+\right) - \mathrm{tr}\left(\sigma_j^{T_m} Z_m^-\right) = \varrho^{(j)}, \tag{A.11}$$

$$\mathrm{tr}\left(\sigma^j Z_m\right) - \mathrm{tr}\left(\sigma_j^{T_m} Z_m^+\right) + \mathrm{tr}\left(\sigma_j^{T_m} Z_m^-\right) = 0, \tag{A.12}$$

with $m \in \{A, B, C\}$. If we multiply Eq. (A.12) by σ_j and sum over all j it immediately follows that

$$Z_m = Z_m^{+T_m} - Z_m^{-T_m}. \tag{A.13}$$

Eq. (A.11) multiplied by σ_j and summed over j together with Eq. (A.13) yields

$$\varrho = \sum_m Z_m^{+T_m} - Z_m^{-T_m} = \sum_m Z_m. \tag{A.14}$$

Actually, the dual problem optimises in state space, which can be made apparent by introducing the following notation. Let $p_m = \mathrm{tr}(Z_m)$ and $\varrho_m = Z_m/\mathrm{tr}(Z_m)$, then the constraint given by Eq. (A.14) corresponds to an optimisation over all possible convex combinations $\varrho = \sum_m p_m \varrho_m$ of mixed quantum states ϱ_m. By introducing $\varrho_m^{\pm} = Z_m^{\pm}/\mathrm{tr}(Z_m)$ the constraint given by Eq. (A.13) can be rewritten as $\varrho_m^{T_m} = \varrho_m^{+} - \varrho_m^{-}$. The dual problem is then given by

$$\begin{aligned} &\inf\ p_A \mathrm{tr}\left(\varrho_A^-\right) + p_B \mathrm{tr}\left(\varrho_B^-\right) + p_C \mathrm{tr}\left(\varrho_C^-\right) \\ &\quad \text{s.t. } \varrho = p_A\varrho_A + p_B\varrho_B + p_C\varrho_C \text{ is a decomposition of } \varrho \text{ and} \\ &\qquad \varrho_m^{T_m} = \varrho_m^{+} - \varrho_m^{-} \ \text{ for all } m \in \{A, B, C\} \text{ with } \varrho^{\pm} \geq 0, \end{aligned} \tag{A.15}$$

which means that given a density matrix ϱ one minimises $\sum_m p_m \mathrm{tr}\left(\varrho_m^-\right)$ over all decompositions $\varrho = \sum_m p_m \varrho_m$ and respective splittings of the partial transposition $\varrho_m^{T_m}$ into a difference of two positive semidefinite operators $\varrho_m^{T_m} = \varrho_m^{+} - \varrho_m^{-}$.

Note that one can split this optimisation into two steps. First, one has to optimise over all mixed state decompositions of ϱ, where each term in the decomposition is assigned to a certain bipartition. For a fixed decomposition $\varrho = \sum_m p_m \varrho_m$ one still has to minimise $\sum_m p_m\ \mathrm{tr}\left(\varrho_m^-\right)$ over $\bigcup_m \mathcal{N}_m$ with $\mathcal{N}_m = \{\varrho_m^{\pm} \geq 0 | \varrho_m^{T_m} = \varrho_m^{+} + \varrho_m^{-}\}$. This can be decomposed into the separate minimisation of each $\mathrm{tr}\left(\varrho_m^-\right)$ over $\mathcal{N}_m$. If one compares these single optimisations to the bipartite negativity [50, 51] of the respective partition

$$N_m(\varrho) = \inf\left\{\mathrm{tr}\left(\varrho^-\right) | \varrho^{T_m} = \varrho^+ - \varrho^-, \varrho^{\pm} \geq 0\right\}, \tag{A.16}$$

then $\min_{\mathcal{N}_m} \mathrm{tr}\left(\varrho_m^-\right) = N_m(\varrho_m)$. Hence we can rewrite the dual problem as given in Eq. (A.15) by

$$\begin{aligned} &\min\ p_A N_A(\varrho_A) + p_B N_B(\varrho_B) + p_C N_C(\varrho_C), \\ &\quad \text{s.t. } \varrho = p_A\varrho_A + p_B\varrho_B + p_C\varrho_C. \end{aligned}$$

Here we replaced the infimum by a minimum, since one optimises a continuous function over a closed and bounded set.

To finish this proof we still have to show that the primal problem is strictly feasible such that both problems have the same optimal value. We find that $\mathcal{W} = P_m + Q_m^{T_m}$ with $Q_m = P_m = \mathbb{1}/2 > 0$ is a strictly feasible point for the primal problem given by Eq. (2.1). Hence, the genuine multiparticle negativity equals the dual optimisation problem

$$N_g(\varrho) = \min_{\varrho = \sum_m p_m \varrho_m} \sum_m p_m N_m(\varrho_m). \tag{A.17}$$

□

A.2 Proof of Theorem 2.6

Within this proof we will shorten the Dirac notation by setting $|\alpha\beta\gamma\delta\rangle\langle\alpha\beta\gamma\delta| = |\alpha\beta\gamma\delta\rangle\langle\cdot|$. First, recall that a four-qubit cluster-diagonal state is biseparable iff the following inequalities are satisfied [85]:

$$F_{\alpha\beta\gamma\delta} \leq \frac{1}{2}\sum_{ij} F_{\bar{\alpha}ij\delta} + F_{\alpha ij\bar{\delta}} + F_{\alpha ij\delta} \quad \text{and} \tag{A.18}$$

$$F_{\alpha\beta\gamma\delta} + F_{\bar{\alpha}\mu\nu\bar{\delta}} \leq \frac{1}{2}\sum_{ij} F_{\alpha ij\delta} + F_{\bar{\alpha}ij\delta} + F_{\alpha ij\bar{\delta}} + F_{\bar{\alpha}ij\bar{\delta}}. \tag{A.19}$$

Furthermore, Lemma 2 in Ref. [85] states that the density matrix

$$\varrho^{\mathrm{bs}} = \frac{1}{2}\left(|ijkl\rangle\langle\cdot| + |\alpha\beta\gamma\delta\rangle\langle\cdot|\right) \tag{A.20}$$

is biseparable, unless $i \neq \alpha$ and $l \neq \delta$ both hold at the same time.

We now prove that the maximal violation of the relations (A.18) and (A.19) [this corresponds to the negative of the expectation values of some witness in Eqs. (2.29)] is an upper bound on the GMN. There are three cases:

Case one: None of the inequalities (A.18) and (A.19) is violated. In this case we already know that the state is biseparable [85] and hence $N_g(\varrho) = 0$, which coincides with the right-hand-side of (2.32) in this case.

Case two: The largest violation occurs in inequality (A.18). We can assume that it occurs for $\alpha = \beta = \gamma = \delta = +$, for other indices the reasoning is similar. Using $\sum_{ijkl} F_{ijkl} = 1$ and the fact that all other inequalities are less violated we obtain

$$F_{++++} - \frac{1}{2}\sum_{ij} F_{+ij+} + F_{-ij+} + F_{+ij-} \geq F_{++++} + F_{-\mu\nu-} - \frac{1}{2}, \tag{A.21}$$

for μ, ν arbitrary. Adding $\frac{1}{2}\sum_{ijkl} F_{ijkl} = \frac{1}{2}$ on both sides yields

$$F_{-\mu\nu-} \leq F_{-\mu\bar{\nu}-} + F_{-\bar{\mu}\nu-} + F_{-\bar{\mu}\bar{\nu}-}. \tag{A.22}$$

Now consider the state $\sigma \sim \sum_{\mu\nu} F_{-\mu\nu-}|\mu\nu-\rangle\langle\cdot|$. One can check that σ does not violate any of the inequalities (A.18) and (A.19), so it is biseparable. We choose $F^{ent}_{++++} = 2F_{++++} - \sum_{ij} F_{+ij+} + F_{-ij+} + F_{+ij-}$, which is two times the violation of inequality (A.18). Then we can decompose ϱ into a genuine multiparticle entangled part with weight F^{ent}_{++++} and a biseparable rest. This rest consists of a convex combination of biseparable states as in Eq. (A.20) and the biseparable state σ

$$\begin{aligned}\varrho = F^{ent}_{++++}|++++\rangle\langle\cdot| \; &+ \sum_{ij,ij\neq++} F_{+ij+}\left(|++++\rangle\langle\cdot| + |+ij+\rangle\langle\cdot|\right) \\ &+ \sum_{ij} F_{+ij-}\left(|++++\rangle\langle\cdot| + |+ij-\rangle\langle\cdot|\right) + \sum_{ij} F_{-ij+}\left(|++++\rangle\langle\cdot| + |-ij+\rangle\langle\cdot|\right) \\ &+ \sum_{ij} F_{-ij-}|-ij-\rangle\langle\cdot|. \end{aligned} \tag{A.23}$$

Since only the first part is not biseparable and the GMN is convex $N_g(\sum_m p_m\varrho_m) \leq \sum_m p_m N_g(\varrho_m)$, we obtain

$$N_g(\varrho) \leq F^{ent}_{++++} N_g(|++++\rangle\langle\cdot|) = \frac{1}{2}\big(2F_{\alpha\beta\gamma\delta} - \sum_{ij} F_{\alpha ij\delta} + F_{\bar{\alpha}ij\delta} + F_{\alpha ij\bar{\delta}}\big) \tag{A.24}$$

which corresponds to the right-hand side of Eq. (2.32) in this case.

Case three: The largest violation occurs in Eq. (A.19). Without loss of generality for $\alpha = \beta = \gamma = \delta = +$ and $\mu = \nu = -$. Relating the inequalities (A.18) and (A.19) with each other as in the second case we obtain

$$F_{+--+} \geq \sum_{ij,ij\neq--} F_{+ij+} = F_{++} \quad \text{and} \quad F_{----} \geq \sum_{ij,ij\neq--} F_{-ij-} = F_{--}. \tag{A.25}$$

As a direct consequence we can split each of F_{++++} and F_{----} into two non-negative parts

$$F_{++++} = \tilde{F}_{++++} + F_{++}, \quad \text{and} \quad F_{----} = \tilde{F}_{----} + F_{--}. \tag{A.26}$$

With this definition of $\tilde{F}_{++++}$ and $\tilde{F}_{----}$ we can write

$$\begin{aligned}\varrho =& \tilde{F}_{++++}|++++\rangle\langle\cdot| + \tilde{F}_{----}|----\rangle\langle\cdot| \\ &+ \sum_{ij} F_{-ij+}|-ij+\rangle\langle\cdot| + F_{+ij-}|+ij-\rangle\langle\cdot| + \sigma_1, \end{aligned} \tag{A.27}$$

where the state

$$\sigma_1 = \sum_{ij,ij\neq++} F_{+ij+}(|+++\rangle\langle\cdot| + |+ij+\rangle\langle\cdot|) + \sum_{ij,ij\neq--} F_{-ij-}(|---\rangle\langle\cdot| + |-ij-\rangle\langle\cdot|) \tag{A.28}$$

is biseparable. With these replacements the largest violation of inequality (A.19) is given by

$$\mathcal{V} \equiv \frac{1}{2}\tilde{F}_{++++} + \frac{1}{2}\tilde{F}_{----} - \frac{1}{2}\sum_{ij} F_{+ij-} + F_{-ij+}. \tag{A.29}$$

One can decompose $\tilde{F}_{++++}$ and $\tilde{F}_{----}$ further into two non-negative parts

$$\tilde{F}_{++++} = F^{ent}_{++++} + F^{bs}_{++++} \quad \text{and} \quad \tilde{F}_{----} = F^{ent}_{----} + F^{bs}_{----}, \tag{A.30}$$

such that

$$F^{bs}_{++++} + F^{bs}_{----} = \textstyle\sum_{ij} F_{+ij-} + F_{-ij+} \quad \text{and } \frac{1}{2}F^{ent}_{++++} + \frac{1}{2}F^{ent}_{----} = \mathcal{V}. \tag{A.31}$$

Using this decomposition of the coefficients one can write $\sigma_2 = F^{bs}_{++++}|++++\rangle\langle\cdot| + F^{bs}_{----} + \sum_{ij} F_{+ij-}| + ij-\rangle\langle\cdot| + F_{-ij+}| - ij+\rangle\langle\cdot|$ as a convex combination of biseparable states of the form in Eq. (A.20). We can then split ϱ into a genuinely multiparticle entangled part and a biseparable rest $\sigma_1 + \sigma_2$

$$\varrho = F^{ent}_{++++}|++++\rangle\langle\cdot| + F^{ent}_{----}|----\rangle\langle\cdot| + \sigma_1 + \sigma_2. \tag{A.32}$$

In this case this yields the last expected estimate on the GMN

$$N_g(\varrho) \leq F^{ent}_{++++} N_g(|++++\rangle) + F^{ent}_{----} N_g(|----\rangle) = \mathcal{V} \tag{A.33}$$

So the maximal violation of the negative expectation values of the witnesses from Eqs. (2.27) and (2.28) gives upper bounds on the GMN, which proves the claim. As in the case of GHZ-diagonal states, the bound holds for both possible normalisations of the GMN. □

Appendix B

Expectation values of four-site operator basis in the XY-model

A list of all non-zero expectation values of the four-site Pauli operators $\langle\sigma_i^m\sigma_j^n\sigma_k^o\sigma_l^p\rangle_{|0\rangle}$ is presented here. The three site expectation values are omitted, since they are already contained in those of four sites. Note that there are 80 non-vanishing expectation values out of 256. This is due to several symmetries the Hamiltonian possesses, which carry over to its ground state. Namely the Hamiltonian is translationally invariant and thus just the spacings $\alpha = j - i$, $\beta = k - j$ and $\delta = k - l$ between the local sites enter the final expectation values. Furthermore, it is invariant under global x-flip $\sigma_i^x \to -\sigma_i^x$ for all i and under global y-flip. Hence the only contribution to the four site Pauli operators are those, in which σ^x and σ^y appear an even number of times. The exact numerical value finally depends on $\langle A_l B_k\rangle_{|0\rangle} = G_{k-l}$ only.
It is

$$\begin{aligned}\langle\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\rangle_{|0\rangle} &= 1,\\ \langle\sigma_i^z\mathbb{1}\mathbb{1}\mathbb{1}\rangle_{|0\rangle} &= -G_0,\\ \langle\sigma_i^z\sigma_j^z\mathbb{1}\mathbb{1}\rangle_{|0\rangle} &= G_0^2 - G_\alpha G_{-\alpha}.\end{aligned} \tag{B.1}$$

Similarly it is $\langle\mathbb{1}\sigma_j^z\mathbb{1}\mathbb{1}\rangle_{|0\rangle} = \langle\mathbb{1}\mathbb{1}\sigma_k^z\mathbb{1}\rangle_{|0\rangle} = \langle\mathbb{1}\mathbb{1}\mathbb{1}\sigma_l^z\rangle_{|0\rangle} = -G_0$. As for the tensor product of two σ^z operators $\langle\sigma_i^z\mathbb{1}\sigma_k^z\mathbb{1}\rangle_{|0\rangle} = G_0^2 - G_{\alpha+\beta}G_{-\alpha-\beta}$, $\langle\sigma_i^z\mathbb{1}\mathbb{1}\sigma_l^z\rangle_{|0\rangle} = G_0^2 - G_{\alpha+\beta+\delta}G_{-\alpha-\beta-\delta}$, $\langle\mathbb{1}\sigma_j^z\sigma_k^z\mathbb{1}\rangle_{|0\rangle} = G_0^2 - G_\beta G_{-\beta}$, $\langle\mathbb{1}\sigma_j^z\mathbb{1}\sigma_j^z\rangle_{|0\rangle} = G_0^2 - G_{\beta+\delta}G_{-\beta-\delta}$, $\langle\mathbb{1}\mathbb{1}\sigma_k^z\sigma_l^z\rangle_{|0\rangle} = G_0^2 - G_\delta G_{-\delta}$.

$$\langle\sigma_i^z\sigma_j^z\sigma_k^z\mathbb{1}\rangle_{|0\rangle} = -\begin{vmatrix} G_0 & G_\alpha & G_{\alpha+\beta}\\ G_{-\alpha} & G_0 & G_\beta\\ G_{-\alpha-\beta} & G_{-\beta} & G_0\end{vmatrix}. \tag{B.2}$$

To get the expectation values of the other possible permutations of the above operator one simply has to perform the following replacements (applied from left to right) $\langle\sigma_i^z\sigma_j^z\mathbb{1}\sigma_l^z\rangle_{|0\rangle} : \beta \to \beta+\delta$, $\langle\sigma_i^z\mathbb{1}\sigma_k^z\sigma_l^z\rangle_{|0\rangle} : \beta \to \delta \wedge \alpha \to \alpha+\beta$ and $\langle\mathbb{1}\sigma_j^z\sigma_k^z\sigma_l^z\rangle_{|0\rangle} : \beta \to \delta \wedge \alpha \to \beta$.

$$\langle\sigma_i^z\sigma_j^z\sigma_k^z\sigma_l^z\rangle_{|0\rangle} = \begin{vmatrix} G_0 & G_\alpha & G_{\alpha+\beta} & G_{\alpha+\beta+\delta}\\ G_{-\alpha} & G_0 & G_\beta & G_{\beta+\delta}\\ G_{-\alpha-\beta} & G_{-\beta} & G_0 & G_\delta\\ G_{-\alpha-\beta-\delta} & G_{-\beta-\delta} & G_{-\delta} & G_0\end{vmatrix}. \tag{B.3}$$

$$\langle\sigma_i^x\sigma_j^x\mathbb{1}\mathbb{1}\rangle_{|0\rangle} = (-1)^\alpha\begin{vmatrix} G_{-1} & \dots & G_{\alpha-2}\\ \vdots & & \vdots\\ G_{-\alpha} & \dots & G_{-1}\end{vmatrix}, \tag{B.4}$$

$\langle\sigma_i^x\mathbb{1}\sigma_k^x\mathbb{1}\rangle_{|0\rangle} : \alpha \to \alpha+\beta$, $\langle\sigma_i^x\mathbb{1}\mathbb{1}\sigma_l^x\rangle_{|0\rangle} : \alpha \to \alpha+\beta+\delta$, $\langle\mathbb{1}\sigma_j^x\sigma_k^x\mathbb{1}\rangle_{|0\rangle} : \alpha \to \beta$, $\langle\mathbb{1}\sigma_j^x\mathbb{1}\sigma_l^x\rangle_{|0\rangle} : \alpha \to \beta+\delta$, $\langle\mathbb{1}\mathbb{1}\sigma_k^x\sigma_l^x\rangle_{|0\rangle} : \alpha \to \delta$.

$$\langle\sigma_i^x\sigma_j^x\sigma_k^z\mathbb{1}\rangle_{|0\rangle} = (-1)^{\alpha+1}\begin{vmatrix} G_{-1} & \dots & G_{\alpha-2} & G_{\alpha+\beta-1} \\ \vdots & & \vdots & \vdots \\ G_{-\alpha} & \dots & G_{-1} & G_{\beta} \\ G_{-\alpha-\beta} & \dots & G_{-\beta-1} & G_0 \end{vmatrix}, \tag{B.5}$$

$\langle\sigma_i^x\sigma_j^x\mathbb{1}\sigma_l^z\rangle_{|0\rangle} : \beta \to \beta+\delta$, $\langle\sigma_i^x\mathbb{1}\sigma_k^x\sigma_l^z\rangle_{|0\rangle} : \beta \to \delta \wedge \alpha \to \alpha+\beta$, $\langle\mathbb{1}\sigma_j^x\sigma_k^x\sigma_l^z\rangle_{|0\rangle} : \beta \to \delta \wedge \alpha \to \beta$.

$$\langle\sigma_i^x\sigma_j^z\sigma_k^x\mathbb{1}\rangle_{|0\rangle} = (-1)^{\alpha+\beta}\begin{vmatrix} G_{-1} & \dots & G_{\alpha-2} & G_{\alpha} & \dots & G_{\alpha+\beta-2} \\ \vdots & & \vdots & \vdots & & \vdots \\ G_{-\alpha+1} & \dots & G_0 & G_2 & \dots & G_{\beta} \\ G_{-\alpha-1} & \dots & G_{-2} & G_0 & \dots & G_{\beta-2} \\ \vdots & & \vdots & \vdots & & \vdots \\ G_{-\alpha-\beta} & \dots & G_{-\beta-1} & G_{-\beta+1} & \dots & G_{-1} \end{vmatrix}, \tag{B.6}$$

$\langle\sigma_i^x\sigma_j^z\mathbb{1}\sigma_l^x\rangle_{|0\rangle} : \beta \to \beta+\delta$, $\langle\sigma_i^x\mathbb{1}\sigma_k^z\sigma_l^x\rangle_{|0\rangle} : \beta \to \delta \wedge \alpha \to \alpha+\beta$, $\langle\mathbb{1}\sigma_j^x\sigma_k^z\sigma_l^x\rangle_{|0\rangle} : \beta \to \delta \wedge \alpha \to \beta$.

$$\langle\sigma_i^z\sigma_j^x\sigma_k^x\mathbb{1}\rangle_{|0\rangle} = (-1)^{\beta+1}\begin{vmatrix} G_0 & G_{\alpha} & \dots & G_{\alpha+\beta-1} \\ G_{-\alpha-1} & G_{-1} & \dots & G_{\beta-2} \\ \vdots & \vdots & & \vdots \\ G_{-\alpha-\beta} & G_{-\beta} & \dots & G_{-1} \end{vmatrix}, \tag{B.7}$$

$\langle\sigma_i^z\sigma_j^x\mathbb{1}\sigma_l^x\rangle_{|0\rangle} : \beta \to \beta+\delta$, $\langle\sigma_i^z\mathbb{1}\sigma_k^x\sigma_l^x\rangle_{|0\rangle} : \beta \to \delta \wedge \alpha \to \alpha+\beta$, $\langle\mathbb{1}\sigma_j^z\sigma_k^x\sigma_l^x\rangle_{|0\rangle} : \beta \to \delta \wedge \alpha \to \beta$.

$$\langle\sigma_i^y\sigma_j^y\mathbb{1}\mathbb{1}\rangle_{|0\rangle} = (-1)^{\alpha}\begin{vmatrix} G_1 & \dots & G_{\alpha} \\ \vdots & & \vdots \\ G_{-\alpha+2} & \dots & G_1 \end{vmatrix}, \tag{B.8}$$

$\langle\sigma_i^y\mathbb{1}\sigma_k^y\mathbb{1}\rangle_{|0\rangle} : \alpha \to \alpha+\beta$, $\langle\sigma_i^y\mathbb{1}\mathbb{1}\sigma_l^y\rangle_{|0\rangle} : \alpha \to \alpha+\beta+\delta$, $\langle\mathbb{1}\sigma_j^y\sigma_k^y\mathbb{1}\rangle_{|0\rangle} : \alpha \to \beta$, $\langle\mathbb{1}\sigma_j^y\mathbb{1}\sigma_l^y\rangle_{|0\rangle} : \alpha \to \beta+\delta$, $\langle\mathbb{1}\mathbb{1}\sigma_k^y\sigma_l^y\rangle_{|0\rangle} : \alpha \to \delta$.

$$\langle\sigma_i^y\sigma_j^y\sigma_k^z\mathbb{1}\rangle_{|0\rangle} = (-1)^{\alpha+1}\begin{vmatrix} G_1 & \dots & G_{\alpha} & G_{\alpha+\beta} \\ \vdots & & \vdots & \vdots \\ G_{-\alpha+2} & \dots & G_1 & G_{\beta+1} \\ G_{-\alpha-\beta+1} & \dots & G_{-\beta} & G_0 \end{vmatrix}, \tag{B.9}$$

$\langle\sigma_i^y\sigma_j^y\mathbb{1}\sigma_l^z\rangle_{|0\rangle} : \beta \to \beta+\delta$, $\langle\sigma_i^y\mathbb{1}\sigma_k^y\sigma_l^z\rangle_{|0\rangle} : \beta \to \delta \wedge \alpha \to \alpha+\beta$, $\langle\mathbb{1}\sigma_j^y\sigma_k^y\sigma_l^z\rangle_{|0\rangle} : \beta \to \delta \wedge \alpha \to \beta$.

$$\langle\sigma_i^y\sigma_j^z\sigma_k^y\mathbb{1}\rangle_{|0\rangle} = (-1)^{\alpha+\beta}\begin{vmatrix} G_1 & \dots & G_{\alpha-1} & G_{\alpha+1} & \dots & G_{\alpha+\beta} \\ \vdots & & \vdots & \vdots & & \vdots \\ G_{-\alpha+2} & \dots & G_0 & G_2 & \dots & G_{\beta+1} \\ G_{-\alpha} & \dots & G_{-2} & G_0 & \dots & G_{\beta-1} \\ \vdots & & \vdots & \vdots & & \vdots \\ G_{-\alpha-\beta+2} & \dots & G_{-\beta} & G_{-\beta+2} & \dots & G_1 \end{vmatrix}, \tag{B.10}$$

$\langle\sigma_i^y\sigma_j^z\mathbb{1}\sigma_l^y\rangle_{|0\rangle} : \beta \to \beta+\delta$, $\langle\sigma_i^y\mathbb{1}\sigma_k^z\sigma_l^y\rangle_{|0\rangle} : \beta \to \delta \wedge \alpha \to \alpha+\beta$, $\langle\mathbb{1}\sigma_j^y\sigma_k^z\sigma_l^y\rangle_{|0\rangle} : \beta \to \delta \wedge \alpha \to \beta$.

$$\langle\sigma_i^z\sigma_j^y\sigma_k^y\mathbb{1}\rangle_{|0\rangle} = (-1)^{\beta+1}\begin{vmatrix} G_0 & G_{\alpha+1} & \dots & G_{\alpha+\beta} \\ G_{-\alpha} & G_1 & \dots & G_{\beta} \\ \vdots & \vdots & & \vdots \\ G_{-\alpha-\beta+1} & G_{-\beta+2} & \dots & G_1 \end{vmatrix}, \tag{B.11}$$

$\langle\sigma_i^z\sigma_j^y\mathbb{1}\sigma_l^y\rangle_{|0\rangle} : \beta \to \beta+\delta$, $\langle\sigma_i^z\mathbb{1}\sigma_k^y\sigma_l^y\rangle_{|0\rangle} : \beta \to \delta \wedge \alpha \to \alpha+\beta$, $\langle\mathbb{1}\sigma_j^z\sigma_k^y\sigma_l^y\rangle_{|0\rangle} : \beta \to \delta \wedge \alpha \to \beta$.

$$\langle\sigma_i^x\sigma_j^x\sigma_k^z\sigma_l^z\rangle_{|0\rangle} = (-1)^{\alpha}\begin{vmatrix} G_{-1} & ... & G_{\alpha-2} & G_{\alpha+\beta-1} & G_{\alpha+\beta+\delta-1} \\ \vdots & & \vdots & \vdots & \vdots \\ G_{-\alpha} & ... & G_{-1} & G_{\beta} & G_{\beta+\delta} \\ G_{-\alpha-\beta} & ... & G_{-\beta-1} & G_0 & G_{\delta} \\ G_{-\alpha-\beta-\delta} & ... & G_{-\beta-\delta-1} & G_{-\delta} & G_0 \end{vmatrix} \tag{B.12}$$

$$\langle\sigma_i^z\sigma_j^z\sigma_k^x\sigma_l^x\rangle_{|0\rangle} = (-1)^{\delta}\begin{vmatrix} G_0 & G_{\alpha} & G_{\alpha+\beta} & ... & G_{\alpha+\beta+\delta-1} \\ G_{-\alpha} & G_0 & G_{\beta} & ... & G_{\beta+\delta-1} \\ G_{-\alpha-\beta-1} & G_{-\beta-1} & G_{-1} & ... & G_{\delta-2} \\ \vdots & \vdots & \vdots & & \vdots \\ G_{-\alpha-\beta-\delta} & G_{-\beta-\delta} & G_{-\delta} & ... & G_{-1} \end{vmatrix} \tag{B.13}$$

$$\langle\sigma_i^z\sigma_j^x\sigma_k^x\sigma_l^z\rangle_{|0\rangle} = (-1)^{\beta}\begin{vmatrix} G_0 & G_{\alpha} & ... & G_{\alpha+\beta-1} & G_{\alpha+\beta+\delta} \\ G_{-\alpha-1} & G_{-1} & ... & G_{\beta-2} & G_{\beta+\delta-1} \\ \vdots & \vdots & & \vdots & \vdots \\ G_{-\alpha-\beta} & G_{-\beta} & ... & G_{-1} & G_{\delta} \\ G_{-\alpha-\beta-\delta} & G_{-\beta-\delta} & ... & G_{-\delta-1} & G_0 \end{vmatrix} \tag{B.14}$$

$$\langle\sigma_i^x\sigma_j^z\sigma_k^z\sigma_l^x\rangle_{|0\rangle} =$$

$$(-1)^{\alpha+\beta+\delta}\begin{vmatrix} G_{-1} & ... & G_{\alpha-2} & G_{\alpha} & ... & G_{\alpha+\beta-2} & G_{\alpha+\beta} & ... & G_{\alpha+\beta+\delta-2} \\ \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ G_{-\alpha+1} & ... & G_0 & G_2 & ... & G_{\beta} & G_{\beta+2} & ... & G_{\beta+\delta} \\ G_{-\alpha-1} & ... & G_{-2} & G_0 & ... & G_{\beta-2} & G_{\beta} & ... & G_{\beta+\delta-2} \\ \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ G_{-\alpha-\beta+1} & ... & G_{-\beta} & G_{-\beta+2} & ... & G_0 & G_2 & ... & G_{\delta} \\ G_{-\alpha-\beta-1} & ... & G_{-\beta-2} & G_{-\beta} & ... & G_{-2} & G_0 & ... & G_{\delta-2} \\ \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ G_{-\alpha-\beta-\delta} & ... & G_{-\beta-\delta-1} & G_{-\beta-\delta+1} & ... & G_{-\delta-1} & G_{-\delta+1} & ... & G_{-1} \end{vmatrix} \tag{B.15}$$

$$\langle\sigma_i^y\sigma_j^y\sigma_k^z\sigma_l^z\rangle_{|0\rangle} = (-1)^{\alpha}\begin{vmatrix} G_1 & ... & G_{\alpha} & G_{\alpha+\beta} & G_{\alpha+\beta+\delta} \\ \vdots & & \vdots & \vdots & \vdots \\ G_{-\alpha+2} & ... & G_1 & G_{\beta+1} & G_{\beta+\delta+1} \\ G_{-\alpha-\beta+1} & ... & G_{-\beta} & G_0 & G_{\delta} \\ G_{-\alpha-\beta-\delta+1} & ... & G_{-\beta-\delta} & G_{-\delta} & G_0 \end{vmatrix} \tag{B.16}$$

$$\langle\sigma_i^z\sigma_j^z\sigma_k^y\sigma_l^y\rangle_{|0\rangle} = (-1)^{\delta}\begin{vmatrix} G_0 & G_{\alpha} & G_{\alpha+\beta+1} & ... & G_{\alpha+\beta+\delta} \\ G_{-\alpha} & G_0 & G_{\beta+1} & ... & G_{\beta+\delta} \\ G_{-\alpha-\beta} & G_{-\beta} & G_1 & ... & G_{\delta} \\ \vdots & \vdots & \vdots & & \vdots \\ G_{-\alpha-\beta-\delta+1} & G_{-\beta-\delta+1} & G_{-\delta+2} & ... & G_1 \end{vmatrix} \tag{B.17}$$

$$\langle\sigma_i^z\sigma_j^y\sigma_k^y\sigma_l^z\rangle_{|0\rangle} = (-1)^{\beta}\begin{vmatrix} G_0 & G_{\alpha+1} & ... & G_{\alpha+\beta} & G_{\alpha+\beta+\delta} \\ G_{-\alpha} & G_1 & ... & G_{\beta} & G_{\beta+\delta} \\ \vdots & \vdots & & \vdots & \vdots \\ G_{-\alpha-\beta+1} & G_{-\beta+2} & ... & G_1 & G_{\delta+1} \\ G_{-\alpha-\beta-\delta} & G_{-\beta-\delta+1} & ... & G_{-\delta} & G_0 \end{vmatrix} \tag{B.18}$$

$$\langle\sigma_i^y\sigma_j^z\sigma_k^z\sigma_l^y\rangle_{|0\rangle} = (-1)^{\alpha+\beta+\delta}\begin{vmatrix} G_1 & \dots & G_{\alpha-1} & G_{\alpha+1} & \dots & G_{\alpha+\beta-1} & G_{\alpha+\beta+1} & \dots & G_{\alpha+\beta+\delta} \\ \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ G_{-\alpha+2} & \dots & G_0 & G_2 & \dots & G_\beta & G_{\beta+2} & \dots & G_{\beta+\delta+1} \\ G_{-\alpha} & \dots & G_{-2} & G_0 & \dots & G_{\beta-2} & G_\beta & \dots & G_{\beta+\delta-1} \\ \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ G_{-\alpha-\beta+2} & \dots & G_{-\beta} & G_{-\beta+2} & \dots & G_0 & G_2 & \dots & G_{\delta+1} \\ G_{-\alpha-\beta} & \dots & G_{-\beta-2} & G_{-\beta} & \dots & G_{-2} & G_0 & \dots & G_{\delta-1} \\ \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ G_{-\alpha-\beta-\delta+2} & \dots & G_{-\beta-\delta} & G_{-\beta-\delta+2} & \dots & G_{-\delta} & G_{-\delta+2} & \dots & G_1 \end{vmatrix} \tag{B.19}$$

$$\langle\sigma_i^x\sigma_j^z\sigma_k^x\sigma_l^z\rangle_{|0\rangle} = (-1)^{\alpha+\beta+1}\begin{vmatrix} G_{-1} & \dots & G_{\alpha-2} & G_\alpha & \dots & G_{\alpha+\beta-2} & G_{\alpha+\beta+\delta-1} \\ \vdots & & \vdots & \vdots & & \vdots & \vdots \\ G_{-\alpha+1} & \dots & G_0 & G_2 & \dots & G_\beta & G_{\beta+\delta+1} \\ G_{-\alpha-1} & \dots & G_{-2} & G_0 & \dots & G_{\beta-2} & G_{\beta+\delta-1} \\ \vdots & & \vdots & \vdots & & \vdots & \vdots \\ G_{-\alpha-\beta} & \dots & G_{-\beta-1} & G_{-\beta+1} & \dots & G_{-1} & G_\delta \\ G_{-\alpha-\beta-\delta} & \dots & G_{-\beta-\delta-1} & G_{-\beta-\delta+1} & \dots & G_{-\delta-1} & G_0 \end{vmatrix} \tag{B.20}$$

$$\langle\sigma_i^z\sigma_j^x\sigma_k^z\sigma_l^x\rangle_{|0\rangle} = (-1)^{\beta+\delta+1}\begin{vmatrix} G_0 & G_\alpha & \dots & G_{\alpha+\beta-1} & G_{\alpha+\beta+1} & \dots & G_{\alpha+\beta+\delta-1} \\ G_{-\alpha-1} & G_{-1} & \dots & G_{\beta-2} & G_\beta & \dots & G_{\beta+\delta-2} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ G_{-\alpha-\beta+1} & G_{-\beta+1} & \dots & G_0 & G_2 & \dots & G_\delta \\ G_{-\alpha-\beta-1} & G_{-\beta-1} & \dots & G_{-2} & G_0 & \dots & G_{\delta-2} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ G_{-\alpha-\beta-\delta} & G_{-\beta-\delta} & \dots & G_{-\delta-1} & G_{-\delta+1} & \dots & G_{-1} \end{vmatrix} \tag{B.21}$$

$$\langle\sigma_i^y\sigma_j^z\sigma_k^y\sigma_l^z\rangle_{|0\rangle} = (-1)^{\alpha+\beta+1}\begin{vmatrix} G_1 & \dots & G_{\alpha-1} & G_{\alpha+1} & \dots & G_{\alpha+\beta} & G_{\alpha+\beta+\delta} \\ \vdots & & \vdots & \vdots & & \vdots & \vdots \\ G_{-\alpha+2} & \dots & G_0 & G_2 & \dots & G_{\beta+1} & G_{\beta+\delta+1} \\ G_{-\alpha} & \dots & G_{-2} & G_0 & \dots & G_{\beta-1} & G_{\beta+\delta-1} \\ \vdots & & \vdots & \vdots & & \vdots & \vdots \\ G_{-\alpha-\beta+2} & \dots & G_{-\beta} & G_{-\beta+2} & \dots & G_1 & G_{\delta+1} \\ G_{-\alpha-\beta-\delta+1} & \dots & G_{-\beta-\delta-1} & G_{-\beta-\delta+1} & \dots & G_{-\delta} & G_0 \end{vmatrix} \tag{B.22}$$

$$\langle\sigma_i^z\sigma_j^y\sigma_k^z\sigma_l^y\rangle_{|0\rangle} = (-1)^{\beta+\delta+1}\begin{vmatrix} G_0 & G_{\alpha+1} & \dots & G_{\alpha+\beta-1} & G_{\alpha+\beta+1} & \dots & G_{\alpha+\beta+\delta} \\ G_{-\alpha} & G_1 & \dots & G_{\beta-1} & G_{\beta+1} & \dots & G_{\beta+\delta} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ G_{-\alpha-\beta+1} & G_{-\beta+2} & \dots & G_0 & G_2 & \dots & G_{\delta+1} \\ G_{-\alpha-\beta-1} & G_{-\beta} & \dots & G_{-2} & G_0 & \dots & G_{\delta-1} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ G_{-\alpha-\beta-\delta+1} & G_{-\beta-\delta-2} & \dots & G_{-\delta} & G_{-\delta+2} & \dots & G_1 \end{vmatrix} \tag{B.23}$$

$$\langle\sigma_i^x\sigma_j^x\sigma_k^y\sigma_l^y\rangle_{|0\rangle} = (-1)^{\alpha+\delta}\begin{vmatrix} G_{-1} & \dots & G_{\alpha-2} & G_{\alpha+\beta} & \dots & G_{\alpha+\beta+\delta-1} \\ \vdots & & \vdots & \vdots & & \vdots \\ G_{-\alpha} & \dots & G_{-1} & G_{\beta+1} & \dots & G_{\beta+\delta} \\ G_{-\alpha-\beta} & \dots & G_{-\beta-1} & G_1 & \dots & G_\delta \\ \vdots & & \vdots & \vdots & & \vdots \\ G_{-\alpha-\beta-\delta+1} & \dots & G_{-\beta-\delta} & G_{-\delta+2} & \dots & G_1 \end{vmatrix} \tag{B.24}$$

$$\langle\sigma_i^y\sigma_j^y\sigma_k^x\sigma_l^x\rangle_{|0\rangle} = (-1)^{\alpha+\delta}\begin{vmatrix} G_1 & \dots & G_\alpha & G_{\alpha+\beta} & \dots & G_{\alpha+\beta+\delta-1} \\ \vdots & & \vdots & \vdots & & \vdots \\ G_{-\alpha+2} & \dots & G_1 & G_{\beta+1} & \dots & G_{\beta+\delta} \\ G_{-\alpha-\beta} & \dots & G_{-\beta-1} & G_{-1} & \dots & G_{\delta-2} \\ \vdots & & \vdots & \vdots & & \vdots \\ G_{-\alpha-\beta-\delta+1} & \dots & G_{-\beta-\delta} & G_{-\delta} & \dots & G_{-1} \end{vmatrix} \tag{B.25}$$

$$\langle\sigma_i^x\sigma_j^y\sigma_k^y\sigma_l^x\rangle_{|0\rangle} = (-1)^{\alpha+\delta}\begin{vmatrix} G_{-1} & \dots & G_{\alpha-1} & G_{\alpha+\beta} & \dots & G_{\alpha+\beta+\delta-2} \\ \vdots & & \vdots & \vdots & & \vdots \\ G_{-\alpha+1} & \dots & G_1 & G_{\beta+2} & \dots & G_{\beta+\delta} \\ G_{-\alpha-\beta} & \dots & G_{-\beta} & G_1 & \dots & G_{\delta-1} \\ \vdots & & \vdots & \vdots & & \vdots \\ G_{-\alpha-\beta-\delta} & \dots & G_{-\beta-\delta} & G_{-\delta+1} & \dots & G_{-1} \end{vmatrix} \tag{B.26}$$

$$\langle\sigma_i^y\sigma_j^x\sigma_k^x\sigma_l^y\rangle_{|0\rangle} = (-1)^{\alpha+\delta}\begin{vmatrix} G_1 & \dots & G_{\alpha-1} & G_{\alpha+\beta} & \dots & G_{\alpha+\beta+\delta} \\ \vdots & & \vdots & \vdots & & \vdots \\ G_{-\alpha+1} & \dots & G_{-1} & G_\beta & \dots & G_{\beta+\delta} \\ G_{-\alpha-\beta} & \dots & G_{-\beta-2} & G_{-1} & \dots & G_{\delta-1} \\ \vdots & & \vdots & \vdots & & \vdots \\ G_{-\alpha-\beta-\delta+2} & \dots & G_{-\beta-\delta} & G_{-\delta+1} & \dots & G_1 \end{vmatrix} \tag{B.27}$$

$$\langle\sigma_i^x\sigma_j^x\sigma_k^x\sigma_l^x\rangle_{|0\rangle} = (-1)^{\alpha+\delta}\begin{vmatrix} G_{-1} & \dots & G_{\alpha-2} & G_{\alpha+\beta-1} & \dots & G_{\alpha+\beta+\delta-2} \\ \vdots & & \vdots & \vdots & & \vdots \\ G_{-\alpha} & \dots & G_{-1} & G_\beta & \dots & G_{\beta+\delta-1} \\ G_{-\alpha-\beta-1} & \dots & G_{-\beta-2} & G_{-1} & \dots & G_{\delta-2} \\ \vdots & & \vdots & \vdots & & \vdots \\ G_{-\alpha-\beta-\delta} & \dots & G_{-\beta-\delta-1} & G_{-\delta} & \dots & G_{-1} \end{vmatrix} \tag{B.28}$$

$$\langle\sigma_i^y\sigma_j^y\sigma_k^y\sigma_l^y\rangle_{|0\rangle} = (-1)^{\alpha+\delta}\begin{vmatrix} G_1 & \dots & G_\alpha & G_{\alpha+\beta+1} & \dots & G_{\alpha+\beta+\delta} \\ \vdots & & \vdots & \vdots & & \vdots \\ G_{-\alpha+2} & \dots & G_1 & G_{\beta+2} & \dots & G_{\beta+\delta+1} \\ G_{-\alpha-\beta+1} & \dots & G_{-\beta} & G_1 & \dots & G_\delta \\ \vdots & & \vdots & \vdots & & \vdots \\ G_{-\alpha-\beta-\delta+2} & \dots & G_{-\beta-\delta+1} & G_{-\delta+2} & \dots & G_1 \end{vmatrix} \tag{B.29}$$

Appendix C

LC-classification of stabiliser subgroups

Here all LC and qubit-permutational inequivalent classes of stabiliser subgroups up to six qubits are given. Each class may contain several equivalent subgroups. Here, one representative of each is given.

C.1 Two qubits

In the case of two qubits there is one LC-inequivalent graph with generators:

1. $\{XZ, ZX\}$.

In the case subgroups with one generator are considered there are only one class:

1. $\{\mathbb{1}\mathbb{1}, XZ\}$

C.2 Three qubits

In the case of three qubits there is one LC-inequivalent graph with generators:

1. $\{XZ\mathbb{1}, ZXZ, \mathbb{1}ZX\}$.

In the case subgroups with one generator are considered there are two classes:

1. $\{\mathbb{1}\mathbb{1}\mathbb{1}, XZ\mathbb{1}\}$
2. $\{\mathbb{1}\mathbb{1}\mathbb{1}, YYZ\}$

In case subgroups with two generators are considered there are two classes:

1. $\{\mathbb{1}\mathbb{1}\mathbb{1}, XZ\mathbb{1}, YYZ, ZXZ\}$
2. $\{\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}ZX, X\mathbb{1}X, XZ\mathbb{1}\}$

C.3 Four qubits

In the case of four qubits there are two LC-inequivalent graphs with generators:

1. $\{XZZZ, ZX\mathbb{1}\mathbb{1}, Z\mathbb{1}X\mathbb{1}, Z\mathbb{1}\mathbb{1}X\}$,
2. $\{XZ\mathbb{1}\mathbb{1}, ZXZ\mathbb{1}, \mathbb{1}ZXZ, \mathbb{1}\mathbb{1}ZX\}$.

In case subgroups with one generator are considered there are three classes:

1. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, XZZZ\}$
2. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}XX\mathbb{1}\}$
3. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, X\mathbb{1}XZ\}$

In case subgroups with two generators are considered there are six classes:

1. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}X\mathbb{1}X, YYZZ, YZZY\}$
2. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, XZZZ, YYYY, ZXXX\}$
3. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}X\mathbb{1}X, Z\mathbb{1}X\mathbb{1}, ZXXX\}$
4. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}ZX, X\mathbb{1}XZ, X\mathbb{1}YY\}$
5. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, XZZX, YY\mathbb{1}X, ZXZ\mathbb{1}\}$
6. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}ZXZ, X\mathbb{1}XZ, XZ\mathbb{1}\mathbb{1}\}$

In case subgroups with three generators are considered there are six classes:

1. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}XX\mathbb{1}, XYYZ, XZZZ, YYZZ, YZYZ, Z\mathbb{1}X\mathbb{1}, ZX\mathbb{1}\mathbb{1}\}$
2. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}X\mathbb{1}X, XYZY, XZZZ, YYYY, YZYZ, Z\mathbb{1}X\mathbb{1}, ZXXX\}$
3. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}XX, \mathbb{1}X\mathbb{1}X, \mathbb{1}XX\mathbb{1}, Z\mathbb{1}\mathbb{1}X, Z\mathbb{1}X\mathbb{1}, ZX\mathbb{1}\mathbb{1}, ZXXX\}$
4. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}ZXZ, X\mathbb{1}XZ, XZ\mathbb{1}\mathbb{1}, YXYZ, YYZ\mathbb{1}, ZXZ\mathbb{1}, ZYYZ\}$
5. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}ZXZ, X\mathbb{1}YY, XZZX, YXXY, YY\mathbb{1}X, ZXZ\mathbb{1}, ZYYZ\}$
6. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}ZX, XZ\mathbb{1}\mathbb{1}, XZZX, YY\mathbb{1}X, YYZ\mathbb{1}, ZX\mathbb{1}X, ZXZ\mathbb{1}\}$

C.4 Five qubits

In the case of five qubits there are four LC-inequivalent graphs with generators:

1. $\{XZZZZ, ZX\mathbb{1}\mathbb{1}\mathbb{1}, Z\mathbb{1}X\mathbb{1}\mathbb{1}, Z\mathbb{1}\mathbb{1}X\mathbb{1}, Z\mathbb{1}\mathbb{1}\mathbb{1}X\}$,
2. $\{XZ\mathbb{1}\mathbb{1}\mathbb{1}, ZXZ\mathbb{1}\mathbb{1}, \mathbb{1}ZXZ\mathbb{1}, \mathbb{1}\mathbb{1}ZXZ, \mathbb{1}\mathbb{1}\mathbb{1}ZX\}$,
3. $\{XZ\mathbb{1}\mathbb{1}\mathbb{1}, ZXZ\mathbb{1}Z, \mathbb{1}ZXZ\mathbb{1}, \mathbb{1}\mathbb{1}ZX\mathbb{1}, \mathbb{1}Z\mathbb{1}\mathbb{1}X\}$,
4. $\{XZ\mathbb{1}\mathbb{1}Z, ZXZ\mathbb{1}\mathbb{1}, \mathbb{1}ZXZ\mathbb{1}, \mathbb{1}\mathbb{1}ZXZ, Z\mathbb{1}\mathbb{1}ZX\}$.

In case subgroups with one generator are considered there are six classes:

1. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, XXXXX\}$
2. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}\mathbb{1}XX\}$
3. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}XXXX\}$
4. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, X\mathbb{1}X\mathbb{1}X\}$
5. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, XZZZZ, YYZZZ, ZX\mathbb{1}\mathbb{1}\mathbb{1}\}$
6. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, XZZZZ, YYZYY, ZX\mathbb{1}XX\}$

In case subgroups with two generators are considered there are 12 classes:

1. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}X\mathbb{1}X, \mathbb{1}X\mathbb{1}\mathbb{1}X, \mathbb{1}XX\mathbb{1}\mathbb{1}\}$
2. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}XX\mathbb{1}, \mathbb{1}X\mathbb{1}\mathbb{1}X, \mathbb{1}XXXX\}$

3. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}XXXX, Z\mathbb{1}\mathbb{1}\mathbb{1}X, ZXXX\mathbb{1}\}$
4. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}ZX\mathbb{1}, X\mathbb{1}XZ\mathbb{1}, X\mathbb{1}YY\mathbb{1}\}$
5. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, XXXXX, YY\mathbb{1}X\mathbb{1}, ZZX\mathbb{1}X\}$
6. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, X\mathbb{1}\mathbb{1}\mathbb{1}X, YYZ\mathbb{1}Z, ZYZ\mathbb{1}Y\}$
7. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, XZZX\mathbb{1}, YXXY\mathbb{1}, ZYYZ\mathbb{1}\}$
8. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}XZX, X\mathbb{1}YY\mathbb{1}, X\mathbb{1}ZXX\}$
9. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, XXY\mathbb{1}Y, Y\mathbb{1}YXX, ZX\mathbb{1}XZ\}$
10. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}\mathbb{1}ZX, X\mathbb{1}X\mathbb{1}X, X\mathbb{1}XZ\mathbb{1}\}$
11. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}ZXZ\mathbb{1}, X\mathbb{1}X\mathbb{1}X, XZ\mathbb{1}ZX\}$
12. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}ZXZ\mathbb{1}, X\mathbb{1}\mathbb{1}\mathbb{1}X, XZXZX\}$

In case subgroups with three generators are considered there are 26 classes:

1. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}XX\mathbb{1}\mathbb{1}, XYYZZ, XZZZZ, YYZZZ, YZYZZ, Z\mathbb{1}X\mathbb{1}\mathbb{1}, ZX\mathbb{1}\mathbb{1}\mathbb{1}\}$
2. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}\mathbb{1}XX, \mathbb{1}XX\mathbb{1}\mathbb{1}, \mathbb{1}XXXX, YYYYZ, YYYZY, YZZYZ, YZZZY\}$
3. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}XXXX, XYZYZ, XZYZY, YYYYZ, YZZZY, Z\mathbb{1}X\mathbb{1}\mathbb{1}, ZX\mathbb{1}XX\}$
4. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}XX\mathbb{1}, \mathbb{1}X\mathbb{1}X\mathbb{1}, \mathbb{1}XX\mathbb{1}\mathbb{1}, Z\mathbb{1}\mathbb{1}X\mathbb{1}, Z\mathbb{1}X\mathbb{1}\mathbb{1}, ZX\mathbb{1}\mathbb{1}\mathbb{1}, ZXXX\mathbb{1}\}$
5. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}X\mathbb{1}X, \mathbb{1}X\mathbb{1}X\mathbb{1}, \mathbb{1}XXXX, Z\mathbb{1}\mathbb{1}X\mathbb{1}, Z\mathbb{1}XXX, ZX\mathbb{1}\mathbb{1}\mathbb{1}, ZXX\mathbb{1}X\}$
6. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}ZXZ\mathbb{1}, X\mathbb{1}XZ\mathbb{1}, XZ\mathbb{1}\mathbb{1}\mathbb{1}, YXYZ\mathbb{1}, YYZ\mathbb{1}\mathbb{1}, ZXZ\mathbb{1}\mathbb{1}, ZYYZ\mathbb{1}\}$
7. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}XZZX, \mathbb{1}YY\mathbb{1}X, \mathbb{1}ZXZ\mathbb{1}, Y\mathbb{1}YXX, YXXY\mathbb{1}, YY\mathbb{1}X\mathbb{1}, YZZYX\}$
8. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}ZX\mathbb{1}X, X\mathbb{1}X\mathbb{1}X, XZ\mathbb{1}\mathbb{1}\mathbb{1}, YXXXY, YY\mathbb{1}XZ, ZX\mathbb{1}XZ, ZYXXY\}$
9. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}XZZX, X\mathbb{1}YY\mathbb{1}, XXXXX, YY\mathbb{1}X\mathbb{1}, YZZYX, ZYYZ\mathbb{1}, ZZX\mathbb{1}X\}$
10. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}ZXZ\mathbb{1}, X\mathbb{1}YYZ, XZZXZ, YXXXY, YY\mathbb{1}YY, ZXZZX, ZYY\mathbb{1}X\}$
11. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}X\mathbb{1}YY, X\mathbb{1}XZZ, XXXXX, Y\mathbb{1}YXX, YXYZZ, Z\mathbb{1}ZYY, ZXZ\mathbb{1}\mathbb{1}\}$
12. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}ZYYZ, X\mathbb{1}YYZ, XZ\mathbb{1}\mathbb{1}\mathbb{1}, YXXYZ, YYZ\mathbb{1}\mathbb{1}, ZXZ\mathbb{1}\mathbb{1}, ZYXYZ\}$
13. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}ZYYZ, X\mathbb{1}YYZ, XZ\mathbb{1}\mathbb{1}\mathbb{1}, YXYZ\mathbb{1}, YY\mathbb{1}XZ, ZX\mathbb{1}XZ, ZYYZ\mathbb{1}\}$
14. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}XZZX, XY\mathbb{1}YX, XZZX\mathbb{1}, Y\mathbb{1}YXX, YXXY\mathbb{1}, ZYYZ\mathbb{1}, ZZX\mathbb{1}X\}$
15. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}ZXZ, XZ\mathbb{1}\mathbb{1}\mathbb{1}, XZZXZ, YY\mathbb{1}XZ, YYZ\mathbb{1}\mathbb{1}, ZX\mathbb{1}XZ, ZXZ\mathbb{1}\mathbb{1}\}$
16. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}ZYY, XZ\mathbb{1}\mathbb{1}\mathbb{1}, XZZYY, YXXXY, YXYZ\mathbb{1}, ZYXXY, ZYYZ\mathbb{1}\}$
17. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}ZXZ, \mathbb{1}ZX\mathbb{1}X, \mathbb{1}ZYXY, X\mathbb{1}X\mathbb{1}X, X\mathbb{1}YXY, XZ\mathbb{1}\mathbb{1}\mathbb{1}, XZZXZ\}$
18. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}\mathbb{1}ZX, XZ\mathbb{1}\mathbb{1}\mathbb{1}, XZ\mathbb{1}ZX, YYZ\mathbb{1}\mathbb{1}, YYZZX, ZXZ\mathbb{1}\mathbb{1}, ZXZZX\}$
19. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}\mathbb{1}ZX, XZ\mathbb{1}\mathbb{1}\mathbb{1}, XZ\mathbb{1}ZX, YY\mathbb{1}XZ, YY\mathbb{1}YY, ZX\mathbb{1}XZ, ZX\mathbb{1}YY\}$
20. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}ZX\mathbb{1}, X\mathbb{1}\mathbb{1}\mathbb{1}X, X\mathbb{1}ZXX, YX\mathbb{1}XY, YXZ\mathbb{1}Y, ZX\mathbb{1}XZ, ZXZ\mathbb{1}Z\}$
21. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}\mathbb{1}ZX, \mathbb{1}ZX\mathbb{1}X, \mathbb{1}ZXZ\mathbb{1}, X\mathbb{1}X\mathbb{1}X, X\mathbb{1}XZ\mathbb{1}, XZ\mathbb{1}\mathbb{1}\mathbb{1}, XZ\mathbb{1}ZX\}$
22. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}ZX\mathbb{1}, \mathbb{1}ZXZ\mathbb{1}, \mathbb{1}ZYY\mathbb{1}, X\mathbb{1}XZ\mathbb{1}, X\mathbb{1}YY\mathbb{1}, XZ\mathbb{1}\mathbb{1}\mathbb{1}, XZZX\mathbb{1}\}$
23. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}Z\mathbb{1}\mathbb{1}X, X\mathbb{1}\mathbb{1}\mathbb{1}X, XZ\mathbb{1}\mathbb{1}\mathbb{1}, YXZ\mathbb{1}Y, YYZ\mathbb{1}Z, ZXZ\mathbb{1}Z, ZYZ\mathbb{1}Y\}$
24. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}XZX, \mathbb{1}Z\mathbb{1}\mathbb{1}X, \mathbb{1}ZXZ\mathbb{1}, X\mathbb{1}\mathbb{1}\mathbb{1}X, X\mathbb{1}XZ\mathbb{1}, XZ\mathbb{1}\mathbb{1}\mathbb{1}, XZXZX\}$
25. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}YY\mathbb{1}X, XXY\mathbb{1}Y, XZ\mathbb{1}\mathbb{1}Z, Y\mathbb{1}X\mathbb{1}Y, YYZ\mathbb{1}Z, ZXZ\mathbb{1}\mathbb{1}, ZZX\mathbb{1}X\}$
26. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}X\mathbb{1}YY, \mathbb{1}YXXY, \mathbb{1}ZXZ\mathbb{1}, Y\mathbb{1}X\mathbb{1}Y, YXXY\mathbb{1}, YY\mathbb{1}X\mathbb{1}, YZ\mathbb{1}ZY\}$

In case subgroups with four generators are considered there are 17 classes:

1. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}X\mathbb{1}\mathbb{1}, \mathbb{1}X\mathbb{1}X\mathbb{1}, \mathbb{1}X\mathbb{1}\mathbb{1}\mathbb{1}, XYYZZ, XYZYZ, XZYYZ, XZZZZ, YYYYZ, YYZZZ, YZYZZ, YZZYZ, Z\mathbb{1}\mathbb{1}X\mathbb{1}, Z\mathbb{1}X\mathbb{1}\mathbb{1}, ZX\mathbb{1}\mathbb{1}\mathbb{1}, ZXXX\mathbb{1}\}$
2. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}X\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}X\mathbb{1}X, \mathbb{1}\mathbb{1}XXX, XYYYY, XYZZY, XZYYZ, XZZZZ, YYYZY, YYZYY, YZYZZ, YZZYZ, Z\mathbb{1}\mathbb{1}X\mathbb{1}, Z\mathbb{1}X\mathbb{1}\mathbb{1}, ZX\mathbb{1}XX, ZXX\mathbb{1}X\}$
3. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}\mathbb{1}XX, \mathbb{1}\mathbb{1}X\mathbb{1}X, \mathbb{1}\mathbb{1}XX\mathbb{1}, \mathbb{1}X\mathbb{1}\mathbb{1}X, \mathbb{1}X\mathbb{1}X\mathbb{1}, \mathbb{1}XX\mathbb{1}\mathbb{1}, \mathbb{1}XXXX, Z\mathbb{1}\mathbb{1}\mathbb{1}X, Z\mathbb{1}\mathbb{1}X\mathbb{1}, Z\mathbb{1}X\mathbb{1}\mathbb{1}, Z\mathbb{1}XXX, ZX\mathbb{1}\mathbb{1}\mathbb{1}, ZX\mathbb{1}XX, ZXX\mathbb{1}X, ZXXX\mathbb{1}\}$
4. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}ZXZ, \mathbb{1}ZXZ\mathbb{1}, \mathbb{1}ZYYZ, X\mathbb{1}XZ\mathbb{1}, X\mathbb{1}YYZ, XZ\mathbb{1}\mathbb{1}\mathbb{1}, XZZXZ, YXXYZ, YXYZ\mathbb{1}, YY\mathbb{1}XZ, YYZ\mathbb{1}\mathbb{1}, ZX\mathbb{1}XZ, ZXZ\mathbb{1}\mathbb{1}, ZYXYZ, ZYYZ\mathbb{1}\}$
5. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}ZXZ, \mathbb{1}ZXZ\mathbb{1}, \mathbb{1}ZYYZ, XXXXX, XXY\mathbb{1}Y, XY\mathbb{1}YX, XYZZY, Y\mathbb{1}X\mathbb{1}Y, Y\mathbb{1}YXX, YZ\mathbb{1}ZY, YZZYX, ZX\mathbb{1}XZ, ZXZ\mathbb{1}\mathbb{1}, ZYXYZ, ZYYZ\mathbb{1}\}$
6. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}ZXZ, \mathbb{1}ZXZ\mathbb{1}, \mathbb{1}ZYYZ, X\mathbb{1}XZ\mathbb{1}, X\mathbb{1}YYZ, XZ\mathbb{1}\mathbb{1}\mathbb{1}, XZZXZ, YXXXY, YXY\mathbb{1}X, YY\mathbb{1}YY, YYZZX, ZX\mathbb{1}YY, ZXZZX, ZYXXY, ZYY\mathbb{1}X\}$
7. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}\mathbb{1}ZX, \mathbb{1}ZX\mathbb{1}X, \mathbb{1}ZXZ\mathbb{1}, X\mathbb{1}X\mathbb{1}X, X\mathbb{1}XZ\mathbb{1}, XZ\mathbb{1}\mathbb{1}\mathbb{1}, XZ\mathbb{1}ZX, YXY\mathbb{1}X, YXYZ\mathbb{1}, YYZ\mathbb{1}\mathbb{1}, YYZZX, ZXZ\mathbb{1}\mathbb{1}, ZXZZX, ZYY\mathbb{1}X, ZYYZ\mathbb{1}\}$
8. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}\mathbb{1}ZX, \mathbb{1}ZX\mathbb{1}X, \mathbb{1}ZXZ\mathbb{1}, X\mathbb{1}X\mathbb{1}X, X\mathbb{1}XZ\mathbb{1}, XZ\mathbb{1}\mathbb{1}\mathbb{1}, XZ\mathbb{1}ZX, YXXXY, YXXYZ, YY\mathbb{1}XZ, YY\mathbb{1}YY, ZX\mathbb{1}XZ, ZX\mathbb{1}YY, ZYXXY, ZYXYZ\}$
9. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}\mathbb{1}ZX, \mathbb{1}ZYXY, \mathbb{1}ZYYZ, X\mathbb{1}YXY, X\mathbb{1}YYZ, XZ\mathbb{1}\mathbb{1}\mathbb{1}, XZ\mathbb{1}ZX, YXXXY, YXXYZ, YYZ\mathbb{1}\mathbb{1}, YYZZX, ZXZ\mathbb{1}\mathbb{1}, ZXZZX, ZYXXY, ZYXYZ\}$
10. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}\mathbb{1}ZX, \mathbb{1}ZYXY, \mathbb{1}ZYYZ, X\mathbb{1}YXY, X\mathbb{1}YYZ, XZ\mathbb{1}\mathbb{1}\mathbb{1}, XZ\mathbb{1}ZX, YXY\mathbb{1}X, YXYZ\mathbb{1}, YY\mathbb{1}XZ, YY\mathbb{1}YY, ZX\mathbb{1}XZ, ZX\mathbb{1}YY, ZYY\mathbb{1}X, ZYYZ\mathbb{1}\}$
11. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}\mathbb{1}ZX, \mathbb{1}\mathbb{1}ZXZ, \mathbb{1}\mathbb{1}ZYY, XZ\mathbb{1}\mathbb{1}\mathbb{1}, XZ\mathbb{1}ZX, XZZXZ, XZZYY, YY\mathbb{1}XZ, YY\mathbb{1}YY, YYZ\mathbb{1}\mathbb{1}, YYZZX, ZX\mathbb{1}XZ, ZX\mathbb{1}YY, ZXZ\mathbb{1}\mathbb{1}, ZXZZX\}$
12. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}ZX\mathbb{1}, \mathbb{1}ZXZ\mathbb{1}, \mathbb{1}ZYY\mathbb{1}, X\mathbb{1}XZ\mathbb{1}, X\mathbb{1}YY\mathbb{1}, XZ\mathbb{1}\mathbb{1}\mathbb{1}, XZZX\mathbb{1}, YXXYZ, YXYZZ, YY\mathbb{1}XZ, YYZ\mathbb{1}Z, ZX\mathbb{1}XZ, ZXZ\mathbb{1}Z, ZYXYZ, ZYYZZ\}$
13. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}XZX, \mathbb{1}Z\mathbb{1}\mathbb{1}X, \mathbb{1}ZXZ\mathbb{1}, X\mathbb{1}\mathbb{1}\mathbb{1}X, X\mathbb{1}XZ\mathbb{1}, XZ\mathbb{1}\mathbb{1}\mathbb{1}, XZXZX, YXYZZ, YXZ\mathbb{1}Y, YYYZY, YYZ\mathbb{1}Z, ZXYZY, ZXZ\mathbb{1}Z, ZYYZZ, ZYZ\mathbb{1}Y\}$
14. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}ZX\mathbb{1}, \mathbb{1}Z\mathbb{1}\mathbb{1}X, \mathbb{1}ZZXX, X\mathbb{1}\mathbb{1}\mathbb{1}X, X\mathbb{1}ZXX, XZ\mathbb{1}\mathbb{1}\mathbb{1}, XZZX\mathbb{1}, YX\mathbb{1}XY, YXZ\mathbb{1}Y, YY\mathbb{1}XZ, YYZ\mathbb{1}Z, ZX\mathbb{1}XZ, ZXZ\mathbb{1}Z, ZY\mathbb{1}XY, ZYZ\mathbb{1}Y\}$
15. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}XZX, \mathbb{1}\mathbb{1}YYX, \mathbb{1}\mathbb{1}ZX\mathbb{1}, \mathbb{1}Z\mathbb{1}\mathbb{1}X, \mathbb{1}ZXZ\mathbb{1}, \mathbb{1}ZYY\mathbb{1}, \mathbb{1}ZZXX, X\mathbb{1}\mathbb{1}\mathbb{1}X, X\mathbb{1}XZ\mathbb{1}, X\mathbb{1}YY\mathbb{1}, X\mathbb{1}ZXX, XZ\mathbb{1}\mathbb{1}\mathbb{1}, XZXZX, XZYYX, XZZX\mathbb{1}\}$
16. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}ZXZ, \mathbb{1}ZXZ\mathbb{1}, \mathbb{1}ZYYZ, X\mathbb{1}XZZ, X\mathbb{1}YY\mathbb{1}, XZ\mathbb{1}\mathbb{1}Z, XZZX\mathbb{1}, YXXY\mathbb{1}, YXYZZ, YY\mathbb{1}X\mathbb{1}, YYZ\mathbb{1}Z, ZX\mathbb{1}XZ, ZXZ\mathbb{1}\mathbb{1}, ZYXYZ, ZYYZ\mathbb{1}\}$
17. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}XZZX, \mathbb{1}YXXY, \mathbb{1}ZYYZ, X\mathbb{1}XZZ, XXY\mathbb{1}Y, XY\mathbb{1}YX, XZZX\mathbb{1}, Y\mathbb{1}YXX, YXXY\mathbb{1}, YYZ\mathbb{1}Z, YZ\mathbb{1}ZY, Z\mathbb{1}ZYY, ZX\mathbb{1}XZ, ZYYZ\mathbb{1}, ZZX\mathbb{1}X\}$

C.5 Six qubits

In the case of five qubits there are four LC-inequivalent graphs with generators:

1. $\{XZZZZZ, ZX1111, Z1X111, Z11X11, Z111X1, Z1111X\}$,
2. $\{X1111Z, 1X111Z, 11X11Z, 111XZ1, 111ZXZ, ZZZ1ZX\}$,
3. $\{X1111Z, 1X111Z, 11X1Z1, 111XZ1, 11ZZXZ, ZZ11ZX\}$,
4. $\{XZ1111, ZX111Z, 1ZXZ11, 11ZXZ1, 111ZX1, 1Z111X\}$,
5. $\{XZ1111, ZXZ111, 1ZXZ1Z, 11ZXZ1, 111ZX1, 11Z11X\}$,
6. $\{XZ1111, ZXZ111, 1ZXZ11, 11ZXZ1, 111ZXZ, 1111ZX\}$,
7. $\{X1111Z, 1X1Z11, 11XZ1Z, 1ZZXZ1, 111ZXZ, Z1Z1ZX\}$,
8. $\{XZ1111, ZXZZ11, 1ZXZ1Z, 1ZZXZ1, 111ZX1, 11Z11X\}$,
9. $\{XZ11ZZ, ZXZ111, 1ZXZ11, 11ZXZ1, Z11ZX1, Z1111X\}$,
10. $\{XZ111Z, ZXZ111, 1ZXZ11, 11ZXZ1, 111ZXZ, Z111ZX\}$,
11. $\{XZZ11Z, ZXZ1Z1, ZZXZ11, 11ZXZZ, 1Z1ZXZ, Z11ZZX\}$.

In case subgroups with one generator are considered there are 5 classes:

1. $\{111111, XZZZZZ\}$
2. $\{111111, ZX1111\}$
3. $\{111111, ZX11XX\}$
4. $\{111111, YZZ1ZY\}$
5. $\{111111, X11ZX1\}$

In case subgroups with two generators are considered there are 34 classes:

1. $\{111111, XZZZZZ, YYZZZZ, ZX1111\}$
2. $\{111111, XZZZZZ, YYZZYY, ZX11XX\}$
3. $\{111111, XZZZZZ, YYYYYY, ZXXXXX\}$
4. $\{111111, 1XX111, Z1X111, ZX1111\}$
5. $\{111111, 1X111X, Z1X111, ZXX11X\}$
6. $\{111111, 1XX1XX, Z1X111, ZX11XX\}$
7. $\{111111, 1X1XXX, Z1X111, ZXXXXX\}$
8. $\{111111, 1XXXX1, Z1X1XX, ZX1X1X\}$
9. $\{111111, 1X111Z, YYZ1ZX, YZZ1ZY\}$
10. $\{111111, 1X111Z, X11ZX1, XX1ZXZ\}$
11. $\{111111, X11ZX1, YYZZYY, ZYZ1ZY\}$
12. $\{111111, XX1ZXZ, YZZZYX, ZYZ1ZY\}$
13. $\{111111, 1X1ZX1, X11ZX1, XX1111\}$
14. $\{111111, X11XZZ, YYZX1X, ZYZ1ZY\}$

15. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}X\mathbb{1}ZX\mathbb{1}, X\mathbb{1}\mathbb{1}XZZ, XX\mathbb{1}YYZ\}$
16. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}X\mathbb{1}ZX\mathbb{1}, X\mathbb{1}\mathbb{1}YY\mathbb{1}, XX\mathbb{1}XZ\mathbb{1}\}$
17. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, XXX\mathbb{1}\mathbb{1}Z, YZY\mathbb{1}ZX, ZYZ\mathbb{1}ZY\}$
18. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}X\mathbb{1}\mathbb{1}\mathbb{1}Z, X\mathbb{1}XZXZ, XXXZX\mathbb{1}\}$
19. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}X\mathbb{1}ZX\mathbb{1}, X\mathbb{1}XZXZ, XXX\mathbb{1}\mathbb{1}Z\}$
20. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}\mathbb{1}XZ\mathbb{1}, X\mathbb{1}\mathbb{1}YY\mathbb{1}, X\mathbb{1}\mathbb{1}ZX\mathbb{1}\}$
21. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}X\mathbb{1}\mathbb{1}\mathbb{1}Z, YY\mathbb{1}\mathbb{1}ZX, YZ\mathbb{1}\mathbb{1}ZY\}$
22. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, X\mathbb{1}ZZX\mathbb{1}, YYZZYY, ZY\mathbb{1}\mathbb{1}ZY\}$
23. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, X\mathbb{1}\mathbb{1}XZZ, YY\mathbb{1}X\mathbb{1}X, ZY\mathbb{1}\mathbb{1}ZY\}$
24. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, X\mathbb{1}XZ\mathbb{1}\mathbb{1}, YXXZ\mathbb{1}Z, ZX\mathbb{1}\mathbb{1}\mathbb{1}Z\}$
25. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, XZXZ\mathbb{1}X, YYXZ\mathbb{1}Y, ZX\mathbb{1}\mathbb{1}\mathbb{1}Z\}$
26. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, X\mathbb{1}XZ\mathbb{1}\mathbb{1}, YXX\mathbb{1}XZ, ZX\mathbb{1}ZXZ\}$
27. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, XZXZ\mathbb{1}X, YYX\mathbb{1}XY, ZX\mathbb{1}ZXZ\}$
28. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, X\mathbb{1}YYZ\mathbb{1}, YXYYZZ, ZX\mathbb{1}\mathbb{1}\mathbb{1}Z\}$
29. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, XZYYZX, YYYYZY, ZX\mathbb{1}\mathbb{1}\mathbb{1}Z\}$
30. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, X\mathbb{1}YYZ\mathbb{1}, YXYXYZ, ZX\mathbb{1}ZXZ\}$
31. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, XZYYZX, YYYXYY, ZX\mathbb{1}ZXZ\}$
32. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}ZXZ\mathbb{1}\mathbb{1}, X\mathbb{1}X\mathbb{1}X\mathbb{1}, XZ\mathbb{1}ZX\mathbb{1}\}$
33. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, \mathbb{1}\mathbb{1}ZXZ\mathbb{1}, YY\mathbb{1}\mathbb{1}\mathbb{1}Z, YYZXZZ\}$
34. $\{\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}\mathbb{1}, XZ\mathbb{1}ZX\mathbb{1}, YY\mathbb{1}YY\mathbb{1}, ZX\mathbb{1}XZ\mathbb{1}\}$

We also did the characterisation for $k = 3$ and $k = 4$ generators, which consist of 120 and 166 classes, respectively. For $k = 5$ the characterisation is already quite demanding and beyond the scope of our Python script. At this point we will refrain from listing all these classes.

Appendix D

Semidefinite programming with Python and MATLAB

D.1 Optimisation frameworks

These frameworks are usually integrated into more versatile numerical computing environments or general purpose scripting languages. Here one package for MATLAB v8.2[1] [121] and one module for Python v2.7.3[2] and Sage v6.0[3] [122], respectively shall be presented. For MATLAB the usage of the framework YALMIP v3 [123, 124] as for Python/Sage the usage of CVXOPT v1.1.6 [125] is introduced. As for the mentioned solvers we will briefly discuss SeDuMi $v1.3$ [126] and SDPT3 $v4.0$ [127, 128], which can be used by YALMIP and DSDP5 $v5.8$ [129], and CONELP [125], which can be accessed by CVXOPT.

In order to discuss the usage of the two optimisation packages a purely academical example of a semidefinite program will be provided. That is, the maximum eigenvalues of four symmetric matrices $A_1, A_2, A_3, A_4 \in \mathbb{R}^{m\times m}$ are computed using semidefinite programming. This can be done using the following SDP:

$$\begin{aligned} \min_x & \sum_{i=1}^{4} x_i \\ \text{s.t. } & x_i \mathbb{1}_m - A_i \geq 0 \text{ for all } i = 1, \ldots, 4. \end{aligned} \tag{D.1}$$

Here the optimum will converge to the sum of the maximal eigenvalues, whereas the optimal point will yields the individual maximal eigenvalues $\vec{x}_i^* = \lambda_{max}(A_i)$. Comparing Eq. (D.1) to the standard formulation of the primal problem as given in Eq. (5.1), we identify

$$\begin{aligned} c &= (1, 1, 1, 1), \\ F_0 &= -\bigoplus_{i=1}^{4} A_i, \\ F_1 &= \mathbb{1}_m \oplus \mathbb{0}_m \oplus \mathbb{0}_m \oplus \mathbb{0}_m, \\ F_2 &= \mathbb{0}_m \oplus \mathbb{1}_m \oplus \mathbb{0}_m \oplus \mathbb{0}_m, \\ F_3 &= \mathbb{0}_m \oplus \mathbb{0}_m \oplus \mathbb{1}_m \oplus \mathbb{0}_m, \\ F_4 &= \mathbb{0}_m \oplus \mathbb{0}_m \oplus \mathbb{0}_m \oplus \mathbb{1}_m, \end{aligned} \tag{D.2}$$

[1] MATLAB is a non-free, closed source mathematical environment, which is widely used for numerical computations.

[2] Python is a cross platform, free, open source, general purpose and high-level programming language that has code readability as its main paradigm.

[3] Sage is a cross platform, free and open source mathematics software, which aims to be an alternative to Mathematica and Matlab. It is based on the Python scripting language, but also includes other scripting languages (such as R) and a graphical user interface.

where $\mathbb{O}_m \in \mathbb{R}^{m\times m}$ is the null matrix. Consider the case, where $A_k \in \mathbb{R}^{m\times m}$ is a matrix having k on the diagonal and anti-diagonal and zero elsewhere. That is in the case $m = 5$

$$A_k = \begin{pmatrix} k & 0 & 0 & 0 & k \\ 0 & k & 0 & k & 0 \\ 0 & 0 & k & 0 & 0 \\ 0 & k & 0 & k & 0 \\ k & 0 & 0 & 0 & k \end{pmatrix}. \tag{D.3}$$

Independent of m the largest eigenvalue of A_k is equal to $2k$. Hence, the optimal point of the SDP given by Eq. (D.1) is $\vec{x} = (2, 4, 6, 8)$ and thus $\vec{c}^T\vec{x} = 20$.

D.1.1 CVXOPT / CONELP & DSDP5.8

CVXOPT is a free Python module for convex optimisation. It can be used directly from within the Python command line as module, executed from within a Python script or used from within the free and open source Sage mathematics framework. Although one might stress that the later does require some care, since Sage has modified some part of the Python. Note that CVXOPT can also be used for more general optimisation problems, such as quadratic programming. However, these problems are not covered in this paragraph.

To illustrate the features, syntax and performance of CVXOPT three versions of the semidefinite program as given by Eq. (D.1) with $m = 50$, $m = 250$ and $m = 500$ will be implemented using the standard solver CONELP and the external solver DSDP5 to numerically calculate the optimum. Observe that the run-time of our programs strongly depends on how the problem is parsed before given to a solver as can be seen in Tab. D.1. In this context parsing refers to the process of transforming an SDP as given in Eq. (5.1) into a form, which can be passed to a solver.

To illustrate this process and give some general guidelines[4]. First, the natural implementation is chosen, where all constraints are collected in one large block-diagonal matrix as given by Eq. (D.2). Second, the block structure is used to implement each block as an independent constraint, which speeds up the run-time of the solvers. Last, the fact that the matrices A_i are sparse[5] is exploited to further boost the performance of the external solver DSDP5.

The complete source code of the natural implementation is provided in Listing D.1 for $m = 50$. At this point, the implementation and syntax shall be discussed in detail: lines 1 to 4 contain the header of the Python script. There the modules `cvxopt, numpy, scipy.linalg` are imported and references to the public objects `matrix, solvers, array, eye, zero, block_diag` are created[6]. In lines 7 to 14 the matrices F_0 to F_4 [see Eqs. (D.2)] are initialised as `numpy.array`. Then the problem vector $\vec{c}$ is set to $(1, 1, 1, 1)$ in line 17 using the `matrix` class of CVXOPT. One finishes parsing the problem by setting the semidefinite constraints in lines 19 and 21. The matrices in `G` and `h` correspond to the constraint $h \geq G(\vec{x}) = \sum_i G_i x_i$. Compare this to $F(x) \geq 0$ then $h = F_0$ and $G_i = -F_i$[7]. Note that `h` is a list containing the matrix F_0 and `G` is a list containing one matrix made of all G_i. That is the i-th row of this matrix is given by the concatenation of the rows of $G_i = -F_i$ (`list(g.flatten())`). Finally, lines 24 and 25 call the standard SDP solver CONELP[8] and print the point $\vec{x}$ solving the problem.

Listing D.1: Natural implementation: CVXOPT & standard solver

```
1 import numpy, cvxopt, scipy.linalg
2 from cvxopt import matrix, solvers
3 from numpy import array, eye, zeros
4 from scipy.linalg import block_diag
5
6 # Initialize problem matrices.
7 m = 50
```

[4] Unrelated to the discussion at hand, the author recommends reading Ref. [130] solely for admiration, since it is clearly one of best written papers ever published.

[5] Note that a matrix is said to be sparse if it is primarily populated with zeros. If a matrix is not sparse, it is said to be dense. Furthermore, linear algebra packages offer special algorithms to process sparse matrices, providing a significant performance boost compared to dense matrices.

[6] In that way `array` instead of `numpy.array` can be used as done for instance in line 8 in Listing D.1.

[7] The signs in front of F_i are easily forgotten, which causes parsing of a different SDP.

[8] Here "standard" refers to CVXOPT.

```
A = array([[1. if i==j or i+j==9 else 0. for i in range(m)] for j in range(m)])
O = zeros((m,m))
F0 = -block_diag(1*A, 2*A, 3*A, 4*A)
F1 = block_diag(eye(m), O, O, O)
F2 = block_diag(O, eye(m), O, O)
F3 = block_diag(O, O, eye(m), O)
F4 = block_diag(O, O, O, eye(m))

# Set problem vector c.
c = matrix([1.,1.,1.,1.])
# Set variable part in the semidefinite constraint F(x)>=0.
G = [ matrix([list(g.flatten()) for g in [-F1,-F2,-F3,-F4]]) ]
# Set constant part F0 in semidefinite constraint F(x)>=0.
h = [ matrix(F0) ]

# Start solver and print solution x.
sol = solvers.sdp(c, Gs=G, hs=h)
print sol['x']
```

A more advanced approach to implement the semidefinite program for $m = 50$ is given in Listing D.2. In this implementation the block-diagonal structure of $F(\vec{x})$ is explicitly used. That is the single constraint $F(\vec{x}) \geq 0$ with $F_0, \ldots, F_4$ as given in Eqs. (D.2) is split up into $\sum_j \delta_{ij} x_j \mathbb{1}_m - A_i \geq 0$, $i = 1, \ldots, 4$. We parse these in lines 12 to 16, where the i-th item in `G` and `h` correspond to the constraint $\sum_j \delta_{ij} x_j \mathbb{1}_m - A_i \geq 0$[9]. Note that the for all practical purposes, the run-time of the program shortens considerably as can be seen in Tab. D.1, where the run-times of all implementations are given relative to the fastest implementation, which is given by the code in Listing D.3.

Listing D.2: Exploiting the block-diagonal structure: CVXOPT & standard solver

```
1  import numpy, cvxopt
2  from cvxopt import matrix, solvers
3  from numpy import array, eye, zeros
4
5  m = 50
6  A = array([[1. if i==j or i+j==9 else 0. for i in range(m)] for j in range(m)])
7  O = zeros((m,m))
8  F = eye(m)
9
10 c = matrix([1.,1.,1.,1.])
11 # Use block-diagonal structure of F(x)>=0. Each block encodes separate constraints.
12 G = [ matrix([list(g.flatten()) for g in [-F,O,O,O]]) ]
13 G += [ matrix([list(g.flatten()) for g in [O,-F,O,O]]) ]
14 G += [ matrix([list(g.flatten()) for g in [O,O,-F,O]]) ]
15 G += [ matrix([list(g.flatten()) for g in [O,O,O,-F]]) ]
16 h = [ matrix(-1*A), matrix(-2*A), matrix(-3*A), matrix(-4*A) ]
17
18 sol = solvers.sdp(c, Gs=G, hs=h)
19 print sol['x']
```

In the last example, two minor modifications to the program given by Listing D.2 are introduced. First, the solver DSDP5 is used instead of the standard solver CONELP in line 23 by providing the additional argument `solver='dsdp'` in the function `solvers.sdp`. Since this solver has its own default settings the step monitor and the intended accuracy goal have to be set to the same value as the ones CONELP uses in lines 6 and 7 to achieve comparability between the run-times of both solvers. Furthermore, the matrices in `G` have to be provided not as dense matrices (`matrix`) but as sparse ones (`sparse`). The main advantage of doing so is that a suitable solver can make use of the sparsity to reduce the total run-time.

Listing D.3: Using sparsity: CVXOPT & DSDP

```
1  import numpy, cvxopt
2  from cvxopt import matrix, sparse, solvers
3  from numpy import array, eye, zeros
4
5  # Enable step monitor and set exit condition.
6  solvers.options['DSDP_Monitor'] = 1
```

[9] Each of the constraints is parsed as discussed in the previous implementation.

```
7  solvers.options['DSDP_GapTolerance'] = 1e-8
8
9  m = 50
10 A = array([[1. if i==j or i+j==9 else 0. for i in range(m)] for j in range(m)])
11 O = zeros((m,m))
12 F = eye(m)
13
14 c = matrix([1.,1.,1.,1.])
15 # Use sparse matrices.
16 G = [ sparse([list(g.flatten()) for g in [-F,O,O,O]]) ]
17 G += [ sparse([list(g.flatten()) for g in [O,-F,O,O]]) ]
18 G += [ sparse([list(g.flatten()) for g in [O,O,-F,O]]) ]
19 G += [ sparse([list(g.flatten()) for g in [O,O,O,-F]]) ]
20 h = [ matrix(-1*A), matrix(-2*A), matrix(-3*A), matrix(-4*A) ]
21
22 # Use alternative solver DSDP.
23 sol = solvers.sdp(c, Gs=G, hs=h, solver='dsdp')
24 print sol['x']
```

A comparison of the run-time of all three versions for $m = 50$, $m = 250$ and $m = 500$ using both available solvers is given in Tab. D.1. Clearly, the natural implementation runs slowest independent of the solver used. Hence, it is crucial to explicitly exploit the block-diagonal structure of a problem. That is to provide semidefinite constraints within each diagonal block independently if possible as we have done in the code given in Listing D.2. No matter what solver used the code in Listing D.2 gives a huge advantage with respect to the run-time of the program.

Furthermore, one may try to make use of sparse matrices as we have shown in the code in Listing D.3. The standard solver CONELP, however, shows no improvement of the run-time and thus does not seem to make use of sparse matrices. As for the solver DSDP5 the explicit use of sparse matrices shortens the run-time of the SDP by roughly 6% for $m = 50$, 16% for $m = 250$ and 26% for $m = 500$ compared to the run times of the code given in Listing D.2 using the same solver.

The data in Tab. D.1 also shows that the advantage of exploiting the block-diagonal and sparse structure of the problem strongly depends on the size of the problem and the solver used. Whereas for $m = 50$ the code in Listing D.3 using the solver DSDP5 runs roughly 19 times faster than the code given in Listing D.1 using CONELP, the speed up increases to 39 times for $m = 500$. That is the run-times scale differently with m and increase slowest for the code given in Listing D.3 using the solver DSDP5.

Remark D.1. We conclude this paragraph with some general remarks about the use of CVXOPT and the solvers CONELP and DSDP5. As can be seen from the examples provided in Listings D.1, D.2 and D.3 the usage of CVXOPT is quite demanding with respect to parsing the problem[10]. It is up to the programmer to explicitly exploit all the given structure to reduce the computational power needed to solve a given semidefinite program. Specifically, if the problem is large, it is often necessary to exploit all the structure for the SDP to fit into the memory and thus be numerically feasible. On the one hand this requires a lot of work, but on the other hand a program written within CVXOPT exploiting all the given structure often leads to fast running programs.

Note that if one wants to exploit sparsity, then one has to rely on the solver DSDP5. Programs passed to this alternative solver often require less memory during run-time and have a remarkable performance compared to CONELP. On the downside this solver does not come along with the CVXOPT package and has to be installed manually[11].

As for the standard solver CONELP the usage is recommended only in the cases, where the involved matrices are dense or DSDP5 runs into numerical problems[12], i.e., if time is no concern[13] and reliability is required.

[10] Badly written code can have a severe effect on the run-time.

[11] Most Linux distributions provide a repository package providing this solver. On Debian/Ubuntu operating systems it can be installed using the command line `$ apt-get install dsdp`.

[12] In certain instances, it might occur that a solver runs into numerical problems. That usually means that the algorithm fails to provide a new iteration step at some point and then stops.

[13] That also includes the case, where the SDP has very few variables with small constraints. That is $n, m \in \mathbb{N}$ as given in Eq. (5.1) are of the order of a few hundred.

	Solver					
Code	CONELP	DSDP5	CONELP	DSDP5	CONELP	DSDP5
Listing D.1	2.80s	0.76s	170s	48s	1100s	310s
Listing D.2	0.29s	0.16s	19s	7.3s	94s	38s
Listing D.3	0.29s	0.15s	19s	6.1s	96s	28s
	$m = 50$		$m = 250$		$m = 500$	

Table D.1: This table shows the performance of the three implementations (see Listings D.1, D.2 and D.3) of the SDP given by Eq. (D.1) for different sizes $m \times m$ of the semidefinite constraint using the native solver CONELP of CVXOPT and the third party solver DSDP5. To get a good estimate on the run-time the run-times are averaged using 100 runs using the Linux command line `$ time for i in {1..100}; do python script.py; done`. The times given here include the overhead of calling Python, loading the modules and parsing the SDP.

D.1.2 YALMIP / SeDuMi & SDPT3

In this paragraph the MATLAB framework for semidefinite programming called YALMIP shall be discussed. Compared to the Python framework CVXOPT, which we discussed previously, it offers a quite natural syntax to formulate semidefinite programs and includes a parser, which translates the problems to be forwarded to a solver. Furthermore, it provides compatibility to a large number of different solvers, a complete discussion of which is beyond the scope of this thesis. Therefore, solely the semidefinite program solvers SeDuMi and SDPT3 are considered. As in the previous paragraph we formulate the SDP given in Eq. (D.1) in three different ways. First, we formulate it naturally. Second, we explicitly make use of the block-diagonal structure and split the single large semidefinite constraint into four. Finally, we try if one of the solvers can make use of the sparsity of the matrices.

The code of the natural implementation is given in Listing D.4[14]. Note that the first eight lines of the code are essentially similar to lines 6 to 14 in Listing D.1. I.e., they set up the matrices involved in defining the semidefinite constraints. Compared to CVXOPT, however, YALMIP offers a very natural syntax to formulate a SDP. Variables are introduced by calling `sdpvar u v w x`. Then one sets the linear function to be minimised by `objective = u+v+w+x` and finally the constraints `constraints = [F0+u*F1+v*F2+w*F3+x*F4>=0]`. Upon calling the solver in line 18 YALMIP invokes its included parser. This automatically does all the low level problem setting done in lines 17, 19 and 21 in Listing D.1 in the natural CVXOPT code manually and solves the SDP. On the one hand side this is quite convenient as it saves a lot of time and makes the code more readable. On the other hand one has no direct influence on the parser, which might lead to otherwise avoidable bottlenecks. In this example we use the solver SeDuMi. A different solver can be used by uncommenting line 16 and commenting line 17.

Listing D.4: Natural Implementation: YALMIP & SeDuMi/SDPT3

```
1  % Initialize problem matrices
2  m = 50;
3  A = zeros(m,m); for i=1:50, A(i,i)=1.; A(i,51-i)=1.; end
4  F0 = blkdiag(-1*A,-2*A,-3*A,-4*A);
5  F1 = blkdiag(eye(m),zeros(m,m),zeros(m,m),zeros(m,m));
6  F2 = blkdiag(zeros(m,m),eye(m),zeros(m,m),zeros(m,m));
7  F3 = blkdiag(zeros(m,m),zeros(m,m),eye(m),zeros(m,m));
8  F4 = blkdiag(zeros(m,m),zeros(m,m),zeros(m,m),eye(m));
9
10 % Initialize SDP
11 sdpvar u v w x
12 objective = u+v+w+x;
13 constraints = [F0+u*F1+v*F2+w*F3+x*F4>=0];
14
15 % Set options, start solver and print solution; Uncomment next line and comment the
       following to set SeDuMi as solver
16 %options = sdpsettings('verbose',1,'solver','sedumi','showprogress',1);
17 options = sdpsettings('verbose',1,'solver','sdpt3','showprogress',1);
18 sol = solvesdp(constraints,objective,options)
```

[14] All code provided in this paragraph can be executed from within MATLAB after the inclusion of YALMIP, SeDuMi and SDPT3.

```
19 solution = double([u,v,w,x])
```

Using YALMIP, it is then easy to provide the semidefinite constraints in the diagonal blocks of our problem as independent constraints. As can be seen in Listing D.5, we first provide the $m \times m$ matrices in lines 2 and 3 to formulate the four independent constraints given as in line 7.

Listing D.5: Exploiting block-diagonal structure: YALMIP & SeDuMi/SDPT3

```
1  m = 50;
2  A = zeros(m,m); for i=1:50, A(i,i)=1.; A(i,51-i)=1.; end
3  F = eye(m);
4
5  sdpvar u v w x
6  objective = u+v+w+x;
7  constraints = [u*F-1*A>=0, v*F-2*A>=0, w*F-3*A>=0, x*F-4*A>=0];
8
9  %options = sdpsettings('verbose',1,'solver','sedumi','showprogress',1);
10 options = sdpsettings('verbose',1,'solver','sdpt3','showprogress',1);
11 sol = solvesdp(constraints,objective,options)
12 solution = double([u,v,w,x])
```

Our last code given in Listing D.6 is a slight alteration of the one given in Listing D.5. The main difference is that all matrices are explicitly defined as sparse matrices. Note that this is in principle not necessary, since YALMIP detects sparsity itself. This case was nonetheless included to highlight the differences between CVXOPT and YALMIP in the following discussion.

Listing D.6: Using sparsity: YALMIP & SeDuMi/SDPT3

```
1  m = 50;
2  A = sparse([1:1:m 1:1:m],[1:1:m m:-1:1],ones(1,2*m),m,m);
3  F = speye(m,m);  % Sparse mxm identity matrix
4  O = sparse(m,m); % Sparse mxm null matrix
5
6  sdpvar u v w x
7  objective = u+v+w+x;
8  constraints = [u*F-1*A>=0, v*F-2*A>=0, w*F-3*A>=0, x*F-4*A>=0];
9
10 %options = sdpsettings('verbose',1,'solver','sedumi','showprogress',1);
11 options = sdpsettings('verbose',1,'solver','sdpt3','showprogress',1);
12 sol = solvesdp(constraints,objective,options)
13 solution = double([u,v,w,x])
```

Similarly to the last paragraph the run-times of our programs were averaged over several runs, whereas the time YALMIP needed to parse the problem was not taken into account. Here, however, we did not show the run-times for different m, since the qualitative behaviour is the same. The results of the performance tests for both solvers (SeDuMi and SDPT3) are given in Tab. D.2.

The solver SeDuMi shows qualitatively a similar behaviour as the CVXOPT solver CONELP (see Tab. D.1). Its run-time is long if the natural implementation is used and significantly decreases if we split the large constraint into four smaller ones exploiting the block-diagonal structure of the semidefinite constraint used in the natural implementation. Our tests shows that the explicit use of sparse matrices can be omitted, since sparsity is automatically detected by the parser and used by SeDuMi [126]. Overall the run-time of SeDuMi is better than that of CONELP, which is to be expected with respect to the sparsity of our constraints. Compared to DSDP5, however, SeDuMi is noticeably slower.

As for the second solver SDPT3 our tests show an unexpected behaviour as can be seen in Tab. D.2. Contrary to SeDuMi, CONELP and DSDP5, its run-time is best and comparable to the fastest run-time of DSDP5 if the natural implementation is used. This leads to the conclusion that the parser detects the block-diagonal structure and the sparsity in this case, both of which boost the performance of SDPT3 [127]. If we explicitly provide the four independent constraints, the parser seems to be unable to fully exploit the problem structure, which leads to an increased run-time of this solver as shown in Tab. D.2.

Remark D.2. Let us conclude this paragraph with some general remarks about YALMIP and the solvers SeDuMi and SDPT3. First, note that the solver SDPT3 has a good performance compared to SeDuMi and is almost as fast as DSDP5 to solve our problem. For larger problems, however, the run-times of DSDP5 scale slightly better than the run-times of SDPT3.

	Solver	
Code	SeDuMi	SDPT3
Listing D.4	$0.72s$	$0.17s$
Listing D.5	$0.24s$	$0.20s$
Listing D.6	$0.24s$	$0.20s$
	$m = 50$	

Table D.2: This table shows the performance of the three implementations (see Listings D.4, D.5 and D.6) of the SDP given by Eq. (D.1) using SeDuMi and SDPT3. Here, the time the solver needed to solve the problem as provided by the output of the function `solvesdp` was averaged over several runs. The time the parser needs to translate the problem is not included.

First, as already mentioned in the last paragraph in certain instances it might occur that a solver runs into numerical problems. In this case SeDuMi should be used, since its algorithm is more robust.

Second, independent of the solver used YALMIP offers a very intuitive and natural syntax to formulate semidefinite problems. This in general saves a lot of programming time compared to CVXOPT, where one needs to manually parse the problem. Although it might occur that the problem structure cannot fully be exploited. In this case, the gain in run-time can justify the use of CVXOPT.

As for the included parser, one should be aware that it does not always behave the way one might expect. That is dependent on the solver different formulations of a SDP can lead to either an increase or decrease of the run-time. Finally, it is worth mentioning that CVXOPT used together with DSDP5 performs best and has the lowest memory requirements in our experience. Hence it is a good choice if large scale semidefinite programs are to be solved. If on the other hand it is foreseeable that the run-time is small, then it might be faster to use the more natural syntax of YALMIP.

D.2 GMN implementation using CVXOPT

Before the CVXOPT implementation of the renormalised genuine multiparticle negativity is sketched, note that the corresponding code is optimised for speed and hence may lack readability.

Assuming $\mathcal{H} = \mathbb{C}^n$, the concatenation of the following sets has been chosen as operator basis: Altogether, $\{E_{ii}\}_{i=1}^{n}$, $\{E_{ij} + E_{ji}\}_{i=1,i<j}^{n}$ and $\{-iE_{ij} + iE_{ji}\}_{i=1,i<j}^{n}$, where $E_{ij} \in \mathbb{C}^{n\times n}$ with $(E_{ij})_{kl} = \delta_{ik}\delta_{jl}$. There are n^2 elements in this basis and thus W and all P_m are described by n^2 independent variables each.

According to the chosen basis, the coefficients $\varrho^{(i)}$ are given in lines 99, 104 and 105 in Listing D.7. From these entries the problem vector is build in line 108 according to Eq. (5.22).

The positivity constraints $0 \leq \phi(P_m(\vec{x}))$ of P_m are then parsed in lines 133 to 148. Note that in this part the sparse matrices are constructed directly from the values and positions of the non-vanishing entries. The constraints $0 \leq \phi\left([\mathcal{W}(\vec{x}) - P_m(\vec{x})]^{T_m}\right) \leq \mathbb{1}$ are parsed in lines 159 to 166. Especially important for this step is the action of the partial transposition, which changes the position of some off-diagonal elements as calculated in lines 166 to 179.

The function then ends by calling the solver DSDP5 in lines 203 and CONELP in line 205, respectively.

Listing D.7: Implementation of the (renormalised) genuine multiparticle negativity using CVXOPT

```
1  #This program implements the (renormalized) genuine multiparticle negativity using cvxopt
2  #    Copyright (C) 2014 Martin Hofmann
3  #
4  #    This program is free software: you can redistribute it and/or modify
5  #    it under the terms of the GNU General Public License as published by
6  #    the Free Software Foundation, either version 3 of the License, or
7  #    (at your option) any later version.
8  #
9  #    This program is distributed in the hope that it will be useful,
10 #    but WITHOUT ANY WARRANTY; without even the implied warranty of
11 #    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
12 #    GNU General Public License for more details.
13 #
14 #    You should have received a copy of the GNU General Public License
```

```
#    along with this program.  If not, see <http://www.gnu.org/licenses/>.


#### load modules and import classes and functions ####
# Modules needed for the partial transposition and problem creation
import numpy as np
from numpy import array, dtype, r_, reshape, transpose
# Modules needed for the sdp solver
import cvxopt
# To use dsdp from within sage comment line above and uncomment 2 lines below
#import imp
#cvxopt = imp.load_module('cvxopt', None,  '/usr/lib/python2.7/dist-packages/cvxopt', ('',
    '', 5))
from cvxopt.base import spmatrix
from cvxopt import solvers


#### set solver options ####
# Set options for standard sdp solver
solvers.options["show_progress"] = False
solvers.options["abstol"] = 1.e-12
solvers.options["reltol"] = 1.e-12
solvers.options["feastol"] = 1.e-8
solvers.options['maxiters'] = 100

# Set options for dsdp solver
solvers.options['DSDP_Monitor'] = 0 # integer (default: 0)
solvers.options['DSDP_MaxIts'] = 100 # positive integer
solvers.options['DSDP_GapTolerance'] = 1e-12 # scalar (default: 1e-5).


#### definition of partial transposition ####
def ptranspose(state, dim, subsys):
    """Partial transpose.
    Returns the partial transpose of the state
    wrt. the subsystems listed in the vector sys.
    """
    # The original partial transposition function
    # was provided by Ville Bergholm (see http://qit.sourceforge.net/)
    # and licenced under the terms of the GPL3.
    nsys = len(dim)
    s = state
    orig_d = s.shape
    # which systems to transpose
    subsys = array(list(set(range(nsys)).intersection(set(subsys))), int)

    # swap the transposed dimensions
    perm = np.arange(2 * nsys)  # identity permutation
    perm[r_[subsys, subsys + nsys]] = perm[r_[subsys + nsys, subsys]]

    # flat matrix into tensor, partial transpose, back into a flat matrix
    res = s.reshape(dim + dim).transpose(perm).reshape(orig_d)

    return res


#### define real and imaginary part ####
def real(_x): return _x.real
def imag(_x): return _x.imag


#### define the (renormalized) genuine multiparticle negativity ####
def gmn(rho,dim,renormalized=True, solver='dsdp'):
    """(renormalized) genuine multiparticle negativity.
    Returns the (renormalized) genuine multiparticle negativity (GMN) of the state rho

    Parameters : rho          : rho array
                                A density matrix whose (renormalized) GMN is to be computed
                 dim          : (n1,n2,...,n_k) tuple
```

```
                                A tuple containing the dimensions of the subsystems
                renormalized : {True,False} boolean
                                A boolean wheather to compute the renormalize GMN
                                introduced in Ref. [1] or the original version in Ref. [2]
                                [1] M. Hofmann et al. arXiv:1401.2424 (2014).
                                [2] B. Jungnitsch et al. PRL 106, 190502 (2011).
                solver       : {'conelp','dsdp'} string
                                A string which solver to use
                                The cvxopt standard conelp [3] or dsdp5 [4]
                                [3] M.S. Andersen et al. abel.ee.ucla.edu/cvxopt (2012).
                                [4] S.J. Benson and Y. Ye. www.mcs.anl.gov/hs/software/DSDP/
                                    (2005).
    """

    nsys = len(dim)
    n = int(np.prod(dim))
    # c is set to minimize tr(rho*W)
    cd = map(real,rho.diagonal().tolist())
    cr = []
    ci = []
    for j in range(n):
        for i in range(j+1,n):
            cr+=[2*real(rho[i][j])]
            ci+=[-2*imag(rho[i][j])]
    cpm = (2**(nsys-1)-1)*[0. for i in xrange(n**2)]

    c = cvxopt.matrix(array(cd+cr+ci+cpm),tc='d')
    del cd, cr, ci, cpm

    # initialization of the lists holding the semidefinite constraints
    G = []
    h = []

    #Introduce the constraints on the P_M: 0 <= P_M (<=1)
    #y coordinates
    X = []
    #x coordinates for diagonal part
    for d in range(n):
        X.extend([d*(2*n+1),(d+n)*(1+2*n)])
    #x coordinates for offdiag real part
    for x in range(0,n-1):
        for y in range(x+1,n):
            X.extend([x+2*n*y, y+2*n*x, x+2*n*y+2*n*n+n, y+2*n*x+2*n*n+n])
    #x coordinates for offdiag imag part
    for x in range(0,n-1):
        for y in range(x+1,n):
            X.extend([x+2*n*y+n, y+2*n*x+n, x+2*n*y+2*n*n, y+2*n*x+2*n*n])

    # values for sparse matrices of both constraints
    Valle1 = n*[1,1]+n*(n-1)/2*[1,1,1,1]+n*(n-1)/2*[1,-1,-1,1]
    Valge0 = n*[-1,-1]+n*(n-1)/2*[-1,-1,-1,-1]+n*(n-1)/2*[-1,1,1,-1]

    for i in range(1,2**(nsys-1)):
        #y coordinates
        Y = []
        for j in range(n):
            Y.extend(2*[j+i*n*n])
        for j in range(n,n*n):
            Y.extend(4*[j+i*n*n])

        # do not upper bound P_m if renormalized genuine mutliparticle negativity is used
        if not renormalized:
            G += [ cvxopt.spmatrix(Valle1, X, Y, (4*n*n,n*n*2**(nsys-1))) ]
            h += [ cvxopt.matrix( np.eye(2*n) , tc='d' ) ]

        G += [ cvxopt.spmatrix(Valge0, X, Y, (4*n*n,n*n*2**(nsys-1))) ]
        h += [ cvxopt.matrix( np.zeros((2*n,2*n)) , tc='d' ) ]

    #In this part we introduce the constraints 0<=(W-P_M)^(T_M)<=1
```

```
    XY = [None for i in xrange(n*(n-1)/2)]

    YW = []
    for j in range(n):
        YW.extend(2*[j])
    for j in range(n,n*n):
        YW.extend(4*[j])

    for i in range(1,2**(nsys-1)):
        temp = map(int, np.binary_repr(i,nsys))
        subsys = []
        for index, j in enumerate(temp):
            if j==1:
                subsys.append(index)

        #create dummy array
        z = np.zeros((n,n), dtype=int)
        for x in range(0,n):
            for y in range(x+1,n):
                z[x][y] = y + n*x- x*(x+1)/2 -x

        #transpose dummy array partially
        z = ptranspose(z,dim,subsys)

        #get posisitions of elements after partial transposition
        for x in xrange(n):
            for y in xrange(n):
                if z[x,y]:
                    XY[z[x,y]-1] = (x,y)

        #x coordinates
        X = []
        for d in range(n):
            X.extend([d*(1+2*n),(d+n)*(1+2*n)])
        for x,y in XY:
            X.extend([x+2*n*y, y+2*n*x, x+2*n*y+2*n*n+n, y+2*n*x+2*n*n+n])
        for x,y in XY:
            X.extend([x+2*n*y+n, y+2*n*x+n, x+2*n*y+2*n*n, y+2*n*x+2*n*n])
        X = 2*X

        #y coordinates
        Y = YW +[y+i*n*n for y in YW]

        G += [ cvxopt.spmatrix(Valle1+Valge0, X, Y, (4*n*n,n*n*2**(nsys-1))) ]
        h += [ cvxopt.matrix( np.eye(2*n) , tc='d' ) ]

        G += [ cvxopt.spmatrix(Valge0+Valle1, X, Y, (4*n*n,n*n*2**(nsys-1))) ]
        h += [ cvxopt.matrix( np.zeros((2*n,2*n)) , tc='d' ) ]

    del X, Y, YW, XY, Valle1, Valge0, z
    # Use either standard solver of cvxopt or the external solver dsdp
    if solver == 'dsdp':
        sol = solvers.sdp(c, Gs=G, hs=h, solver="dsdp")
    elif solver == '':
        sol = solvers.sdp(c, Gs=G, hs=h)
    else:
        return "invalid choice of solver"
    return -(c.T*sol['x'])[0]


#### Example usage ####
# three qubit W state
print 'Initialize three qubit W state'
W = 1/np.sqrt(3)*np.array([0,1,1,0,1,0,0,0])
rho = np.outer(W,W.conjugate())
# calculate GMN
print 'The GMN is given by:', gmn(rho,(2,2,2),False,'')
# calculate renormalized GMN
print 'The renormalized GMN is given by:', gmn(rho,(2,2,2),True,'dsdp')
```

Bibliography

[1] A. Einstein, B. Podolsky, and N. Rosen. Can quantum-mechanical description of physical reality be considered complete? *Phys. Rev.*, 47:777–780, May 1935.

[2] E. Schrödinger. Discussion of probability relations between separated systems. *Mathematical Proceedings of the Cambridge Philosophical Society*, 31:555–563, 10 1935.

[3] J. S. Bell. On the Einstein-Podolsky-Rosen paradox. *Physics*, 1:195–200, 1964.

[4] A. Aspect, P. Grangier, and G. Roger. Experimental tests of realistic local theories via bell's theorem. *Phys. Rev. Lett.*, 47:460–463, Aug 1981.

[5] A. Aspect, J. Dalibard, and G. Roger. Experimental test of bell's inequalities using time- varying analyzers. *Phys. Rev. Lett.*, 49:1804–1807, Dec 1982.

[6] A. K. Ekert. Quantum cryptography based on bell's theorem. *Phys. Rev. Lett.*, 67:661–663, Aug 1991.

[7] C. H. Bennett and S. J. Wiesner. Communication via one- and two-particle operators on einstein-podolsky-rosen states. *Phys. Rev. Lett.*, 69:2881–2884, Nov 1992.

[8] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters. Teleporting an unknown quantum state via dual classical and einstein-podolsky-rosen channels. *Phys. Rev. Lett.*, 70:1895–1899, Mar 1993.

[9] R. P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6-7):467–488, 1982.

[10] D. Deutsch. Quantum theory, the church-turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818):97–117, 1985.

[11] P. W. Shor. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, 52:R2493–R2496, Oct 1995.

[12] D. M. Greenberger, M. A. Horne, and A. Zeilinger. Going beyond bell's theorem. In *Bell's theorem, quantum theory and conceptions of the universe*, pages 69–72. Springer, 1989.

[13] R. Raussendorf and H. J. Briegel. A one-way quantum computer. *Phys. Rev. Lett.*, 86:5188–5191, May 2001.

[14] V. Giovannetti, S. Lloyd, and L. Maccone. Quantum metrology. *Phys. Rev. Lett.*, 96:010401, Jan 2006.

[15] B. Jungnitsch, T. Moroder, and O. Gühne. Taming multiparticle entanglement. *Phys. Rev. Lett.*, 106:190502, May 2011.

[16] E. Lötstedt and U. D. Jentschura. Triple compton effect: A photon splitting into three upon collision with a free electron. *Phys. Rev. Lett.*, 109:059901, Aug 2012.

[17] E. Lötstedt and U. D. Jentschura. Theoretical study of the compton effect with correlated three-photon emission: From the differential cross section to high-energy triple-photon entanglement. *Phys. Rev. A*, 87:033401, Mar 2013.

[18] S. N. Filippov, A. A. Melnikov, and M. Ziman. Dissociation and annihilation of multipartite entanglement structure in dissipative quantum dynamics. *Phys. Rev. A*, 88:062328, Dec 2013.

[19] M. Ali and O. Gühne. Robustness of multiparticle entanglement: specific entanglement classes and random states. *Journal of Physics B: Atomic, Molecular and Optical Physics*, 47(5):055503, 2014.

[20] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge Series on Information and the Natural Sciences. Cambridge University Press, 2000.

[21] D. Gottesman. *Stabilizer codes and quantum error correction*. PhD thesis, California Institute of Technology, 1997.

[22] D. Gottesman. Class of quantum error-correcting codes saturating the quantum hamming bound. *Phys. Rev. A*, 54:1862–1868, Sep 1996.

[23] O. Gühne and M. Seevinck. Separability criteria for genuine multiparticle entanglement. *New Journal of Physics*, 12(5):053002, 2010.

[24] B. Jungnitsch, T. Moroder, and O. Gühne. Entanglement witnesses for graph states: General theory and examples. *Phys. Rev. A*, 84:032310, Sep 2011.

[25] D. Cavalcanti, R. Chaves, L. Aolita, L. Davidovich, and A. Acín. Open-system dynamics of graph-state entanglement. *Phys. Rev. Lett.*, 103:030502, Jul 2009.

[26] M. Van den Nest, J. Dehaene, and B. De Moor. Graphical description of the action of local clifford transformations on graph states. *Phys. Rev. A*, 69:022316, Feb 2004.

[27] M. Van den Nest, J. Dehaene, and B. De Moor. Finite set of invariants to characterize local clifford equivalence of stabilizer states. *Phys. Rev. A*, 72:014307, Jul 2005.

[28] M. Van den Nest. *Local equivalence of stabilizer states and codes (PhD thesis)*. PhD thesis, Katholieke Universiteit Leuven (Belgium), 2005.

[29] Z. Ji, J. Chen, Z. Wei, and M. Ying. The lu-lc conjecture is false. *Quantum Info. Comput.*, 10(1):97–108, January 2010.

[30] A. Osterloh, L. Amico, G. Falci, and R. Fazio. Scaling of entanglement close to a quantum phase transition. *Nature*, 416(6881):608–610, Apr 2002.

[31] T. J. Osborne and M. A. Nielsen. Entanglement in a simple quantum phase transition. *Phys. Rev. A*, 66:032110, Sep 2002.

[32] G. Vidal, J. I. Latorre, E. Rico, and A. Kitaev. Entanglement in quantum critical phenomena. *Phys. Rev. Lett.*, 90:227902, Jun 2003.

[33] L.-A. Wu, M. S. Sarandy, and D. A. Lidar. Quantum phase transitions and bipartite entanglement. *Phys. Rev. Lett.*, 93:250404, Dec 2004.

[34] T. Roscilde, P. Verrucchi, A. Fubini, S. Haas, and V. Tognetti. Studying quantum spin systems through entanglement estimators. *Phys. Rev. Lett.*, 93:167203, Oct 2004.

[35] P. Nataf, M. Dogan, and K. Le Hur. Heisenberg uncertainty principle as a probe of entanglement entropy: Application to superradiant quantum phase transitions. *Phys. Rev. A*, 86:043807, Oct 2012.

[36] Y. Yao, H.-W. Li, C.-M. Zhang, Z.-Q. Yin, W. Chen, G.-C. Guo, and Z.-F. Han. Performance of various correlation measures in quantum phase transitions using the quantum renormalization-group method. *Phys. Rev. A*, 86:042102, Oct 2012.

[37] F. Altintas and R. Eryigit. Correlation and nonlocality measures as indicators of quantum phase transitions in several critical systems. *Annals of Physics*, 327(12):3084 – 3101, 2012.

[38] L. Lepori, G. De Chiara, and A. Sanpera. Scaling of the entanglement spectrum near quantum phase transitions. *Phys. Rev. B*, 87:235107, Jun 2013.

[39] V. Azimi Mousolou, C. M. Canali, and E. Sjöqvist. Unifying geometric entanglement and geometric phase in a quantum phase transition. *Phys. Rev. A*, 88:012310, Jul 2013.

[40] L. Amico, R. Fazio, A. Osterloh, and V. Vedral. Entanglement in many-body systems. *Rev. Mod. Phys.*, 80:517–576, May 2008.

[41] F. Verstraete, V. Murg, and J. I. Cirac. Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems. *Advances in Physics*, 57(2):143–224, 2008.

[42] N. Schuch. Condensed Matter Applications of Entanglement Theory. *ArXiv e-prints*, June 2013.

[43] T.-C. Wei, D. Das, S. Mukhopadyay, S. Vishveshwara, and P. M. Goldbart. Global entanglement and quantum criticality in spin chains. *Phys. Rev. A*, 71:060305, Jun 2005.

[44] O. Gühne, G. Tóth, and H. J. Briegel. Multipartite entanglement in spin chains. *New Journal of Physics*, 7(1):229, 2005.

[45] R. Orùs. Universal geometric entanglement close to quantum phase transitions. *Phys. Rev. Lett.*, 100:130502, Apr 2008.

[46] H. S. Dhar, A. Sen(De), and U. Sen. Characterizing genuine multisite entanglement in isotropic spin lattices. *Phys. Rev. Lett.*, 111:070501, Aug 2013.

[47] H. S. Dhar, A. Sen(De), and U. Sen. The density matrix recursion method: genuine multisite entanglement distinguishes odd from even quantum spin ladder states. *New Journal of Physics*, 15(1):013043, 2013.

[48] S. M. Giampaolo and B. C. Hiesmayr. Genuine multipartite entanglement in the *xy* model. *Phys. Rev. A*, 88:052305, Nov 2013.

[49] J. Stasińska, B. Rogers, M. Paternostro, G. De Chiara, and A. Sanpera. Long-range multipartite entanglement close to a first-order quantum phase transition. *Phys. Rev. A*, 89:032330, Mar 2014.

[50] K. Życzkowski, P. Horodecki, A. Sanpera, and M. Lewenstein. Volume of the set of separable states. *Phys. Rev. A*, 58:883–892, Aug 1998.

[51] G. Vidal and R. F. Werner. Computable measure of entanglement. *Physical Review A*, 65(3):1–11, February 2002.

[52] D. Yang, M. Horodecki, R. Horodecki, and B. Synak-Radtke. Irreversibility for all bound entangled states. *Phys. Rev. Lett.*, 95:190501, Oct 2005.

[53] B. Synak-Radtke and M. Horodecki. On asymptotic continuity of functions of quantum states. *Journal of Physics A: Mathematical and General*, 39(26):L423, 2006.

[54] G. A. Paz-Silva and J. H. Reina. Total correlations as multi-additive entanglement monotones. *Journal of Physics A: Mathematical and Theoretical*, 42(5):055306, 2009.

[55] Y. Dong, K. Horodecki, M. Horodecki, P. Horodecki, J. Oppenheim, and W. Song. Squashed entanglement for multipartite states and entanglement measures based on the mixed convex roof. *Information Theory, IEEE Transactions on*, 55(7):3375–3387, 2009.

[56] J. T. Barreiro, P. Schindler, O. Gühne, T. Monz, M. Chwalla, C. F. Roos, M. Hennrich, and R. Blatt. Experimental multiparticle entanglement dynamics induced by decoherence. *Nat Phys*, 6(12):943–946, Dec 2010.

[57] H. Kampermann, O. Gühne, C. Wilmott, and D. Bruß. Algorithm for characterizing stochastic local operations and classical communication classes of multiparticle entanglement. *Phys. Rev. A*, 86:032307, Sep 2012.

[58] R. Horodecki, P. Horodecki, M. Horodecki, and K. Horodecki. Quantum entanglement. *Rev. Mod. Phys.*, 81:865–942, Jun 2009.

[59] A. Peres. Separability criterion for density matrices. *Phys. Rev. Lett.*, 77:1413–1415, Aug 1996.

[60] O. Gühne and G. Tóth. Entanglement detection. *Physics Reports*, 474(1–6):1 – 75, 2009.

[61] M. Hein, W. Dür, J. Eisert, R. Raußendorf, M. Van den Nest, and H. J. Briegel. Entanglement in graph states and its applications. In G. Casati, D.L. Shepelyansky, and P. Zoller, editors, *Quantum computers, algorithms and chaos*, volume 162 of *International School of Physics "Enrico Fermi"*. IOS Press, 2006.

[62] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, February 1978.

[63] D. Deutsch. Uncertainty in quantum measurements. *Phys. Rev. Lett.*, 50:631–633, Feb 1983.

[64] M. Horodecki, P. Horodecki, and R. Horodecki. Separability of mixed states: necessary and sufficient conditions. *Physics Letters A*, 223(1–2):1 – 8, 1996.

[65] B. M. Terhal. Bell inequalities and the separability criterion. *Physics Letters A*, 271(5–6):319 – 326, 2000.

[66] M. Lewenstein, B. Kraus, J. I. Cirac, and P. Horodecki. Optimization of entanglement witnesses. *Phys. Rev. A*, 62:052310, Oct 2000.

[67] W. Dür, G. Vidal, and J. I. Cirac. Three qubits can be entangled in two inequivalent ways. *Phys. Rev. A*, 62:062314, Nov 2000.

[68] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.

[69] B. Jungnitsch. PPTmixer. `www.mathworks.com/matlabcentral/fileexchange/30968`, 2011.

[70] C. H. Bennett, G. Brassard, S. Popescu, B. Schumacher, J. A. Smolin, and W. K. Wootters. Purification of noisy entanglement and faithful teleportation via noisy channels. *Phys. Rev. Lett.*, 76:722–725, Jan 1996.

[71] C. H. Bennett, D. P. DiVincenzo, J. A. Smolin, and W. K. Wootters. Mixed-state entanglement and quantum error correction. *Phys. Rev. A*, 54:3824–3851, Nov 1996.

[72] G. Vidal. Entanglement monotones. *Journal of Modern Optics*, 47(2-3):355–376, 2000.

[73] A. Uhlmann. Entropy and optimal decompositions of states relative to a maximal commutative subalgebra. *Open Systems & Information Dynamics*, 5(3):209–228, 1998.

[74] W. K. Wootters. Entanglement of formation of an arbitrary state of two qubits. *Phys. Rev. Lett.*, 80:2245–2248, Mar 1998.

[75] Z.-H. Ma, Z.-H. Chen, J.-L. Chen, C. Spengler, A. Gabriel, and M. Huber. Measure of genuine multipartite entanglement with computable lower bounds. *Phys. Rev. A*, 83:062325, Jun 2011.

[76] S. M. Hashemi Rafsanjani, M. Huber, C. J. Broadbent, and J. H. Eberly. Genuinely multipartite concurrence of n-qubit x matrices. *Phys. Rev. A*, 86:062303, Dec 2012.

[77] F. G. S. L. Brandão. Quantifying entanglement with witness operators. *Phys. Rev. A*, 72:022310, Aug 2005.

[78] M. Grassl, A. Klappenecker, and M. Rotteler. Graphs, quadratic forms, and quantum codes. In *Information Theory, 2002. Proceedings. 2002 IEEE International Symposium on*, pages 45–, 2002.

[79] D. Schlingemann. Stabilizer codes can be realized as graph codes. *Quantum Info. Comput.*, 2(4):307–323, June 2002.

[80] Z.-H. Ma, Z.-H. Chen, J.-L. Chen, C. Spengler, A. Gabriel, and M. Huber. Measure of genuine multipartite entanglement with computable lower bounds. *Phys. Rev. A*, 83:062325, Jun 2011.

[81] J.-Y. Wu, H. Kampermann, D. Bruß, C. Klöckl, and M. Huber. Determining lower bounds on a measure of multipartite entanglement from few local observables. *Phys. Rev. A*, 86:022319, Aug 2012.

[82] M. Huber, P. Erker, H. Schimpf, A. Gabriel, and B. Hiesmayr. Experimentally feasible set of criteria detecting genuine multipartite entanglement in n-qubit dicke states and in higher-dimensional systems. *Phys. Rev. A*, 83:040301, Apr 2011.

[83] T. Moroder, J.-D. Bancal, Y.-C. Liang, M. Hofmann, and O. Gühne. Device-independent entanglement quantification and related applications. *Phys. Rev. Lett.*, 111:030501, Jul 2013.

[84] K. Chen, S. Albeverio, and S.-M. Fei. Concurrence of arbitrary dimensional bipartite quantum states. *Phys. Rev. Lett.*, 95:040504, Jul 2005.

[85] O. Gühne, B. Jungnitsch, T. Moroder, and Y. S. Weinstein. Multiparticle entanglement in graph-diagonal states: Necessary and sufficient conditions for four qubits. *Phys. Rev. A*, 84:052319, Nov 2011.

[86] X. Chen, P. Yu, L. Jiang, and M. Tian. Genuine entanglement of four-qubit cluster diagonal states. *Phys. Rev. A*, 87:012322, Jan 2013.

[87] T. Eggeling and R. F. Werner. Separability properties of tripartite states with $U \otimes U \otimes U$ symmetry. *Phys. Rev. A*, 63:042111, Mar 2001.

[88] M. Huber and J. I. de Vicente. Structure of multidimensional entanglement in multipartite systems. *Phys. Rev. Lett.*, 110:030501, Jan 2013.

[89] C. Eltschka and J. Siewert. Negativity as an estimator of entanglement dimension. *Phys. Rev. Lett.*, 111:100503, Sep 2013.

[90] V. Coffman, J. Kundu, and W. K. Wootters. Distributed entanglement. *Phys. Rev. A*, 61:052306, Apr 2000.

[91] M. Koashi and A. Winter. Monogamy of quantum entanglement and other correlations. *Phys. Rev. A*, 69:022309, Feb 2004.

[92] T. J. Osborne and F. Verstraete. General monogamy inequality for bipartite qubit entanglement. *Phys. Rev. Lett.*, 96:220503, Jun 2006.

[93] H. He and G. Vidal. Disentangling Theorem and Monogamy for Entanglement Negativity. *ArXiv e-prints*, January 2014.

[94] E. Lieb, T. Schultz, and D. Mattis. Two soluble models of an antiferromagnetic chain. *Annals of Physics*, 16(3):407 – 466, 1961.

[95] P. Pfeuty. The one-dimensional ising model with a transverse field. *Annals of Physics*, 57(1):79 – 90, 1970.

[96] E. Barouch, B. M. McCoy, and M. Dresden. Statistical mechanics of the xy model. i. *Phys. Rev. A*, 2:1075–1092, Sep 1970.

[97] E. Barouch and B. M. McCoy. Statistical mechanics of the xy model. ii. spin-correlation functions. *Phys. Rev. A*, 3:786–804, Feb 1971.

[98] D. Patanè, R. Fazio, and L. Amico. Bound entanglement in the xy model. *New Journal of Physics*, 9(9):322, 2007.

[99] E. R. Caianiello and S. Fubini. On the algorithm of dirac spurs. *Il Nuovo Cimento*, 9(12):1218–1226, 1952.

[100] L. Gurvits and H. Barnum. Largest separable balls around the maximally mixed bipartite quantum state. *Phys. Rev. A*, 66:062311, Dec 2002.

[101] M. E. Fisher. in critical phenomena. In M.S. Green, editor, *Proceedings of 1970 Enrico Fermi International School of Physics*, pages 1–99. Academic Press, New York, 1971.

[102] M. N. Barber. *Phase transitions and critical phenomena*. Access Online via Elsevier, 1983.

[103] A. Sen(De), U. Sen, and M. Lewenstein. Dynamical phase transitions and temperature-induced quantum correlations in an infinite spin chain. *Phys. Rev. A*, 72:052319, Nov 2005.

[104] T. Yu and J. H. Eberly. Evolution from entanglement to decoherence of bipartite mixed "x" states. *Quantum Information & Computation*, 7(5):459–468, 2007.

[105] T. Yu and J. H. Eberly. Finite-time disentanglement via spontaneous emission. *Phys. Rev. Lett.*, 93:140404, Sep 2004.

[106] T. Yu and J. H. Eberly. Quantum open system theory: Bipartite aspects. *Phys. Rev. Lett.*, 97:140403, Oct 2006.

[107] A. R. P. Rau. Algebraic characterization of x -states in quantum information. *Journal of Physics A: Mathematical and Theoretical*, 42(41):412002, 2009.

[108] S. Vinjanampathy and A. R. P. Rau. Generalized x states of n qubits and their symmetries. *Phys. Rev. A*, 82:032336, Sep 2010.

[109] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 2010.

[110] A. Cabello, A. J. López-Tarrida, P. Moreno, and J. R. Portillo. Entanglement in eight-qubit graph states. *Physics Letters A*, 373(26):2219 – 2225, 2009.

[111] André Bouchet. An efficient algorithm to recognize locally equivalent graphs. *Combinatorica*, 11(4):315–329, 1991.

[112] E. M. Rains. A semidefinite program for distillable entanglement. *Information Theory, IEEE Transactions on*, 47(7):2921–2933, 2001.

[113] A. C. Doherty, P. A. Parrilo, and F. M. Spedalieri. Distinguishing separable and entangled states. *Phys. Rev. Lett.*, 88:187904, Apr 2002.

[114] F. G. S. L. Brandão and R. O. Vianna. Robust semidefinite programming approach to the separability problem. *Phys. Rev. A*, 70:062309, Dec 2004.

[115] M. Ježek, J. Řeháček, and J. Fiurášek. Finding optimal strategies for minimum-error quantum-state discrimination. *Phys. Rev. A*, 65:060301, Jun 2002.

[116] Y. C. Eldar. A semidefinite programming approach to optimal unambiguous discrimination of quantum states. *Information Theory, IEEE Transactions on*, 49(2):446–456, 2003.

[117] K. Audenaert and B. De Moor. Optimizing completely positive maps using semidefinite programming. *Phys. Rev. A*, 65:030302, Feb 2002.

[118] R. T. Rockafellar. *Convex analysis*. Princeton university press, 2nd ed. edition, 1970.

[119] S. Boyd and L. Vandenberghe. Convex optimization. 2004, 2004.

[120] L. Novo, T. Moroder, and O. Gühne. Genuine multiparticle entanglement of permutationally invariant states. *Phys. Rev. A*, 88:012305, Jul 2013.

[121] MATLAB. *version 8.2 (R2013b)*. The MathWorks Inc., Natick, Massachusetts, 2013.

[122] W. A. Stein et al. *Sage Mathematics Software (Version 6.0)*. The Sage Development Team, 2014. `http://www.sagemath.org`.

[123] J. Lofberg. YALMIP : a toolbox for modeling and optimization in matlab. `http://www.mathworks.com/matlabcentral/newsreader/view_thread/59636`, 2004.

[124] J. Lofberg. YALMIP : a toolbox for modeling and optimization in matlab. In *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, pages 284–289, 2004.

[125] M. S. Andersen, J. Dahl, and L. Vandenberghe. CVXOPT: A python package for convex optimization, version 1.1.5. `abel.ee.ucla.edu/cvxopt`, 2012.

[126] J. F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12:625–653, 1999. Version 1.05 RC5 available from `http://sedumi.ie.lehigh.edu/`.

[127] K. C. Toh, M. J. Todd, and R. H. Tütüncü. Sdpt3 – a matlab software package for semidefinite programming. *OPTIMIZATION METHODS AND SOFTWARE*, 11:545–581, 1999.

[128] R. H. Tütüncü, K. C. Toh, and M. J. Todd. Solving semidefinite-quadratic-linear programs using sdpt3. *Mathematical Programming*, 95(2):189–217, 2003.

[129] S. J. Benson and Y. Ye. DSDP5: Software for semidefinite programming. Technical Report ANL/MCS-P1289-0905, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, sep 2005. Submitted to ACM Transactions on Mathematical Software.

[130] D. Upper. The unsuccessful self-treatment of a case of "writer's block"[1]. *J Appl Behav Anal.*, 7:497, Fall 1974.

Danksagung

Eine wissenschaftliche Arbeit ist viel mehr als nur die Summe ihrer Teile. Ihr zugrunde liegt ein langer Prozess, welcher nicht zu denken wäre ohne all jene, die direkt oder indirekt zu dessen Gelingen beitragen. Daher möchte ich mich an dieser Stelle bei einigen Menschen bedanken, welche in ganz besonderem Maße diesen Prozess unterstützt haben.

Zuallererst danke ich meinem Doktorvater Prof. Dr. Otfried Gühne dafür, dass er mir die Gelegenheit gegeben hat, an einem Thema zu forschen, welches sowohl fordernd als auch interessant war, und dass er mich in den drei Jahren meiner Promotion dabei stetig unterstützt hat.

Dann möchte ich auch meinen Kollegen Tobias Moroder, Matthias Kleinmann, Sönke Niekamp, Leonardo Novo, Costantino Budroni, Frank E. S. Steinhoff, Sabine Wölk, Mazhar Ali, Marius Paraschiv, Cristina Ritz, Florian Köppen und Lars Erik Buchholz für anregende Diskussionen und gemeinsames Kuchenschnabulieren im Kaffeeraum danken. Auch meinen Koauthoren Prof. Dr. Otfried Gühne, Dr. habil. Andreas Osterloh und Dr. Tobias Moroder, welche mit ihren Beiträgen, Anregungen und Hilfestellungen die gemeinsamen Publikationen voran gebracht haben, möchte ich an dieser Stelle meinen Dank zukommen lassen. Dank gebührt auch meinen Adminkollegen Hans-Werner Boschmann, Sascha Turczyk, Florian Hartmann und Marco Sekulla für die hervorragende Zusammenarbeit im Dienste der IT.

Das Leben hält natürlich mehr parat als nur die Wissenschaft und daher möchte ich auch all jenen danken, die mir halfen neue Kraft und Energie für meine Arbeit zu schöpfen:

Allen voran all jenen, die mir halfen als ich sprichwörtlich das Dach über dem Kopf verlor: Margot Reichmann, Michael Och-Reichmann und Jasmin Reichmann, Daniela Lehmann, Margit Steinleitner und all den Umzugshelfern. All meinen Freunden, die mir so lieb sind, und insbesondere Henrik Grundmann und Eva Bär-Grundmann die mich in Siegen besucht haben. Meiner wunderbaren Familie danke ich für all die Unterstützung und dafür, dass sie dafür Verständnis aufbrachten, dass ich so selten zugegen war. Den meisten Dank bekommt zu guter Letzt allerdings meine Liebe Monika. Dafür, dass du bist und mehr noch wie du bist.

List of publications

The Chapters 3 and 2 are based on the publications [3] and [4], respectively. The publication [1] has been written on the basis of my diploma thesis.

[1] M. Hofmann, G. Rudolph and M. Schmidt. On the reflection type decomposition of the adjoint reduced phase space of a compact semisimple Lie group. *Journal of Mathematical Physics*, 54:083505, Aug 2013.

[2] T. Moroder, J. Bancal, Y. Liang, M. Hofmann and O. Gühne. Device-independent entanglement quantification and related applications. *Phys. Rev. Lett.*, 111:030501, Jul 2013.

[3] M. Hofmann, A. Osterloh and O. Gühne. Scaling of genuine multiparticle entanglement at a quantum phase transition. *Phys. Rev. B*, 89:134101, Apr 2014.

[4] M. Hofmann, T. Moroder and O. Gühne. Analytical characterization of the genuine multiparticle negativity. *J. Phys. A: Math. Theor.*, 47:155301, Apr 2014.